RUTCOR
RESEARCH
REPORT

On complexity of the acyclic hypergraph sandwich problem [a]

Georg Gottlob [b] Vladimir Gurvich [c]
Zoltán Miklós [d]

RRR 11-2006, April 2006

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/∼rrr

[a]This paper was supported by the Austrian Science Fund (FWF) project: *Nr. P17222-N04, Complementary Approaches to Constraint Satisfaction*. The research of the second author was partially supported by the National Science Foundation, Grant IIS-0118635

[b]Vienna University of Technology, Favoritenstr. 9, A-1040, Vienna, Austria email: gottlob@dbai.tuwien.ac.at

[c]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854-8003; e-mail: gurvich@rutcor.rutgers.edu.

[d]Vienna University of Technology, Favoritenstr. 9, A-1040, Vienna, Austria e-mail: miklos@dbai.tuwien.ac.at

Rutcor Research Report
RRR 11-2006, April 2006

On complexity of the acyclic hypergraph sandwich problem [1]

Georg Gottlob Vladimir Gurvich Zoltán Miklós

Abstract. Given two hypergraphs H and H' on a common vertex set, we write $H < H'$ if each edge of H is contained in an edge of H'. Given $H < H'$, either find an acyclic hypergraph A between them, $H < A < H'$, or claim that there is no such A. This problem is referred to as the Acyclic Hypergraph Sandwich Problem (AHSP) (H, H'). We show that one can assume without loss of generality that H is a graph. The AHSP (H, H') generalizes the concept of treewidth as follows. Let $H = G$ be a graph, $|V(G)| = n$, and let $H' = \binom{n}{k}$ consists of all subsets of $V(G)$ of cardinality k. Then the AHSP is solvable if and only if the treewidth of G is strictly less than k, that is $TW(G) \leq k - 1$. Another important special case of the AHSP is $H' = H^k$, that is the edges of H' are the unions of all subfamilies of k edges of H. In this case the AHSP generalizes the hypertreewidth of H. It was recently proved [0] that the ASHP is NP-complete already in case $H' = H^3$. However, it is known that verifying $TW(G) \leq k - 1$ is polynomial when k is bounded. Respectively, the AHSP (H, H') is polynomial when $H' = \binom{n}{k}$. Here we extend this result and show that the AHSP can be solved in time, $t = n^{(d+1)(\log n + d + 1)}$, where $n = |V(H)|$ and $d = dimH'$ is the maximum edge size (so-called dimension) of H'. In particular, t is quasi-polynomial in n whenever d is bounded or polylogarithmic in n. Hence, in this case the AHSP is not NP-complete unless every problem from NP can be solved in quasi-polynomial time. In particular, the AHSP (H, H^k) is quasi-polynomial, $t = n^{(kd+1)(\log n + kd + 1)}$, whenever both k and $dimH$ are bounded or polylogarithmic in n.
Key words: acyclic hypergraph, chordal graph, hypertreewidth, quasi-polynomial problem, sandwich problem, treewidth, triangulated graph

1 Introduction

In [0] one can find 12 equivalent definitions of the acyclic hypergraphs. We will need 2 of these 12.

Given a hypergraph H with the vertex set $V = V(H)$ and edge set set $E = E(H)$, introduce two operations $O1$ and $O2$ as follows:

$O1$. If $e \subseteq e'$ for a pair of edges $e, e' \in E(H)$ then delete e from $E(H)$.

$O2$. If $deg_H(v) = 1$ for a vertex $v \in V(H)$ then delete v from each edge which contains it.

Recall that $deg_H(v)$, the *degree* of v in H, is the number of edges in H which contain v.

It is not difficult to see that applying recursively $O1$ and $O2$ to H we get a unique (well defined) irreducible hypergraph H_O. In other words, though in some cases we can delete several vertices and/or edges, yet, our choice does not matter and the resulting hypergraph H_O is always the same. The proof can be found in [0, 0], where the above recursive procedure reducing H to H_O is attributed to Marc H. Graham.

Definition A. A hypergraph $A = (V, E)$ is *acyclic* if applying recursively $O1$ and $O2$ we can reduce A to nothing, that is A_O is the empty hypergraph.

Acyclic hypergraphs admit the following convenient representation. Given a tree T_0 and a family $\mathcal{T}$ of its subtrees, define a hypergraph $A = A(T_0, \mathcal{T})$ as follows: $V(A) = \mathcal{T}$, that is to each subtree $T \in \mathcal{T}$ we assign a vertex of A, while $E(A) = V(T_0)$, that is to each vertex $v \in T_0$ we assign an edge $e = e_v$ of A such that e consists of all subtrees $T \in \mathcal{T}$ which contain v.

Definition B. A hypergraph $A = (V, E)$ is *acyclic* if $A = A(T_0, \mathcal{T})$ for some T_0 and $\mathcal{T}$.

It is shown in [0] that definitions A and B are equivalent.

Given two hypergraphs H and H' we will write $H < H'$ if for every edge e of H there is an edge e' of H' such that $e \subseteq e'$. Obviously, this relation is transitive. Yet, it is possible that $H < H'$ and $H' < H$. Clearly, it happens if and only if applying $O1$ we obtain the same irreducible hypergraph from H and H'. In this case we will call H and H' *equivalent*.

Acyclic Hypergraph Sandwich Problem (AHSP) is formulated as follows. Given two hypergraphs H and H', either find an acyclic hypergraph A such that $H < A < H'$ or claim that there is no such A. The input of the AHSP is the pair of hypergraphs (H, H'). Obviously, one can assume that $H < H'$, since otherwise the AHSP has no solution for sure. It is also clear that the AHSP belongs to NP.

Remark 1. *The sandwich problems for graphs and hypergraphs are surveyed in [0, 0, 0]. Let us note, however, that the definitions are slightly different. In the cited papers it is assumed that three hypergraphs H, H' and A have the same number of edges which are labelled by the same set of indices $[m] = \{1, \ldots, m\}$ and $e_i(H) \subseteq e_i(A) \subseteq e_i(H')$ for each $i \in [m]$.*

Computing the treewidth [0] is reduced to the AHSP as follows. Let $H = G$ be an arbitrary graph, that is $|e| = 2$ for each $e \in E(H)$, and let $H' = \binom{n}{k}$, that is the edges of

$E(H')$ are all subsets of cardinality k of the given vertex set $V(H') = V(G)$ of cardinality n. Then the AHSP is solvable if and only if the treewidth of the graph G is strictly less than k, that is $TW(G) \leq k-1$. See surveys [0, 0] for this and many other equivalent characterizations of the treewidth.

In section **??** we will show that an arbitrary AHSP (H, H') is polynomially reduced to the case when $H = G$ is a graph.

Given G and k, it is NP-hard to decide whether $TW(G) \leq k-1$, [0]. However, this does not imply that the corresponding AHSP is NP-hard, since the hypergraph $H' = \binom{n}{k}$ may be exponential if k is a part of the input. Yet, if k is bounded then the inequality $TW(G) \leq k-1$ can be verified in polynomial time. The first polynomial algorithm was given in [0]; its performance time was n to a function of k. Then the time was successively reduced to $\emptyset(n^{k+2})$ in [0], $\emptyset(n^2)$ in [0], $\emptyset(n \log^2 n)$ in [0, 0], $\emptyset(n \log n)$ in [0], and finally to $\emptyset(n)$ in [0]. For $k \leq 5$ there are special, simpler, methods; see [0] section 3.3 or [0] section 3 for a survey.

Respectively, the AHSP (H, H') is polynomial when $H' = \binom{n}{k}$. In section **??** we will extend this result as follows. The *dimension* of a hypergraph is defined as the maximum of its edge sizes, $dimH = \max_{e \in E(H)} |e|$. For example, $\dim \binom{n}{k} = k$.

Theorem 1. *The AHSP can be solved in time $t = n^{(d+1)(\log n + d + 1)}$, where $n = |V(H)|$ and $d = dimH'$.*

In particular, the ASHP is quasi-polynomial in n when d is bounded, or polylogarithmic in n, that is $d \leq \log^c n$, where c is a constant. Hence, in this case the AHSP is not NP-complete unless every problem from NP can be solved in quasi-polynomial time.

Another important special case of the AHSP is $H' = H^k$, that is $E(H')$ consists of the unions of all subfamilies of k edges from H. More precisely, $e \in E(H')$ if and only if $e = \cup_{i=1}^k e_i$, for some $e_1, \ldots, e_k \in E(H)$. In this case the AHSP is related to the generalized hypertreewidth [0, 0] It was recently proven [0] that the AHSP is NP-complete already in case $H' = H^3$. However, it is very unlikely that the AHSP (H, H^k) is NP-complete if k and $d = dimH$ are both bounded or polylogarithmic in n. Indeed, in this case Theorem **??** immediately implies

Corollary 1. *The AHSP (H, H^k) can be solved in time $n^{(kd+1)(\log n + kd + 1)}$.*

Remark 2. *Given an AHSP (H, H'), where $H' = H^k$ or $H' = \binom{n}{k}$, both H and H' or respectively H and k may be considered as the input. If k is bounded then these two possible inputs are polynomially equivalent, yet in general, the size of the first one may be exponential in size of the second one.*

2 Reformulation in terms of chordal graphs and its corollaries

A graph is called *chordal* (or *triangulated*) if each its simple cycle of length at least 4 has a chord.

Proposition 1. *A graph is chordal if and only if each its 2-connected component is chordal.*

Proof. . By definition, each simple cycle is contained in a 2-connected component and has no common edges with other components. □

Given a hypergraph H, its *intersection* graph $G = G^I(H)$ is defined as follows: the vertices of G are the edges of H and two vertices form an edge in G if and only if the corresponding edges of H intersect. The following characterization of the chordal graphs will play an important role. Given a tree T_0 and a family $\mathcal{T}$ of its subtrees, denote by $G^I(T_0, \mathcal{T})$ their intersection graph.

Theorem 2. *([0, 0, 0, 0]) Graph G is chordal if and only if $G = G^I(T_0, \mathcal{T})$ for some $T_0, \mathcal{T}$.*

Proposition 2. *Subtrees of T_0 have the Helly property, that is all subtrees of a subfamily $\mathcal{T}' \subseteq \mathcal{T}$ intersect whenever each two of them intersect.*

Proof. is obvious and well-known. Assume that three subtrees of T_0 pairwise intersect but have no common vertex. Then they form a cycle, but T_0 is a tree, so we get a contradiction. □

By Theorem **??**, the AHSP (H, H') can be reformulated as follows. Given a graph G and a hypergraph H' (which may have common vertices), either extend G to a chordal graph G' every (maximal) clique of which is contained in an edge of H' or claim that there are no such extensions. It suffice to set $G = G^I(H)$ to prove the equivalence of the two definitions. In particular, $TW(G) \leq k - 1$ if and only if G is a subgraph of a chordal graph with maximal clique-size k, [0].

Let us show that for an AHSP (H, H') it can be assumed without loss of generality that H is a (2-connected) graph.

Given a hypergraph H, its *co-occurrence* (or *primal*) graph $G(H)$ is defined as follows. The vertex set is the same, $V(G(H)) = V(H) = V$ and two vertices form an edge in $G(H)$ if and only if they both belong to an edge of H. Obviously, the edges of H form cliques in $G(H)$ and these cliques contain all edges of $G(H)$. The hypergraph H is called *normal* if every maximal clique of $G(H)$ is an edge of H. Note that several normal hypergraphs may have the same co-occurrence graph G, yet clearly, all these hypergraphs are equivalent. From now on let us restrict ourselves by the hypergraphs irreducible with respect to $O1$. Then each class of equivalent hypergraphs is of cardinality 1 and $<$ is a partial order relation.

Given a graph G, denote by $\mathcal{H} = \mathcal{H}(G)$ the set of all hypergraphs whose co-occurrence graph is G, that is $G = G(H)$ if and only if $H \in \mathcal{H}(G)$. It is easy to see that $\mathcal{H}$ is a lattice with respect to $<$. The minimum of this lattice is the graph G itself and the maximum is the normal hypergraph $H^0(G)$.

Let us remark that the family $\mathcal{H}(G)$, as well as some hypergraphs $H \in \mathcal{H}(G)$, including $H^0(G)$, may be exponential in $n = |V(G)|$, since G may have exponentially many (maximal) cliques.

The next 3 claims follow immediately from Theorem **??** and Proposition **??**.

Corollary 2. *Let A be an acyclic hypergraph such that $A > G$, then $A > H$ for each $H \in \mathcal{H}(G)$. In particular, $A > H^0(G)$, that is every (maximal) clique of G is contained in an edge of A.*

Corollary 3. *Given G and H', for $H \in \mathcal{H}(G)$ all AHSPs (H, H') are equivalent. In particular, they are all equivalent to the AHSP $(G(H), H')$.*

Thus, without loss of generality, it can be assumes that H is a graph for any AHSP (H, H'). Indeed, given an AHSP (H, H'), one can just substitute H by its co-occurrence graph $G(H)$. Moreover, according to Proposition **??**, it also can be assumed that the graph $G(H)$ is 2-connected, otherwise the AHSP can be considered for each 2-connected component separately.

In addition we get one more characterization of acyclic hypergraphs and chordal graphs.

Corollary 4. *Graph G is chordal if and only if its normal hypergraph $H^0(G)$ is acyclic. Hypergraph H is acyclic if and only if it is normal and its co-ocurrence graph $G(H)$ is chordal.*

Remark 3. *It is interesting to compare characterizations of the acyclic and read-once hypergraphs: H is read-once (respectively, acyclic) if and only if H is normal and its co-occurrence graph $G(H)$ is P_4-free [0, 0, 0] (respectively, C_i-free for all $i \geq 4$). Hence, a hypergraph H is read-once and acyclic if and only if H is normal and $G(H)$ is P_4- and C_4-free. Some applications of P_4-free chordal graphs can be found in [0, 0, 0].*

Given a hypergraph H, the *neighborhood* $N_H(v)$ of a vertex $v \in V(H)$ is defined as the union of all edges of H which contain v, that is $N_H(v) = \bigcup_{e \mid v \in e \in E(H)} e$. Let us remark that operation $O1$ keeps all neighborhoods unchanged.

Proposition 3. *Given a graph G and a hypergraph H which may have vertices in common, the equality $N_H(v) = N_G(v)$ holds for all $v \in V(G) \cup V(H)$ if and only if $H \in \mathcal{H}(G)$.*

Proof. follows immediately from the definitions of N and $\mathcal{H}(G)$. □

The next result of [0] shows that the AHSP (G, H') is closely related to the following difficult decision problem.

Triangulating of Colored Graphs (TCG): Given a graph G properly colored by n colors, either extend G to a chordal graph respecting the coloring, or prove that there is no such extension.

In [0] it is proved that TCG is NP-complete already in the case when each chromatic component in G is of cardinality 2. This result can be reformulated as follows.

Theorem 3. *([0]) Given a graph G with $2n$ vertices $V(G) = \{1, 1', \ldots, n, n'\}$, it is NP-hard to decide whether G can be extended to a chordal graph G' such that $(i, i') \in E(G')$ for no $i \in \{1, \ldots, n\}$.*

However, TCG becomes polynomial if n is bounded, [0] section 5.

By Theorems **??**, Theorem **??** can be reformulated in the AHSP terms as follows. Let us consider the perfect matching on the vertex set $V = \{1, 1', \ldots, n, n'\}$ and let G^0 be the complementary graph, that is $V(G^0) = V$ and $(i, j), (i, j'), (i', j') \in E(G^0)$ if and only if $i \neq j$. Furthermore, let $H' = H^0(G^0)$ be the corresponding normal hypergraph, that is $V(H') = V$ and $E(H')$ consists of 2^n edges of cardinality n each, $e \in E(H')$ if and only if e contains exactly one vertex from each pair i, i'.

Theorem 4. *The AHSP (G, H') is NP-complete with respect to the input G, provided $H' = H^0(G^0)$ is fixed.*

Proof. follows immediately from Theorems **??** and **??**. □

Of course, formally this AHSP is polynomial with respect to the standard input (G, H'), since the size of H' is exponential in n. However, H' is invariable, so it seems more logical to treat it as a fixed parameter rather than a part of the input.

Let us recall that if $H' = \binom{n}{k}$ and k is bounded then the AHSP (H, H') is polynomial and in this case the size of H' is polynomial in n, see remark **??**

3 More about chordal graphs

Proposition 4. *([0], see also [0]) Graph G is chordal if and only if there is a tree T whose vertices are in one-to-one correspondence with all maximal cliques of G and for every vertex $v \in V(G)$ all maximal cliques of G which contain v induce a subtree in T.*

Such a tree $T = T(G)$ is called a *tree representation* of G.

Though in general, the number of maximal cliques of a graph G may be exponential in $|V(G)|$, yet, if G is chordal then the number of its maximal cliques is at most the number of its vertices and equality holds if and only if $E(G) = \emptyset$, [0]. From this, by Corollary **??**, we derive that $|E(H)| \leq |V(H)|$ for every acyclic hypergraph H which is irreducible with respect to $O1$.

Given a connected graph G, a set of vertices $S \subseteq V(G)$ is called a *separator* if the graph $G[V \setminus S]$ induced by the remaining vertices is not connected. A separator is called *minimal* if no its proper subset is a separator. Let us recall that the chordal graphs were introduced in [0] where the following characterization in terms of separators was given.

Proposition 5. *([0]) A connected graph is chordal if and only if every its minimal separator is a clique.*

In general, the number of minimal separators of a connected graph G may be exponential in $|V(G)|$, yet, if G is a connected chordal graph then the number of its minimal separators is at most $|V(G)| - 1$. Moreover, they all are explicitly given by a tree representation of G.

Proposition 6. *([0]) Let G be a connected chordal graph and $T = T(G)$ be its tree representation, a subset $S \subseteq V(G)$ is a minimal separator if and only if $S = C' \cap C''$, where C'*

Figure 1: Example 1

Figure 2: Example 2

and C'' are two maximal cliques of G such that the corresponding vertices are neighbors in T.

Thus any tree representation contains complete information about all maximal cliques and minimal separators of a connected chordal graph.

4 Proof of Theorem ??

First, we will prove Theorem **??** for a very special case. However, we will see that the main lines of this proof appear in the general one, too. Let $H = C_n$ be the n-cycle, that is $V(C_n) = [n] = \{1, \dots, n\}$, $E(C_n) = \{(1,2), (2,3), \dots, (n-1, n), (n, 1)\}$ and let H' be an arbitrary family of m triangles, that is m subsets of $[n]$ of cardinality 3. In this case the AHSP can be solved in quasi-polynomial time $(3m)^{\log n}$ as follows.

Let us consider the polygon P_n and an arbitrary its triangulation. For example, $A_1 = \{(1,2,3), (3,4,5), (5,6,1), (1,3,5)\}$ and $A_2 = \{(1,2,3), (1,3,4), (1,4,5), (1,5,6)\}$ are two triangulations of P_6, see figures **??** and **??**.

It is easy to see that both hypergraphs A_1 and A_2 are acyclic and majorize C_n, that is $C_n < A_1, C_n < A_2$. It is well-known that in general, too, each minimal (with respect to ¡) acyclic hypergraph $A > C_n$ is a triangulations of P_n, and vice versa.

Remark 4. *Let us also mention that each triangulation of P_n consists of $n' = n-2$ triangles and there are $c_{n'}$ different triangulations of P_n, where $c_{n'} = \binom{2n'}{n'}/(n'+1)$ are so-called Catalan numbers.*

In our two examples on figures **??** and **??** the co-occurrence graphs $G_1 = G(A_1), G_2 = G(A_2)$ have common vertex set $V = V(G_1) = V(G_2) = \{1,2,3,4,5,6\}$ and the edge sets

$E(G_1) = \{(1,2), (2,3), (3,4), (4,5), (5,6), (6,1), (1,3), (3,5), (5,1)\}$;

$E(G_2) = \{(1,2), (2,3), (3,4), (4,5), (5,6), (6,1), (1,3), (1,4), (1,5)\}$.

Obviously, both graphs are chordal. Their maximal cliques are the edges of A_1 and A_2, respectively. The tree representation of G_2 is the simple path P_4 whose vertices correspond to the maximal cliques $(1,2,3), (1,3,4), (1,4,5)$, and $(1,5,6)$; while the tree representation of G_1 is the claw $K_{1,3}$ whose central vertex is $(1,3,5)$ and the leaves are $(1,2,3), (3,4,5)$, and $(5,6,1)$, see figures **??** and **??**.

Clearly, each triangle of P_n partitions it into 2, 3 or 4 smaller polygons. We will call a triangle *cental* if each of these polygons is of cardinality at most $n/2$. It is obvious that every triangulation of P_n contains a central triangle. For example, $(1,3,5)$ is a unique central triangle in A_1, while $(1,3,4)$ and $(1,4,5)$ are two central triangles of A_2. This simple observation results in the following quasi-polynomial recursive procedure for the AHSP (C_n, H'). If H' contains no central triangle of P_n then report that the AHSP has no solution and halt. Otherwise, for each cental triangle in H' reduce the problem to 2 or 3 smaller problems of size at most $n/2$ each. Obviously, the tree representing this recursion has at most $(3m)^{\log n}$ leaves. This proves the claim.

In the next paragraph we will show that for the AHSP (C_n, H') the "divide and conquer" technique considered above is less efficient than the standard dynamic programming approach. However, unlike the latter, the former one can be applied, and remains quasi-polynomial, for the general AHSP (H, H') whenever $dimH'$ is bounded.

For each ordered pair $i, j \in [n]$ let us define an entry of a dynamic programming table to be either 0 or 1 depending on whether the part P_n defined by the vertices $i, i+1, ..., j$ admits a triangulation. (All numbers taken $\mod n$). The first row of the table correpsonds to $(i, j) = (1, 3), (2, 4), \ldots, (n-2, n), (n-1, 1), (n, 2)$, that is $j = (i+2)(\mod n)$; the second one to $(i, j) = (1, 4), (2, 5), \ldots, (n-3, n), (n-2, 1)(n-1, 2)(n, 3)$, that is $j = (i+3)(\mod n)$; etc. Each new row l can be computed in lmn^2 time using H' and the entries in the previous rows. The AHSP (C_n, H') is solvable if and only if there exist i and j for which the entries for both pairs i, j and j, i are 1.

Proof. of Theorem **??**. Given an AHSP (H, H'), consider the equivalent AHSP (G, H'), where $G = G(H)$ is the co-occurrence graph of H. Also assume without loss of generality that G is 2-connected, otherwise consider separately the AHSPs (G_i, H') for all 2-connected components of G. We would like either to extend G to a chordal graph G' such that each (maximal) clique of G' is contained in an edge of H' or prove that no such G' exists. Let us assume that it exists and let $T = T(G')$ be a tree representation of it. We define now a decomposition of G' which will allow us later to get it from G. Recall that the vertices of T and the maximal cliques of G' are in one-to-one correspondence. Denote by $C(v)$ the clique corresponding to a vertex $v \in V(T)$ and let $c(v) = |C(v)|$. By assumption, each vertex set $C(v)$ is contained in an edge of H'. In particular, $c(v) \le d = dimH'$ for each $v \in V(T)$.

Let $v \in V(T)$ be a vertex of T which is not a leaf, that is $m = deg_T(v) > 1$. Then v is a separator in T and $C(v)$ is a separator in G'; furthermore, $V(T) \setminus \{v\}$ induces a forest which consists of m subtrees $T_1, \ldots, T_m$ of T. Denote by T'_i the subtree induced by the vertex set $V(T_i) \cup \{v\}$ and by G'_i the corresponding subgraph of G'; in other words, $V(G'_i) = \bigcup_{u \in V(T'_i)} C(u)$; $i \in [m] = \{1, \ldots, m\}$.

Set $n = |V(G')|$ and consider 3 cases: $n > 2d$, $2d \ge n > d$, and $n \le d$. In the last case we just stop the decomposition, otherwise we proceed as follows. Phase I: $n > 2d$.

Lemma 1. *If $n > 2d$ then there is a vertex $v \in V(T)$ such that $|V(G'_i)| \le n/2 + d$ for all $i \in [m]$.*

The corresponding separator $C(v)$ in G' will be called *central.*

Proof. is obvious. Indeed, if $n > 2d$ then $T(G')$ contains at least 3 vertices. Hence, there is a vertex v between them which is not a leaf. If G'_i contains more than $n/2 + d$ vertices for some $i \in [m]$ then let us just substitute v by its (unique) neighbor in T_i, etc. Clearly, after several such steps we get a vertex which satisfies the lemma. □

By this lemma, m graphs of size at most $n/2 + d$ substitute one graph of size n, where $2 \le m \le n$. Applying this reduction recursively $\lceil \log n \rceil$ times we get at most $n^{\lceil \log n \rceil}$ graphs with at most $2d$ vertices in each. Indeed, consider the sequjence n, $n/2+d$, $n/4+d/2+d$, $n/8+d/4+d/2+d, \ldots$ etc. Since $d\sum_{j=0}^{\infty} 2^{-j} = 2d$, each partial sum is strictly less than $2d$. In particular $d\sum_{j=0}^{\lceil \log n \rceil} 2^{-j} < 2d$. Also $n/2^{\lceil \log n \rceil} \leq 1$, since $n = 2^{\log n}$. However, if $n \leq 2d$ then $n/2+d \geq n$, so there is no reduction.

Phase II: $2d \geq n > d$. Yet, we will show thatin this case n is reduced by the same procedure, though not that fast. The original chordal graph G' with n vertices is reduced to m graphs G'_i with at most $n-1$ vertices in each. Indeed, $C(v)$ is a separator (and hence it reduces $|V(G')|$ by at least 1) if and only if the corresponding vertex $v \in V(T)$ is not a leaf in $T = T(G')$. Such a vertex v exists in T whenever T contains at least 3 vertices. If T contains only 2 vertices v' and v'' then $C(v') \cap C(v'')$ is the reducing separator. Applying this reduction at most d times we get at most $n^{\log d}$ graphs with at most d vertices in each.

Let us recall the original graph G. By definition, $V = V(G) = V(G')$, $E(G) \subseteq E(G')$ and hence, $S \subseteq V$ is a separator in G whenever S is a separator in G'. So we can apply the "divide and conquer" technique considered above to the AHSP (G, H'). More precisely, for each subset $S \subseteq V$ which is covered by an edge of H' let us check whether S is a central separator in G. In other words, we have to verify that $G[V \setminus S]$ contains at least two connected components and each of these components contains at most $n/2$ vertices in case $n > 2d$ (and at least one vertex in case $2d \geq n > d$, which is always true).

Obviously, the AHSP (G, H') is solvable whenever $|V(G)| = n \leq d$, since in this case G can be extended a chordal graph G' which is just the clique on $V(G)$.

The above recursive procedure can be represented by a tree for whose width W and depth D the following upper bounds hold: $W \leq n^{d+1}$ and $D \leq \log n + 1 + d$. The last inequality is already proven: reducing $n > 2d$ to $2d$ takes at most $\log n + 1$ steps (phase I), while reducing $n \leq 2d$ to d takes at most d steps (phase II). Let us show that $W \leq n^{d+1}$. Since $dim H' = d$, the "central separator" test should be applied only to sets S of cardinality at most d. The number s of such sets is bounded by n^d. Indeed,

$s < \sum_{j=0}^{d} \binom{n}{j} < (1/2)\sum_{j=0}^{d} n^j = (n^{d+1}-1)/(2(n-1)) < n^d$

whenever $n > 1$. From this $W \leq n^{d+1}$ follows, since by each separation m subgraphs substitute the original graph, where $2 \leq m \leq n$. Thus, the number N of leaves of the recursion tree is bounded by $N \leq W^D \leq n^{(d+1)(\log n + d + 1)}$.

The recursion either shows that the AHSP (G, H') has no solution or solves it. In the last case we just substitute a clique for each found separator and get a chordal graph G' such that G' contains G as a subgraph and each (maximal) clique of G' is contained in an edge of H'. In fact, the maximal cliques of G' are exactly the separators obtained by the recursion. Theorem **??** is proved. □

Can we strengthen Theorem **??** and get a polynomial algorithm ? We know that it is possible in case when $H = C_n$ and H' is a family of triangles. However, this method succeeds, because in case of C_n we can restrict ourselves by only $n^2 - 3n$ "natural" subsets of $V(C_n)$, which are defined by the $\binom{n}{2} - n = (n^2-3n)/2$ chords of C_n. For an arbitrary 2-connected graph G its vertex set contains 2^n subsets and we have no "natural" polynomial subfamily.

In case $H' = \binom{n}{k}$ the above algorithm requires at most $n^{(k+1)(\log n + k + 1)}$ time. Yet, in this case polynomial (and even linear) in n algorithms are known for bounded k [0, 0, 0, 0, 0, 0, 0]. However, all these algorithms are more sophisticated than the above recursive procedure.

If $H' = H^k$ then $dimH' \leq k\ dimH$. Hence, in this case the performance time is bounded by $n^{(kd+1)(\log n+kd+1)}$, where $n = |V(H)|$ and $d = dimH$. This time is quasi-polynomial in n provided d and k are both bounded or polylogarithmic in n.

Remark 5. *Let us finally note that the following "obvious" polynomial reduction of the AHSP (G, H') to its special case $(G, \binom{n}{k})$, where $d = dimH' < k < const$, fails. Consider the family $\mathcal{S}$ of all inclusion-minimal vertex sets $S \subseteq V(G)$ which are not contained in an edge of H'. Clearly, $|S| \leq d+1$ for each $S \in \mathcal{S}$, otherwise S would not be minimal. Hence, $|\mathcal{S}| \leq \binom{n}{d+1}$. Let us fix an integer $k > d$ and for every $S \in \mathcal{S}$ introduce a vertex set $V(S)$ which consists of $k+1-|S|$ new vertices. For every $S \in \mathcal{S}$ let us introduce an edge (u, v) for each $u, v \in V(S)$ and for each $u \in S, v \in V(S)$. Thus we obtain a graph G^+ such that (i) $|V(G^+) = |V(G)| + (k+1)|\mathcal{S}| - \sum_{S\in\mathcal{S}} |S|$, (ii) G is a subraph of G^+, (iii) for every $S \in \mathcal{S}$ the vertex set $V(S)$ induces a clique in G^+ and also G^+ contains all edges between S and $V(S)$. Now, it seems that the AHSPs (G, H') and $(G^+, \binom{n}{k})$ are equivalent. Indeed, let G' be a chordal graph such that G is its subgraph and $V(G') = V(G)$. By construction, G' has a clique which is not contained in an edge of H' if and only if G^+ has a clique of cardinality $k+1$. Moreover, every triangulation of G^+ induces a triangulation of G. Yet, not vice versa. It is not clear how to assign a triangulation of G' to a triangulation of G.*

Conjecture 1. *The AHSP (H, H') is polynomial if $dimH'$ is bounded.*

Acknowledgments The authors thank Nysret Musliu and Vladimir Udalov for encouraging computations, also Endre Boros and Khaled Elbassioni for helpful discussions.

References

[1] Richa Agarwala and David Fernandez-Baca, Fast and simple algorithms for perfect phylogeny and triangulating colored graphs, DIMACS Tech Report 94-51, 1994.

[2] Stefan Arnborg, D.G. Corneil, and A. Proskurowsky, Characterization and recognition of partial 3-trees, SIAM J. Alg. Disc. Meth. (1986) 7 : 305-314.

[3] Stefan Arnborg, Jens Lagergren, and Detlef Seese, Easy problems for tree decomposable graphs, J. Algorithms (1991) 12 : 308-340

[4] Catriel Beeri, Ronald Fagin, David Maier, and Mihalis Yannakakis, On the desirability of database schemes, Journal of the ACM 30(3) : 479-513,(1983).

[5] Hans L. Bodlaender, A linear time algorithm for finding tree-decompositions of small treewidth, SIAM J. Comput. (1996) 25 : 1305-1317.

[6] Hans L. Bodlaender, A partial k-arboretum of graphs with bounded treewidth, Theoretical Computer Science (1998) 209 : 1-45.

[7] Hans L. Bodlaender, Discovering treewidth, In the Proceedings of the 31st Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2005), LNCS, 3381.

[8] Hans L. Bodlaender, Mike R. Fellows, and Tandy J. Warnow, Two strikes against perfect phylogeny, In the Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP 1992), LNCS, 623 : 273-283.

[9] Hans Bodlaender and Ton Kloks, Better algorithms for path width and tree width of graphs, In the Proceedings of the 18th International Colloquium on Automata, Languages and Programming (1991) p. 544-555.

[10] P. Buneman, A characterization of rigid circuit graphs, Discrete Math. (1974) 9 : 205-212.

[11] R. Castelo and A. Siebes, A characterization of moral transitive directed acyclic graph Markov models as trees and its properties, Tech. Report, CS-2000-44, Univ. of Utrecht, 2000

[12] A. Dirac, On rigid circuit graph, Abhandlungen Mathematischer Seminare der Universitat Hamburg (1961) 25 : 71-76.

[13] Thomas Eiter, Exact transversal hypergraphs and application to Boolean μ-functions, J. Symbolic Computation (1994) 17 (3) : 215-225.

[14] Fanica Gavril, The intersection graphs of subtrees in trees are exactly the chordal graph, J. Comb. Theory Series B, (1974) 16 : 47-56

[15] Vibhav Gogate and Rina Dechter, A complete anytime algorithm for treewidth, In Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence, UAI-04 (2004) p. 201-208.

[16] Martin Charles Golumbic, Trivially perfect graphs, Discrete Math. (1978) 24 : 105-107

[17] Martin Charles Golumbic, Haim Kaplan, and Ron Shamir, Graph sandwich problems. Journal of Algorithms 19(3) 449-473. (1995).

[18] Martin Charles Golumbic and Amir Wasserman, Complexity and algorithms for graph and hypergraph sandwich problems, Graph and Combinatorics (1998) 14 : 223 239.

[19] Georg Gottlob, Nicola Leone, and Francesco Scarcello, Hypertree Decompositions and Tractable Queries, Journal of Computer and System Sciences (JCSS), 64(3) 579-627, (2002).

[20] Georg Gottlob, Nicola Leone, and Francesco Scarcello, Robbers, marshals, and guards: game theoretic and logical characterizations of hypertree width, Journal of Computer and System Sciences (JCSS), 66(4) 775-808, (2003).

[21] Vladimir Gurvich, On repetition-free Boolean functions, Russian Math. Surveys (Uspechi Mat. Nauk) 32 (1) (1977) : 183-184 (in Russian).

[22] Leonid Libkin and Vladimir Gurvich, Trees as semilattices, Discrete Math. (1995) : 145 : 321-327.

[23] C.W. Ho and R.C.T. Lee, Counting of clique-trees and computing perfect elimination schemes in parallel, Information Processing Letters (1989) 31 : 61 - 68.

[24] M. Karchmer, N. Linial, I. Newman, M. Saks, A. Wigderson, Combinatorial characterization of read-once formulae, Discrete Math. (1993) 114 (1-3) : 275-282

[25] Aviv Lustig and Oded Smueli, Acyclic hypergraph projections, Journal of Algorithms 30(2) 400-422,(1999).

[26] Bruce A. Reed, Finding approximate separators and computing tree width quickly, Proc. 24-th Annual Symposium on Theory of Computing, New York (1992) 221-228; Preprint (1991).

[27] Neil Robertson and Paul D. Seymour, Disjoint paths - a survey, SIAM J. Alg. Disc. Meth. (1985) 6 : 300-305.

[28] Neil Robertson and Paul D. Seymour, Graph minors. II. Algorithmic aspects of tree-width. J. Algorithms (1986) 7 : 309-322.

[29] Neil Robertson and Paul D. Seymour, Graph minors. XIII. The disjoint path problem, J. Comb. Theory, ser. B, (1995) 63 (1) : 65-110; Preprint (1986).

[30] Michael Steel, The complexity of the reconstructing trees from qualitative characters and subtrees, Journal of Classification (1992) 9 : 91-116.

[31] Thomas Schwentick, Georg Gottlob, and Zoltán Miklós, private communications, 2005.

[32] R.E. Tarjan and M. Yannakakis, Simple linear-time algorithm to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, SIAM J. Comput. (1984) 13 : 566-579.

[33] J.R. Walter, Representation of rigid cycle graphs, PhD Thesis (1972), Wayne State Univ., Detroit, Michigan.

[34] J.R. Walter, Representation of chordal graphs as subtrees of a tree, J. Graph Theory (1978) 2 : 265-267