The Necessity of Bounded Treewidth for Efficient Inference in Bayesian Networks

Johan H.P. Kwisthout and **Hans L. Bodlaender** and **L.C. van der Gaag** [1]

Abstract. Algorithms for probabilistic inference in Bayesian networks are known to have running times that are worst-case exponential in the size of the network. For networks with a moralised graph of bounded treewidth, however, these algorithms take a time which is linear in the network's size. In this paper, we show that under the assumption of the Exponential Time Hypothesis (ETH), small treewidth of the moralised graph actually is a *necessary* condition for a Bayesian network to render inference efficient by an algorithm accepting arbitrary instances. We thus show that no algorithm can exist that performs inference on arbitrary Bayesian networks of unbounded treewidth in polynomial time, unless the ETH fails.

1 INTRODUCTION

The most important computational problem for Bayesian networks is probabilistic inference, that is, the problem of establishing a posterior probability distribution $\Pr(X \mid \mathbf{e})$ for a variable X of interest, given evidence $\mathbf{e}$ for some (other) variables in the network. Several researchers have investigated this problem and have designed various algorithms taking different approaches, such as message passing [1], variable elimination [2], and junction-tree propagation [3]. Current Bayesian-network tools mostly implement the junction-tree propagation algorithm, or a variant thereof, for probabilistic inference.

Algorithms for probabilistic inference with arbitrary Bayesian networks all have a running time that is worst-case exponential in the size of the network at hand. When the graphical structure of the network is a polytree with bounded indegree, probabilistic inference can be done in polynomial time, however, for example using the message-passing algorithm. So, while for specific classes of Bayesian networks probabilistic inference can be performed efficiently, for each algorithm there are networks for which inference will take exponential time. Researchers have investigated the computational complexity of the problem of probabilistic inference in general and have established unfavourable complexity results. Cooper [4] was the first to prove NP-hardness of the problem. Other researchers since then showed that specific variants of the problem are not merely NP-hard: some variants were proven PP-complete [5] or #P-complete [6].

The complexity results cited above concern inference in Bayesian networks in general, that is, these results pertain to arbitrary instances of the problem of probabilistic inference. Investigation of the runtime properties of the junction-tree propagation algorithm has shown that computing a posterior probability distribution for a variable in a network whose moralised graph has bounded treewidth, actually is exponential *only* in this graph's treewidth and linear in the size of the network. This complexity result is of high practical value for real-life application of Bayesian networks, since it implies that probabilistic inference can be feasibly performed on networks of bounded size whose moralised graph has a small treewidth. This property is well-known among network engineers and is commonly translated into the heuristic guideline of ensuring that all variables in a network under construction have a limited number of parents.

In this paper, we investigate the necessity of the property of bounded treewidth for efficient probabilistic inference in Bayesian networks. We show that under assumption of the Exponential Time Hypothesis (ETH), small treewidth of a network's moralised graph is not just a sufficient but actually a *necessary* condition for the network to render probabilistic inference efficient by an algorithm accepting arbitrary instances. In other words, we show that, unless the ETH fails, no algorithm can exist that solves arbitrary instances of probabilistic inference with large treewidth in polynomial time. There might nevertheless be some specific structural graph property that may be exploited by an algorithm to solve particular classes of instances in polynomial time.

The necessity of small treewidth for algorithms to run in polynomial time has also been investigated for the closely related problems of constraint satisfaction and graph homomorphism [7, 8]. Under assumption of the ETH, these problems were shown to not allow algorithms solving instances of unbounded treewidth in polynomial time. In addition, a sub-exponential lower bound was derived on the running time of any algorithm taking arbitrary instances with large treewidth [9]. We build upon this result and show that the constraint-satisfaction problem can be reduced to the problem of probabilistic inference in Bayesian networks, in polynomial time and preserving treewidth. From this reduction, we then have that if an algorithm for probabilistic inference exists that solves arbitrary instances with large treewidth in sub-exponential time, then this algorithm also solves such instances of the constraint-satisfaction problem in sub-exponential time, which would contradict the ETH.

The paper is organised as follows. In Section 2, we introduce our notational conventions and review some concepts from complexity theory. Section 3 outlines our basic approach to proving the paper's main result. Section 4 then presents the actual proof of the result stating that bounded treewidth is a necessary condition for efficient inference in Bayesian networks. The paper ends with our concluding observations in Section 5.

2 PRELIMINARIES

We introduce our notational conventions, and provide some preliminaries from graph theory and from complexity theory.

[1] Department of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands; email: {johank,hansb,linda}@cs.uu.nl

2.1 Bayesian networks

A Bayesian network $\mathcal{B}$ is a model of a joint probability distribution $\Pr$ over a set of stochastic variables. The network includes a directed acyclic graph $\mathbf{G}_\mathcal{B} = (\mathbf{V}, \mathbf{A})$, where $\mathbf{V}$ denotes the set of variables and $\mathbf{A}$ captures the probabilistic (in)dependencies between them. We use upper case letters X to denote individual variables from $\mathbf{V}$ and bold-faced upper case letters $\mathbf{X}$ to denote sets of variables. A lower case letter x is used to indicate a value of a variable X, and a bold-faced lower case letter $\mathbf{x}$ denotes a joint value assignment to a set of variables $\mathbf{X}$; note that variables can have arbitrarily many values. To capture the strengths of the dependency relationships between the variables, a network further includes a set $\Gamma = \{\Pr_X \mid X \in \mathbf{V}\}$ of (conditional) probability distributions $\Pr_X(X \mid \mathbf{y})$ for each variable X given all value assignments $\mathbf{y}$ to the set of parents $\pi(X)$ of X in the graph $\mathbf{G}$. The network thereby models the joint probability distribution $\Pr(\mathbf{V}) = \prod_{X \in \mathbf{V}} \Pr_X(X \,|\, \pi(X))$ over its variables.

In this paper, we study the computational complexity of probabilistic inference in Bayesian networks, that is, we study the problem of computing a posterior probability distribution $\Pr(X \mid \mathbf{e})$ over a variable X of interest, given evidence $\mathbf{e}$ for some (other) variables in the network. For formulating our results in the sequel, we introduce the decision variant of the problem of positive inference:

POSITIVE INFERENCE
Instance: A Bayesian network $\mathcal{B} = (\mathbf{G}_\mathcal{B}, \Gamma)$ with its joint probability distribution $\Pr$, an output variable $X \in \mathbf{V}$ with a value x, and a set of evidence variables $\mathbf{E} \subseteq \mathbf{V}$ with a joint value assignment $\mathbf{e}$.
Question: Does $\Pr(x \,|\, \mathbf{e}) > 0$ hold ?

The size of a Bayesian network $\mathcal{B}$, denoted $\|\mathcal{B}\|$, is taken to be the number of bits needed to describe $\mathcal{B}$ by a reasonable encoding. Its complexity is measured in this paper by the treewidth of the moralisation $\mathbf{G}_\mathcal{B}^\mathbf{M}$ of its graph $\mathbf{G}_\mathcal{B}$. This moralisation is the undirected graph that is obtained from $\mathbf{G}_\mathcal{B}$ by adding arcs so as to connect all pairs of parents of a variable, and then dropping all directions; we will use the phrase 'moralised graph' to refer to the moralisation of the graph of a network. A triangulation of the moralised graph $\mathbf{G}_\mathcal{B}^\mathbf{M}$ is any graph $\mathbf{G}_\mathbf{T}$ that embeds $\mathbf{G}_\mathcal{B}^\mathbf{M}$ as a subgraph and in addition is chordal, that is, it does not include loops of more than three variables without any pair being adjacent in $\mathbf{G}_\mathbf{T}$. A tree-decomposition of a triangulation $\mathbf{G}_\mathbf{T}$ is a tree $\mathbf{T}_\mathbf{G}$ such that

- each node $\mathbf{X}_i$ in $\mathbf{T}_\mathbf{G}$ is a bag of nodes which constitute a clique in $\mathbf{G}_\mathbf{T}$;
- for every i, j, k, if $\mathbf{X}_j$ lies on the path from $\mathbf{X}_i$ to $\mathbf{X}_k$ in $\mathbf{T}_\mathbf{G}$, then $\mathbf{X}_i \cap \mathbf{X}_k \subseteq \mathbf{X}_j$.

The width of the tree-decomposition $\mathbf{T}_\mathbf{G}$ of the graph $\mathbf{G}_\mathbf{T}$ equals $\max_i(|\mathbf{X}_i| - 1)$, that is, it equals the size of the largest clique in $\mathbf{G}_\mathbf{T}$, minus 1. The treewidth of the moralised graph of the network $\mathcal{B}$, denoted $\mathrm{tw}(\mathbf{G}_\mathcal{B}^\mathbf{M})$, now is the minimum width over all possible tree-decompositions of $\mathbf{G}_\mathcal{B}^\mathbf{M}$.

In the proofs of our results in the sequel, we use a so-called nice tree-decomposition of the moralised graph of a Bayesian network. Such a decomposition has a particularly simple structure: it is a rooted tree in which every node has at most two children. More specifically, each node in a nice tree-decomposition $\mathbf{T}$ is either a leaf node, an insert node, a forget node, or a join node:

- a leaf node $\mathbf{X}_i$ is a leaf in $\mathbf{T}$ with $|\mathbf{X}_i| = 1$;
- an insert node $\mathbf{X}_i$ is a node in $\mathbf{T}$ with a single child $\mathbf{X}_j$ such that $\mathbf{X}_i = \mathbf{X}_j \cup \{Y\}$ for some $Y \in \mathbf{V} \setminus \mathbf{X}_j$;
- a forget node $\mathbf{X}_i$ is a node in $\mathbf{T}$ with a single child $\mathbf{X}_j$ such that $\mathbf{X}_i = \mathbf{X}_j \setminus \{Y\}$ for some $Y \in \mathbf{X}_j$;
- a join node $\mathbf{X}_i$ is a node in $\mathbf{T}$ with two children $\mathbf{X}_j$ and $\mathbf{X}_k$ such that $\mathbf{X}_i = \mathbf{X}_j = \mathbf{X}_k$.

From graph theory, we have that any tree-decomposition $\mathbf{T}$ of width w with b nodes, can be converted into a nice tree-decomposition of the same width with $\mathcal{O}(w \cdot b)$ nodes, in time $\mathcal{O}(f(w) \cdot b)$ for a polynomially computable function f [10, 11].

2.2 Complexity theory

In this paper, we use some basic constructs from computational complexity theory. We will briefly review these constructs here; for further details, we refer to for example [12, 13].

We assume that for every computational problem P, there exists an encoding which translates arbitrary instances of P into strings, such that the *yes*-instances of P constitute a language; the *no*-instances of the problem are not included in the language. We now say that a computational problem Q is polynomial-time reducible to another problem P if there exists a polynomial-time computable function f such that $x \in Q$ if and only if $f(x) \in P$. Such reductions are commonly assumed to be polynomial-time many-one reductions. In this paper, however, we will encounter another type of reduction which, in addition to being computable in polynomial time, serves to preserve some structural property among instances.

Formally, a complexity class is a class of languages, where each language is an encoding of a computational problem. We say that a problem P is hard for a specific complexity class if every problem Q from the class can be reduced to P by a polynomial-time reduction. The problem P is complete for the class if it is hard for the class and in addition is a member of the class. The problem P may then be regarded at least as hard as any other problem from the class: since any problem Q from the class can be reduced to P in polynomial time, a polynomial-time algorithm for P would imply a polynomial-time algorithm for every problem in the class.

The main complexity result presented in this paper pertains to the problem of positive inference in Bayesian networks reviewed above. Our result is proved basically by a reduction from the constraint-satisfaction problem. We state the latter problem more formally:

CONSTRAINT SATISFACTION
Instance: A constraint-satisfaction tuple $(\mathbf{V}, \mathbf{D}, \mathbf{C})$, where $\mathbf{V}$ is a set of variables, $\mathbf{D}$ is a set of values, and $\mathbf{C}$ is a set of constraints $\langle t, \mathbf{R}\rangle$, where $t \in \mathbf{V} \times \mathbf{V}$ is a pair of variables and $\mathbf{R} \subset \mathbf{D} \times \mathbf{D}$ is a (non-universal) binary relation over $\mathbf{D}$.
Question: Is there an assignment function $f\colon \mathbf{V} \to \mathbf{D}$ such that every constraint from $\mathbf{C}$ is satisfied, that is, such that for each constraint $\langle t, \mathbf{R}\rangle \in \mathbf{C}$, with $t = (V_i, V_j)$ the property $(f(V_i), f(V_j)) \in \mathbf{R}$ holds ?

A constraint-satisfaction instance is often represented by its so-called primal graph. The primal graph $\mathbf{G}_\mathcal{I}$ of a constraint-satisfaction instance $\mathcal{I}$ is the undirected graph $\mathbf{G}_\mathcal{I} = (\mathbf{V}, \mathbf{E})$ such that $(V_i, V_j) \in \mathbf{E}$ if and only if there is a constraint $\langle t, \mathbf{R}\rangle \in \mathbf{C}$ with $t = (V_i, V_j)$.

For our main result, we further exploit the *Exponential Time Hypothesis* (ETH). This hypothesis states that there exists a constant $c > 1$ such that deciding any 3SAT instance with n variables takes

at least $\Omega(c^n)$ time [14]. Note that assuming that the ETH holds is a stronger assumption than assuming that P $\neq$ NP: a sub-exponential but not polynomial-time algorithm for the 3SAT problem would contradict the ETH but would not invalidate P $\neq$ NP.

3 THE BASIC APPROACH

The necessity of small treewidth for algorithms to run in polynomial time was recently investigated for the constraint-satisfaction and graph-homomorphism problems [7] and for the problem of inference in undirected graphical models [8]. These problems are closely related to our probabilistic-inference problem in terms of their underlying graph constructs. Under common assumptions from complexity theory, these problems were shown to not allow algorithms solving instances with large treewidth in polynomial time. Marx [9] further derived a sub-exponential lower bound on the running time of any algorithm taking instances of the constraint-satisfaction or graph-homomorphism problems with large treewidth. More specifically, he formulated the following result with respect to constraint satisfaction: for any recursively enumerable class $\mathcal{G}$ of graphs with unbounded treewidth, if there exists a computable function f such that CONSTRAINT SATISFACTION can be decided by an algorithm running in time

$$f(\mathbf{G}_{\mathcal{I}}) \cdot \|\mathcal{I}\|^{o(\frac{\mathrm{tw}(\mathbf{G}_{\mathcal{I}})}{\log \mathrm{tw}(\mathbf{G}_{\mathcal{I}})})}$$

for arbitrary instances $\mathcal{I}$ with a primal graph $\mathbf{G}_{\mathcal{I}} \in \mathcal{G}$ with treewidth $\mathrm{tw}(\mathbf{G}_{\mathcal{I}})$, then the ETH fails. Note that the stated property holds for *any* computable function f and, hence, also for functions that are exponential in the treewidth of the instance's graph.

In this paper, we build upon Marx' result. We show that the constraint-satisfaction problem can be reduced to the problem of positive inference in Bayesian networks, using a polynomial-time reduction which preserves the treewidth of an instance; note that since we are interested in the effect of treewidth on the feasibility of probabilistic inference, it is important that our reduction preserves treewidth. Given an instance $\mathcal{I}$ of the constraint-satisfaction problem, we construct, in polynomial time, an instance $\mathcal{P}$ of the inference problem with the same treewidth up to a constant term, such that a solution to $\mathcal{P}$ yields also a solution to $\mathcal{I}$. Intuitively speaking, we then have that if an algorithm A exists that solves arbitrary instances of the inference problem with large treewidth in sub-exponential time, then we can construct an algorithm B solving instances of the constraint-satisfaction problem with large treewidth in sub-exponential time, which would contradict the ETH.

4 THE COMPLEXITY RESULT

In this section we present a reduction from the constraint-satisfaction problem to the problem of probabilistic inference, which has the special property of preserving the treewidth of an instance. We begin by formally defining this type of reduction.

Definition 1 *Let A and B be computational problems such that* treewidth *is defined on instances of both A and B. We say that A is* polynomial-time treewidth-preserving reducible*, or* tw-reducible*, to B if there exists a polynomial-time computable function g and a linear function l such that $x \in A$ if and only if $g(x) \in B$ and* $\mathrm{tw}(g(x)) = l(\mathrm{tw}(x))$. *The pair (g, l) is called a* tw-reduction.

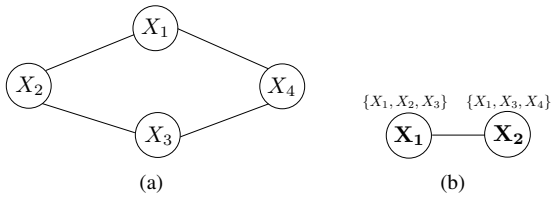

Figure 1. The primal graph $\mathbf{G}_{\mathcal{I}_{\mathrm{ex}}}$ (a) and an associated tree-decomposition (b) of the constraint-satisfaction instance $\mathcal{I}_{\mathrm{ex}}$

We now show that the constraint-satisfaction problem is tw-reducible to the problem of probabilistic inference. Given an instance $\mathcal{I} = \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$ of the constraint-satisfaction problem, we will construct a Bayesian network $\mathcal{B}_{\mathcal{I}} = (\mathbf{G}_{\mathcal{B}_{\mathcal{I}}}, \Gamma)$ that simulates $\mathcal{I}$. Upon doing so, we will take special care that the moralisation of $\mathbf{G}_{\mathcal{B}_{\mathcal{I}}}$ has the same treewidth, up to a constant term, as the primal graph of $\mathcal{I}$. In the construction, we will introduce a new, designated variable A_1 with values TRUE and FALSE, and then show that in the resulting network $\mathcal{B}_{\mathcal{I}}$ we have that $\Pr(A_1 = \mathrm{TRUE}) > 0$ if and only if $\mathcal{I}$ has a solution.

Example 1 *Throughout this section, we will illustrate the various steps in the construction of the Bayesian network by the instance $\mathcal{I}_{\mathrm{ex}} = \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$ of the constraint-satisfaction problem, where $\mathbf{V} = \{X_1, X_2, X_3, X_4\}$, $\mathbf{D} = \{a, b, c\}$, and $\mathbf{C}$ includes*

$$\langle (X_1, X_2),\ \{(a,a),(b,a)\} \rangle$$
$$\langle (X_1, X_4),\ \{(a,a),(a,b),(b,a),(c,a)\} \rangle$$
$$\langle (X_2, X_3), \{(a,b),(b,a),(b,c),(c,b)\} \rangle$$
$$\langle (X_3, X_4), \{(b,a),(b,b)\} \rangle$$

Note that the instance $\mathcal{I}_{\mathrm{ex}}$ is a yes*-instance of the constraint-satisfaction problem; an example solution f_{ex} sets $X_1 = b$, $X_2 = a$, $X_3 = b$, and $X_4 = a$. The primal graph $\mathbf{G}_{\mathcal{I}_{\mathrm{ex}}}$ of the instance and an associated tree-decomposition are given in Figure 1.*

Given a constraint-satisfaction instance $\mathcal{I}$, we begin the construction of the network $\mathcal{B}_{\mathcal{I}}$ by first modelling the instance's constraints separately. For each variable X_i from $\mathcal{I}$, a root node X_i is introduced in the network, with the domain $\mathbf{D}$ for its values; X_i is associated with a uniform probability distribution $\Pr_{X_i}(X_i)$. For every constraint $\langle t, \mathbf{R} \rangle$ with $t = (X_j, X_k)$ from $\mathcal{I}$, we further add a new node R_i to the network, with TRUE and FALSE for its values and with X_j and X_k as its parents. For each joint value assignment $\mathbf{x}$ to X_j and X_k, we set the conditional probability distribution for R_i given $\mathbf{x}$ to

$$\Pr_{R_i}(R_i = \mathrm{TRUE} \,|\, \mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \mathbf{R} \\ 0 & \text{otherwise} \end{cases}$$

In the sequel, we will use the phrase 'relation node' when referring to such a node R_i. We will further use a tilde notation to indicate the intermediate Bayesian network and its graphical structures obtained so far. Figure 2 now shows the graph $\widetilde{\mathbf{G}}_{\mathrm{ex}}$ of the network $\widetilde{\mathcal{B}}_{\mathrm{ex}}$ which is thus far constructed from our running example $\mathcal{I}_{\mathrm{ex}}$.

With respect to treewidth, we observe that the relation nodes R_i are simplicial in the moralisation of the graph $\widetilde{\mathbf{G}}$ constructed so far, that is, in the moralised graph $\widetilde{\mathbf{G}}^{\mathbf{M}}$ they are adjacent to a complete set of nodes. Informally speaking, a node R_i and its two parents X_j and X_k from $\widetilde{\mathbf{G}}$ are joined in a clique in $\widetilde{\mathbf{G}}^{\mathbf{M}}$, which implies a treewidth of at least two of the moralised graph. More formally,

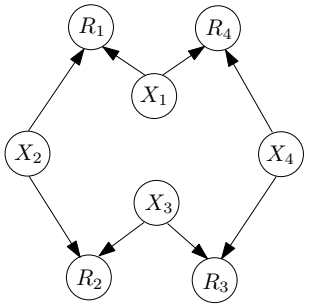

Figure 2. The graph $\widetilde{\mathbf{G}}_{\mathrm{ex}}$ constructed in the first step of the reduction of the constraint-satisfaction instance $\mathcal{I}_{\mathrm{ex}}$

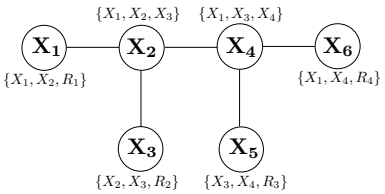

Figure 3. The tree-decomposition $\widetilde{\mathbf{T}}_{\mathrm{ex}}$ of the moralisation of the graph $\widetilde{\mathbf{G}}_{\mathrm{ex}}$ constructed so far from the instance $\mathcal{I}_{\mathrm{ex}}$

from the theory of treewidth (see for example [15]), we know that for any simplicial node V with degree d in an undirected graph $\mathbf{G}$, the treewidth of $\mathbf{G}$ equals the maximum of d and the treewidth of $\mathbf{G}$ minus V. Using this result, we have that the moralisation of the graph $\widetilde{\mathbf{G}}$ constructed so far from the constraint-satisfaction instance $\mathcal{I}$ has a treewidth of $\max(2, \mathrm{tw}(\mathbf{G}_{\mathcal{I}}))$, where $\mathbf{G}_{\mathcal{I}}$ is the primal graph of the instance $\mathcal{I}$. The first step of the construction can thus have increased the treewidth of the original instance by at most 1.

So far the various constraints from the constraint-satisfaction instance $\mathcal{I}$ have been modelled separately in the Bayesian network under construction. A solution to $\mathcal{I}$ has to satisfy all constraints simultaneously, however. This requirement will be incorporated in the intermediate network $\widetilde{\mathcal{B}}$ constructed so far by joining the nodes representing the separate constraints by extra nodes mimicking the 'and'-operator. Note that modelling the 'and' has to be done with care to avoid an exponential blow-up of the treewidth of the network's moralised graph. Such a blow-up would typically occur if we were to add a single designated node A_1 with all relation nodes R_i for its parents; it would even occur if we were to construct a log-deep binary tree to connect the relation nodes to A_1.

To mimic the 'and'-operator without blowing up treewidth, we will exploit the structure of a specific tree-decomposition of the moralised graph $\widetilde{\mathbf{G}}^{\mathbf{M}}$ obtained so far. The basic idea is that by using this decomposition, we can monitor the treewidth when adding nodes and arcs to the graph $\widetilde{\mathbf{G}}$. For this purpose, we will use a tree-decomposition $\widetilde{\mathbf{T}}$ of $\widetilde{\mathbf{G}}^{\mathbf{M}}$ such that $\widetilde{\mathbf{T}}$ is a rooted tree and every node in $\widetilde{\mathbf{T}}$ has at most two children. In the proof of our result, we will assume, for ease of exposition, that the tree-decomposition $\widetilde{\mathbf{T}}$ used in the construction is a nice decomposition. In our running example, however, we will use a non-nice decomposition meeting the two requirements mentioned above, simply because a nice tree-decomposition would take too much space.

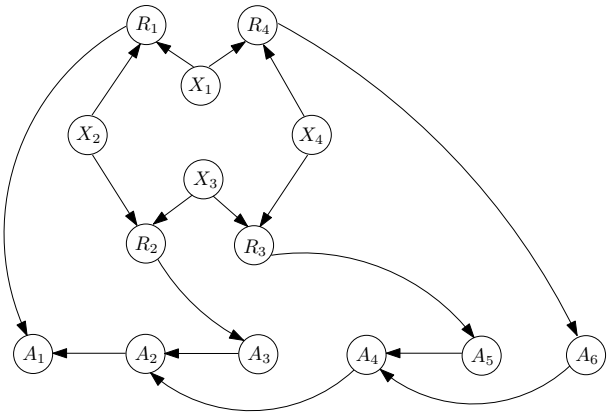

Figure 4. The graph $\mathbf{G}_{\mathrm{ex}}$ which results for the Bayesian network $\mathcal{B}_{\mathrm{ex}}$ under construction after adding nodes A_i and appropriate arcs to $\widetilde{\mathbf{G}}_{\mathrm{ex}}$

Now, let $\widetilde{\mathbf{T}}$ be a nice tree-decomposition of the moralised graph $\widetilde{\mathbf{G}}^{\mathbf{M}}$ obtained so far, and let m be the number of nodes in $\widetilde{\mathbf{T}}$. Note that each node in $\widetilde{\mathbf{T}}$ is a bag $\mathbf{X}_k$ of nodes R_i and X_j from the originally constructed graph $\widetilde{\mathbf{G}}$. Based upon this decomposition, we will now add new nodes $A_1, \ldots, A_m$ and arcs (R_i, A_k) to the network under construction. For each node $\mathbf{X}_k$ in $\widetilde{\mathbf{T}}$, we add a node A_k to $\widetilde{\mathbf{G}}$ and, for every arc $(\mathbf{X}_k, \mathbf{X}_l)$ from $\widetilde{\mathbf{T}}$, we add an arc (A_l, A_k); we further add an arc (R_i, A_k) for each relation node R_i in the bag $\mathbf{X}_k$. For every newly added node A_i, $i = 1, \ldots, m$, we set, for each joint value assignment $\mathbf{x}$ to its parents, the conditional probability distribution given $\mathbf{x}$ to

$$\Pr_{A_i}(A_i = \textsc{true} \,|\, \mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \bigwedge_{V \in \pi(A_i)} (V = \textsc{true}) \\ 0 & \text{otherwise} \end{cases}$$

For a node A_i without any parents, we set $\Pr_{A_i}(A_i = \textsc{true}) = 1$. Note that the conditional probability distributions for a node A_i correspond to the logical 'and' of the values of its parents $\pi(A_i)$; in the sequel, we will therefore sometimes use the phrase 'and node' to refer such a node A_i. From the above construction, we now have that all relation nodes R_i from the originally constructed graph $\widetilde{\mathbf{G}}$ are chained together into A_1 and that $\Pr(A_1 = \textsc{true}) > 0$ if $\Pr(\bigwedge_i (R_i = \textsc{true})) > 0$.

For our running example, Figure 3 shows an appropriate (yet non-nice) tree-decomposition $\widetilde{\mathbf{T}}_{\mathrm{ex}}$ of the moralisation of the graph $\widetilde{\mathbf{G}}_{\mathrm{ex}}$ constructed from $\mathcal{I}_{\mathrm{ex}}$ so far. We assume that node $\mathbf{X}_1$ is the root of the tree. Since the decomposition includes six nodes, we add six new nodes $A_1, \ldots, A_6$ to $\widetilde{\mathbf{G}}_{\mathrm{ex}}$. For the first two nodes $\mathbf{X}_1$ and $\mathbf{X}_2$ from $\widetilde{\mathbf{T}}_{\mathrm{ex}}$, we add the nodes A_1 and A_2 to the graph under construction, along with the arcs (A_2, A_1) and (R_1, A_1); the first of these arcs is added because $\mathbf{X}_1$ is the parent of $\mathbf{X}_2$ in the tree $\widetilde{\mathbf{T}}_{\mathrm{ex}}$, and the second arc is added because the relation node R_1 is included in the bag $\mathbf{X}_1$. For the consecutive nodes $\mathbf{X}_i$, $i = 3, \ldots, 6$, from $\widetilde{\mathbf{T}}_{\mathrm{ex}}$, we further 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). After adding appropriate arcs from the relation nodes R_i to the and-nodes A_j, the graph $\mathbf{G}_{\mathrm{ex}}$ from Figure 4 results. This graph now is the graph of the Bayesian network $\mathcal{B}_{\mathrm{ex}}$ constructed from the constraint-satisfaction instance $\mathcal{I}_{\mathrm{ex}}$.

To allow investigation of the treewidth of the constructed Bayesian network $\mathcal{B}_{\mathcal{I}}$, we complete our reduction by constructing a tree-decomposition $\mathbf{T}_{\mathcal{B}_{\mathcal{I}}}$ of the moralised graph $\mathbf{G}^{\mathbf{M}}_{\mathcal{B}_{\mathcal{I}}}$ of the network.

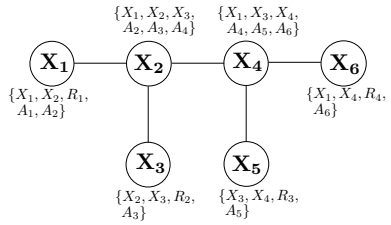

Figure 5. The tree-decomposition $\mathbf{T}_{\text{ex}}$ of the moralisation of the graph $\mathbf{G}_{\text{ex}}$ constructed from the constraint-satisfaction instance $\mathcal{I}_{\text{ex}}$

We will then use this decomposition to show that the moralisation of $\mathbf{G}_{\mathcal{B}_\mathcal{I}}$ has the same treewidth, up to a constant term, as the primal graph of the original constraint-satisfaction instance $\mathcal{I}$. The tree-decomposition $\mathbf{T}_{\mathcal{B}_\mathcal{I}}$ used for this purpose is obtained from $\widetilde{\mathbf{T}}$ by adding to each node $\mathbf{X}_k$ the and-node A_k and the nodes A_l that are contained in the children of $\mathbf{X}_k$ in $\widetilde{\mathbf{T}}$. It is readily verified that the thus constructed tree $\mathbf{T}_{\mathcal{B}_\mathcal{I}}$ indeed is a tree-decomposition of the moralised graph $\mathbf{G}^{\mathbf{M}}_{\mathcal{B}_\mathcal{I}}$. The tree-decomposition $\mathbf{T}_{\text{ex}}$ that is thus obtained for our running example, is shown in Figure 5.

The following theorem now states that the constraint-satisfaction problem tw-reduces to the problem of positive inference in Bayesian networks. More specifically, we will show that for any instance $\mathcal{I}$ of the constraint-satisfaction problem, we can construct a probabilistic network $\mathcal{B}_\mathcal{I}$ as described above such that in this network we have that $\Pr(A_1 = \text{TRUE}) > 0$ if and only if the instance $\mathcal{I}$ is satisfiable; we further have that the treewidth of the moralised graph of the constructed network $\mathcal{B}_\mathcal{I}$ is equal, up to a constant term, to the treewidth of the primal graph of the constraint-satisfaction instance $\mathcal{I}$.

Theorem 1 CONSTRAINT SATISFACTION *tw-reduces to* POSITIVE INFERENCE.

Proof. To show that the construction described above indeed gives a polynomial-time treewidth-preserving reduction f from CONSTRAINT SATISFACTION to POSITIVE INFERENCE, we need to show that the construction maps instances x of CONSTRAINT SATISFACTION to instances $f(x)$ of POSITIVE INFERENCE, such that $f(x)$ is a *yes*-instance of POSITIVE INFERENCE if and only if x is a *yes*-instance of CONSTRAINT SATISFACTION; we further have to show that the construction is computable in polynomial time and preserves the treewidth of an instance up to a constant term.

Let $\mathcal{I} = (\mathbf{V}, \mathbf{D}, \mathbf{C})$ be an instance of CONSTRAINT SATISFACTION, and let $\mathcal{P} = (\mathcal{B}_\mathcal{I}, A_1, \text{TRUE}, \varnothing, \top)$ be the instance of POSITIVE INFERENCE that is constructed from $\mathcal{I}$ as described above; note that since the inference instance does not include any evidence variables, its evidence is set to universal truth. Now suppose that the instance $\mathcal{P}$ is a *yes*-instance of POSITIVE INFERENCE, that is, suppose that in the network $\mathcal{B}_\mathcal{I}$ we have that $\Pr(A_1 = \text{TRUE}) > 0$. The 'and'-construct modelled with the nodes A_j then guarantees that $\Pr(\bigwedge_i (R_i = \text{TRUE})) > 0$. We now observe that, for any relation node R_i with the parent nodes X_j and X_k, we have that $\Pr(R_i = \text{TRUE} \,|\, \mathbf{x}) = 1$ for a joint value assignment $\mathbf{x}$ to $\{X_j, X_k\}$ if and only if $\mathbf{x}$ is included in the relation $\mathbf{R}$ from the constraint $\langle (X_j, X_k), \mathbf{R} \rangle$ in $\mathcal{I}$. So, from $\Pr(\bigwedge_i (R_i = \text{TRUE})) > 0$, we conclude that there must exist a joint value assignment to all constraint variables X_j that satisfies all constraints; hence, the instance $\mathcal{I}$ is a *yes*-instance of the constraint-satisfaction problem. Now suppose that there exists a satisfying assignment to the variables X_j from the constraint-satisfaction instance $\mathcal{I}$. Then, $\Pr(\bigwedge_i (R_i = \text{TRUE})) > 0$ and hence $\Pr(A_1 = \text{TRUE}) > 0$. We further observe that the construction described above can be carried out in polynomial time, since we introduce into the Bayesian network $\mathcal{B}_\mathcal{I}$ only a polynomial number of nodes for each variable from the instance $\mathcal{I}$.

Having shown that the constraint-satisfaction problem reduces to the problem of positive inference in Bayesian networks, we still have to show that the reduction f described above preserves the treewidth of an instance up to a constant term. We thus have to show that there exists a linear function l with $\text{tw}(f(x)) = l(\text{tw}(x))$ for every instance x of the constraint-satisfaction problem. We already argued above that the moralisation of the intermediate graph $\widetilde{\mathbf{G}}$ constructed in the reduction has a treewidth of $\max(2, \text{tw}(\mathbf{G}_\mathcal{I}))$ where $\text{tw}(\mathbf{G}_\mathcal{I})$ is the treewidth of the primal graph of the constraint-satisfaction instance $\mathcal{I}$. The first step in the construction thus can have increased the treewidth by at most 1. Note that this increase can have been effectuated only if the primal graph $\mathbf{G}_\mathcal{I}$ had a treewidth of 1, that is, only if the primal graph was a rooted tree. We now show that by adding the and-nodes A_j and their associated arcs in the second step of the construction of $\mathbf{G}_{\mathcal{B}_\mathcal{I}}$, the treewidth can have been increased by at most three. To facilitate our proof, we assume that the tree-decomposition $\widetilde{\mathbf{T}}$ that is used for the construction of $\mathbf{G}_{\mathcal{B}_\mathcal{I}}$, is nice; we would like to note, however, that the result can be proved for any rooted decomposition in which every node has at most two children. From the niceness assumption, we have that every node in $\widetilde{\mathbf{T}}$ is either a leaf node, an insert node, a forget node, or a join node. We consider each of these types of node separately:

- let $\mathbf{X}_i$ be a leaf node from the tree-decomposition $\widetilde{\mathbf{T}}$. Since this node has no children, just the and node A_i is added to the bag $\mathbf{X}_i$ in the construction of $\mathbf{T}_{\mathcal{B}_\mathcal{I}}$. By doing so, the width of the decomposition can have increased by at most 1.
- now let $\mathbf{X}_i$ be either an insert node or a forget node from $\widetilde{\mathbf{T}}$. Since this node has a single child $\mathbf{X}_j$, the nodes A_i and A_j are added to the bag $\mathbf{X}_i$ in the construction of $\mathbf{T}_{\mathcal{B}_\mathcal{I}}$. The width of the decomposition thus can have increased by 2 at the most.
- let $\mathbf{X}_i$ be a join node from $\widetilde{\mathbf{T}}$. Since this node has two children $\mathbf{X}_j$ and $\mathbf{X}_k$, in the construction of $\mathbf{T}_{\mathcal{B}_\mathcal{I}}$ the three and nodes A_i, A_j and A_k are added to the bag $\mathbf{X}_i$. Compared to $\widetilde{\mathbf{T}}$, the width of $\mathbf{T}_{\mathcal{B}_\mathcal{I}}$ can thus have increased by 3 at the most.

We conclude that the construction of the tree-decomposition $\mathbf{T}_{\mathcal{B}_\mathcal{I}}$ from $\widetilde{\mathbf{T}}$ can have increased treewidth by at most three. Now recall that we showed before that the first step of the reduction resulted in a graph whose moralisation had a treewidth of $\max(2, \text{tw}(\mathbf{G}_\mathcal{I}))$. Suppose that the intermediate graph $\widetilde{\mathbf{G}}$ resulting from the first step had a treewidth equal to $\text{tw}(\mathbf{G}_\mathcal{I})$. The above arguments now show that the graph $\mathbf{G}_{\mathcal{B}_\mathcal{I}}$ resulting after the second step can have a treewidth of at most $\text{tw}(\mathbf{G}_\mathcal{I}) + 3$. Now suppose that the graph $\widetilde{\mathbf{G}}$ had a treewidth equal to 2. Any node in the tree-decomposition $\widetilde{\mathbf{T}}$ whose bag includes three nodes from $\widetilde{\mathbf{G}}$ then is a leaf node. From this observation, we have that the treewidth of $\mathbf{G}_{\mathcal{B}_\mathcal{I}}$ is at most 4. We conclude that in either case the treewidth is increased by at most three. The above reduction thus preserves treewidth up to a constant term.

From the above considerations, we conclude that CONSTRAINT SATISFACTION tw-reduces to POSITIVE INFERENCE as stated in the theorem. □

The following theorem now states our main result, which is derived directly from the previous theorem and the result from Marx cited

before. Intuitively speaking, the theorem states that if an algorithm A exists that solves arbitrary instances of the inference problem with large treewidth in sub-exponential time, then we can construct an algorithm B solving instances of the constraint-satisfaction problem with large treewidth in sub-exponential time, which would contradict the Exponential Time Hypothesis.

Theorem 2 *If there exists a computable function f such that* POSITIVE INFERENCE *can be decided by an algorithm running in time*

$$f(\mathbf{G}_{\mathcal{B}}^{\mathbf{M}}) \cdot \|\mathcal{B}\|^{o(\frac{\mathrm{tw}(\mathbf{G}_{\mathcal{B}}^{\mathbf{M}})}{\log \mathrm{tw}(\mathbf{G}_{\mathcal{B}}^{\mathbf{M}})})}$$

for arbitrary instances $\mathcal{P} = (\mathcal{B}, C, c, \mathbf{E}, \mathbf{e})$ with a moralised graph $\mathbf{G}_{\mathcal{B}}^{\mathbf{M}}$ with treewidth $\mathrm{tw}(\mathbf{G}_{\mathcal{B}}^{\mathbf{M}})$, then the ETH *fails.*

Proof. We suppose that there exists an algorithm A that solves arbitrary instances $\mathcal{P}$ of the POSITIVE INFERENCE problem with unbounded treewidth in time

$$f(\mathbf{G}_{\mathcal{B}}^{\mathbf{M}}) \cdot \|\mathcal{B}\|^{o(\frac{\mathrm{tw}(\mathbf{G}_{\mathcal{B}}^{\mathbf{M}})}{\log \mathrm{tw}(\mathbf{G}_{\mathcal{B}}^{\mathbf{M}})})}$$

where f is a computable function and $\mathbf{G}_{\mathcal{B}}^{\mathbf{M}}$ denotes the moralised graph of $\mathcal{B}$. Now, let $\mathcal{I}$ be an instance of CONSTRAINT SATISFACTION whose primal graph $\mathbf{G}_{\mathcal{I}}$ has sufficiently large treewidth. From Theorem 1, we have that $\mathcal{I}$ can be reduced, in polynomial time, to an instance of POSITIVE INFERENCE with a network $\mathcal{B}_{\mathcal{I}}$ with a moralised graph $\mathbf{G}_{\mathcal{B}_{\mathcal{I}}}$ of treewidth $\mathrm{tw}(\mathbf{G}_{\mathcal{B}_{\mathcal{I}}}) \leq \mathrm{tw}(\mathbf{G}_{\mathcal{I}}) + 3$. Since we assumed that A solves the inference problem on the network $\mathcal{B}_{\mathcal{I}}$ in time

$$f(\mathbf{G}_{\mathcal{B}_{\mathcal{I}}}^{\mathbf{M}}) \cdot \|\mathcal{B}_{\mathcal{I}}\|^{o(\frac{\mathrm{tw}(\mathbf{G}_{\mathcal{B}_{\mathcal{I}}}^{\mathbf{M}})}{\log \mathrm{tw}(\mathbf{G}_{\mathcal{B}_{\mathcal{I}}}^{\mathbf{M}})})}$$

there exists a computable function g such that $\mathcal{I}$ can be solved in time

$$g(\mathbf{G}_{\mathcal{I}}) \cdot \|\mathcal{I}\|^{o(\frac{\mathrm{tw}(\mathbf{G}_{\mathcal{I}})}{\log \mathrm{tw}(\mathbf{G}_{\mathcal{I}})})}$$

By Marx' result reviewed in Section 3, this finding contradicts the ETH. □

5 CONCLUSIONS

Algorithms for probabilistic inference with arbitrary Bayesian networks all have a running time that is worst-case exponential in the size of the network. A well-known result from studies of the runtime properties of the commonly used junction-tree propagation algorithm, is that computing a posterior probability distribution for a variable in a network whose moralised graph has bounded treewidth, is exponential only in this treewidth. For networks of bounded size with small treewidth, therefore, inference can be feasibly performed. In this paper, we showed that small treewidth of a network's moralised graph is not just a sufficient but actually a necessary condition for a network to render probabilistic inference efficient by an algorithm accepting arbitrary instances. We showed, more specifically, that there cannot exist an algorithm solving arbitrary instances of the probabilistic-inference problem with large treewidth in polynomial time, unless the Exponential Time Hypothesis fails. We showed, in fact, that any algorithm solving arbitrary instances of the problem of probabilistic inference must have a running time of

$$f(\mathbf{G}^{\mathbf{M}}) \cdot \|\mathcal{B}\|^{\omega(\frac{\mathrm{tw}(\mathbf{G}^{\mathbf{M}})}{\log \mathrm{tw}(\mathbf{G}^{\mathbf{M}})})}$$

where $\mathcal{B}$ is the network at hand and $\mathbf{G}^{\mathbf{M}}$ is its moralised graph. Even in the absence of evidence any such algorithm will take exponential time in the treewidth of the moralised graph, up to a logarithmic factor in the exponent.

To conclude, we would like to note that our result for the problem of probabilistic inference is weaker than Marx' result for the constraint-satisfaction and graph-homomorphism problems, which provides a lower bound on the running time for algorithms solving these problems on *any* recursively enumerable class of graphs: while Marx' result thus holds also for restricted classes of graphs, our result still allows algorithms to use specific structural properties of a network, such as particular arc configurations or planarity properties of the moralised graph, to arrive at sub-exponential running times. Whether such properties can indeed be identified or whether our result can be extended to hold for *any* instance of the inference problem remains an open question for now.

REFERENCES

[1] J. Pearl (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, Palo Alto.

[2] R. Dechter (1999). Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, vol. 113, pp. 41 – 85.

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

[4] G.F. Cooper (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, vol. 42, pp. 393 – 405.

[5] M.L. Littman, S.M. Majercik, T. Pitassi (2001). Stochastic Boolean satisfiability. *International Journal of Automated Reasoning*, vol. 27, pp. 251 – 296.

[6] D. Roth (1996). On the hardness of approximate reasoning. *Artificial Intelligence*, vol. 82, pp. 273 – 302.

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

[8] V. Chandrasekaran, N. Srebro, P. Harsha (2008). Complexity of inference in graphical models. In: *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, AUAI Press, pp. 70 – 78.

[9] D. Marx (2007). Can you beat treewidth ? *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Press, pp. 169 – 179.

[10] T. Kloks (1994). *Treewidth. Computations and Approximations*, Lecture Notes in Computer Science, vol. 842, Springer-Verlag, Berlin.

[11] H.L. Bodlaender (1997). Treewidth: Algorithmic techniques and results. *Proceedings of the Twenty-second International Symposium on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, vol. 1295, Springer-Verlag, Berlin, pp. 19 – 36.

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

[13] C.H. Papadimitriou (1994). *Computational Complexity*. Addison-Wesley.

[14] R. Impagliazzo, R. Paturi (2001). On the complexity of k-SAT. *Journal of Computer and System Sciences*, pp. 62, pp. 367 – 375.

[15] H.L. Bodlaender (2006). Treewidth: characterizations, applications, and computations. In: F.V. Fomin (editor). *Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science, vol. 4271, Springer, New York, pp. 1 – 14.