Algorithmica (2007) 47: 27–51
DOI: 10.1007/s00453-006-1209-y

Algorithmica

Tree Spanners for Bipartite Graphs and Probe Interval Graphs[1]

Andreas Brandstädt,[2] Feodor F. Dragan,[3] Hoang-Oanh Le,[2]
Van Bang Le,[2] and Ryuhei Uehara[4]

Abstract. A tree t-spanner T in a graph G is a spanning tree of G such that the distance between every pair of vertices in T is at most t times their distance in G. The tree t-spanner problem asks whether a graph admits a tree t-spanner, given t. We first substantially strengthen the known results for bipartite graphs. We prove that the tree t-spanner problem is NP-complete even for chordal bipartite graphs for $t \geq 5$, and every bipartite ATE-free graph has a tree 3-spanner, which can be found in linear time. The previous best known results were NP-completeness for general bipartite graphs, and that every convex graph has a tree 3-spanner. We next focus on the tree t-spanner problem for probe interval graphs and related graph classes. The graph classes were introduced to deal with the physical mapping of DNA. From a graph theoretical point of view, the classes are natural generalizations of interval graphs. We show that these classes are tree 7-spanner admissible, and a tree 7-spanner can be constructed in $O(m \log n)$ time.

Key Words. Chordal bipartite graph, Interval bigraph, NP-completeness, Probe interval graph, Tree spanner.

1. Introduction. A *tree t-spanner T* in a graph G is a spanning tree of G such that the distance between every pair of vertices in T is at most t times their distance in G. The *tree t-spanner problem* asks whether a graph admits a tree t-spanner, given t. The notion is introduced by Cai and Corneil [1], [2], who found numerous applications in distributed systems and communication networks; for example, it was shown that tree spanners can be used as models for broadcast operations [3] (see also [4]). Moreover, tree spanners were used in the area of biology [5], and approximating the bandwidth of graphs [6]. We refer to [7]–[9] for more background information on tree spanners.

The tree t-spanner problem is NP-complete in general [2] for any $t \geq 4$. However, it can be solved efficiently for some particular graph classes. Especially, the complexity of the tree t-spanner problem is well investigated for the class of chordal graphs and its subclasses. For $t \geq 4$, the problem is NP-complete for chordal graphs [9], strongly chordal graphs are tree 4-spanner admissible [10] (i.e., every strongly chordal graph has

[1] An extended abstract of this paper appeared in the *Proceedings of the* 29*th Workshop on Graph Theoretic Concepts in Computer Science* (*WG* 2003), June 19–21, 2003, *Elspeet*, *The Netherlands*, pp. 106–118. Lecture Notes in Compter Science 2880, Springer, Berlin, 2003. The research of H.-O. Le was supported by DFG, Project No. Br1446-4/1. This work was done while R. Uehara was visiting the University of Waterloo.
[2] Institut für Theoretische Informatik, Fachbereich Informatik, Universität Rostock, 18051 Rostock, Germany. {ab,hoang-oanh.le,le}@informatik.uni-rostock.de.
[3] Department of Computer Science, Kent State University, Kent, OH 44242, USA. dragan@cs.kent.edu.
[4] School of Information Science, Japan Advanced Institute of Science and Technology, Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan.

Received June 17, 2003; revised August 22, 2005. Communicated by H. Gabow.
Online publication July 25, 2006.

a tree 4-spanner), and the following graph classes are tree 3-spanner admissible: interval graphs [11], directed path graphs [12], split graphs [6] (see also [9] for other known results).

We first focus on the tree t-spanner problem for bipartite graphs and its subclasses. The class of bipartite graphs is a wide and important class from both practical and theoretical points of view. However, the known results for the complexity of the tree t-spanner problem for bipartite graphs and their subclasses are few compared with the chordal graphs and their subclasses. NP-completeness is only known for general bipartite graphs (this result can be deduced from the construction in [2]), and the problem can be solved for regular bipartite graphs and convex graphs as follows: a regular bipartite graph is tree 3-spanner admissible if and only if it is complete [11]; and any convex graph is tree 3-spanner admissible [6]. (Convex graphs were introduced by Brandstädt et al. [13]; refer to Section 2 for a definition, and see the Appendix for further details.)

We substantially strengthen the known results for bipartite graph classes, and reduce the gap. We show that the tree t-spanner problem is NP-complete even for chordal bipartite graphs for $t \geq 5$. The class of chordal bipartite graphs is a bipartite analog of chordal graphs, introduced by Golumbic and Goss [14], and has applications to nonsymmetric matrices (see [15]). We also show that every bipartite asteroidal-triple-edge-free (ATE-free) graph has a tree 3-spanner, and such a tree spanner can be found in linear time. The class of ATE-free graphs was introduced by Müller [16] to characterize interval bigraphs. The class of interval bigraphs is a bipartite analog of interval graphs and was introduced by Harary et al. [17].

Our results reduce the gap between the upper and lower bounds of the complexity of the tree t-spanner problem for bipartite graph classes since the following proper inclusions are known [16], [18]:

$$\begin{aligned} \text{convex graphs} &\subset \text{interval bigraphs} \subset \text{bipartite ATE-free graphs} \\ &\subset \text{chordal bipartite graphs} \subset \text{bipartite graphs.} \end{aligned}$$

We next focus on the tree t-spanner problem on probe interval graphs and related graph classes. The class of probe interval graphs was introduced by Zhang to deal with the physical mapping of DNA, which is a problem arising in the sequencing of DNA (see [19]–[22] for the background). A probe interval graph is obtained from an interval graph by designating a subset P of vertices as *probes*, and removing the edges between pairs of vertices in the remaining set N of *nonprobes*. In the original papers [19], [22], Zhang introduced two variations of probe interval graphs. An enhanced probe interval graph is the graph obtained from a probe interval graph by adding the edges joining two nonprobes if they are adjacent to two independent probes. The class of STS-probe interval graphs is a subset of the probe interval graphs; in those graphs all probes are independent.

From the graph theoretical point of view, it has been shown that all probe interval graphs are weakly chordal [20], and enhanced probe interval graphs are chordal [19], [22]. In the Appendix we show that (1) the class of STS-probe interval graphs is equivalent to the class of convex graphs (hence the class is tree 3-spanner admissible), and (2) the class of the (enhanced) probe interval graphs is incomparable with the classes of strongly chordal graphs and rooted directed path graphs. We also mention that, for any given probe interval graph, the graph obtained by removing all edges joining probe vertices is an interval bigraph.

Hence, from both viewpoints of graph theory and biology, the tree t-spanner problem for (enhanced) probe interval graphs is worth investigating. Especially, it is natural to ask if those graph classes are tree t-spanner admissible for fixed integer t. We give the positive answer to that question: The classes of probe interval graphs and enhanced probe interval graphs are tree 7-spanner admissible. A tree 7-spanner of a (enhanced) probe interval graph can be constructed in $O(m + n \log n)$ time if it is given with an interval model. Recently, Johnson and Spinrad showed that the recognition problem for the class of probe interval graphs can be solved in $O(n^2)$ time if each vertex is given with information whether it is probe or nonprobe [23], and the time complexity was improved to $O(m \log n)$ time by McConnell and Spinrad [24]. Those recognition algorithms also construct within the same time bounds an intersection model of a probe interval graph. Therefore, using their algorithms, we can construct a tree 7-spanner for a given (enhanced) probe interval graph $G = (P, N, E)$ in $O(m \log n)$ time.

2. Preliminaries. Given a graph $G = (V, E)$ and a subset $U \subseteq V$, the *subgraph of* G *induced by* U is the graph (U, F), where $F = \{\{u, v\} | \{u, v\} \in E \text{ for } u, v \in U\}$, and is denoted by $G[U]$. For a subset F of E, we sometimes unify the edge set F and its *edge induced subgraph* (U, F) with $U = \{v | \{u, v\} \in F \text{ for some } u \in V\}$. A sequence of the vertices $v_0, v_1, \ldots, v_l$ is a *path*, denoted by $(v_0, v_1, \ldots, v_l)$, if $\{v_j, v_{j+1}\} \in E$ for each $0 \leq j \leq l - 1$. The *length* of a path is the number of edges on the path. For two vertices u and v on G, the *distance* of the vertices is the minimum length of the paths joining u and v, and is denoted by $d_G(u, v)$. A *cycle* is a path beginning and ending with the same vertex.

The *disk* of radius k centered at v is the set of all vertices with distance at most k to v,

$$D_k(v) = \{w \in V\colon d_G(v, w) \leq k\},$$

and the kth *neighborhood* $N_k(v)$ of v is defined as the set of all vertices at distance k to v, that is

$$N_k(v) = \{w \in V\colon d_G(v, w) = k\}.$$

By $N(v)$ we denote the *neighborhood* of v, i.e., $N(v) := N_1(v)$. More generally, for a subset $S \subseteq V$ let $N(S) = \bigcup_{v \in S} N(v)$ denote the *neighborhood* of S. (We note that $S \cap N(S)$ may be nonempty.)

A connected acyclic edge set is called a *tree*. A tree joining all vertices is called a *spanning tree*. A *tree t-spanner* T in a graph G is a spanning tree of G such that for each pair u and v in G, $d_T(u, v) \leq t \cdot d_G(u, v)$. We say that G is *tree t-spanner admissible* if it contains a tree t-spanner. The *tree t-spanner problem* is to determine, for given graph and positive integer t, if the graph admits a tree t-spanner. A class C of graphs is said to be *tree t-spanner admissible* if every graph in C is tree t-spanner admissible. On the tree t-spanner problem, the following result plays an important role:

LEMMA 1 [2]. *A spanning tree T of G is a tree t-spanner if and only if for every edge $\{u, v\}$ of G, $d_T(u, v) \leq t$.*

A graph $G = (V, E)$ is *bipartite* if V can be divided into two sets V_1 and V_2 with $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$ such that every edge joins a vertex in V_1 and another one in V_2. It is well known that a graph G is bipartite if and only if G contains no cycle of odd length [25]. Thus, for each positive integer k, a tree $2k$-spanner of a bipartite graph G is also a tree $(2k-1)$-spanner. Hence we consider a tree t-spanner for each odd number t for bipartite graphs in this paper.

Here we define the graph classes dealt with in this paper. See the Appendix and [18] and [21] for further details and references.

Interval graphs and related classes. A graph (V, E) with $V = \{v_1, v_2, \ldots, v_n\}$ is an *interval graph* if there is a set of intervals $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ such that $\{v_i, v_j\} \in E$ if and only if $I_i \cap I_j \neq \emptyset$ for each i and j with $1 \leq i, j \leq n$. We call the set $\mathcal{I}$ the *interval representation* of the graph. For each interval I, we denote by $R(I)$ and $L(I)$ the right and left endpoints of the interval, respectively (hence we have $L(I) \leq R(I)$). A bipartite graph (X, Y, E) with $X = \{x_1, x_2, \ldots, x_{n_1}\}$ and $Y = \{y_1, y_2, \ldots, y_{n_2}\}$ is an *interval bigraph* if there are families of intervals $\mathcal{I}_X = \{I_1, I_2, \ldots, I_{n_1}\}$ and $\mathcal{I}_Y = \{J_1, J_2, \ldots, J_{n_2}\}$ such that $\{x_i, y_j\} \in E$ if and only if $I_i \cap J_j \neq \emptyset$ for each i and j with $1 \leq i \leq n_1$ and $1 \leq j \leq n_2$. We also call the families of intervals $(\mathcal{I}_X, \mathcal{I}_Y)$ *interval representation* of the graph. We sometimes unify a vertex v_i and its corresponding interval I_i; I_v denotes the interval corresponding to the vertex v, and $R(v)$ and $L(v)$ denote $R(I_v)$ and $L(I_v)$, respectively.

Chordal graphs and related classes. An edge which joins two vertices of a cycle but is not itself an edge of the cycle is a *chord* of that cycle. A graph is *chordal* if each cycle of length at least 4 has a chord. A graph G is *weakly chordal* if G and $\bar{G}$ contain no induced cycle C_k with $k \geq 5$. A bipartite graph G is *chordal bipartite* if each cycle of length at least 6 has a chord. Let the neighborhood $N(e)$ of an edge $e = \{v, w\}$ be the union $N(v) \cup N(w)$ of the neighborhoods of the end-vertices of e. Three edges of a graph G form an *asteroidal triple of edges* (ATE) if for any two of them there is a path from the vertex set from one to the vertex set of the other that avoids the neighborhood of the third edge. *Asteroidal-triple-edge-free* (*ATE-free*) graphs are those graphs which do not contain any ATE. This class of graphs was introduced in [16], where it was also shown that any interval bigraph is an ATE-free graph, and any bipartite ATE-free graph is chordal bipartite. For a bipartite graph (X, Y, E), an ordering $<$ of X has the *adjacency property* if for each vertex $y \in Y$, $N(y)$ consists of vertices that are consecutive (an interval) in the ordering $<$ of X. A bipartite graph is *convex* if there is an ordering of X or Y that fulfills the adjacency property [13].

Probe interval graphs and related classes. A graph $G = (V, E)$ is a *probe interval graph* if V can be partitioned into subsets P and N (corresponding to the *probes* and *nonprobes*) and each $v \in V$ can be assigned to an interval I_v such that $\{u, v\} \in E$ if and only if both $I_u \cap I_v \neq \emptyset$ and at least one of u and v is in P. In this paper we assume that P and N are given, and we denote the considered probe interval graph by $G = (P, N, E)$. Note that N is an independent set, $G[P]$ is an interval graph, and $G[P \cup \{v\}]$ is also an interval graph for any $v \in N$. Let $G = (P, N, E)$ be a probe interval graph. Let E^+ be a set of edges $\{u_1, u_2\}$ with $u_1, u_2 \in N$ such that there are two probes v_1 and v_2 in P such that $\{v_1, u_1\} \in E, \{v_1, u_2\} \in E, \{v_2, u_1\} \in E, \{v_2, u_2\} \in E$, and $\{v_1, v_2\} \notin E$. Intuitively, nonprobes u_1 and u_2 are joined by the edge in E^+ if (1) there are two independent probes v_1 and v_2, and (2) both v_1 and v_2 overlap u_1 and u_2. In the case we know that intervals I_{u_1} and I_{u_2} have to overlap in any affirmative interval representations. Each edge in E^+ is called an *enhanced edge*, and the resulting graph $G^+ = (P, N, E \cup E^+)$ is said to be an *enhanced probe interval graph*. See [19]–[22] for further details.

3. NP-Completeness for Chordal Bipartite Graphs. In this section we show that, for any $t \geq 5$, the tree t-spanner problem is NP-complete for chordal bipartite graphs. The proof is a reduction from Monotone 3SAT which consists of instances of 3SAT such that each clause contains either only negated variables or only non-negated variables (see [LO2] of [26]), for which the following family of chordal bipartite graphs will play an important role.

First, $S_0[a, b]$ is an edge $\{a, b\}$, and $S_1[a, b]$ is the 4-cycle (a, b, b', a', a). Next, for a fixed integer $\ell > 1$, $S_{\ell+1}[a, b]$ is obtained from one cycle (a, b, b', a', a), $S_\ell[a, a']$, $S_\ell[b, b']$, and $S_\ell[a', b']$ by identifying the corresponding vertices (see Figure 1).

We connect the vertices a and b to other graphs, and use $S_\ell[a, b]$ as a subgraph of bigger graphs. Sometimes, when the context is clear, we simply write S_ℓ for $S_\ell[a, b]$. In case $\ell > 0$ we write (a, a', b', b, a) for the 4-cycle in $S_\ell[a, b]$ containing the edge $\{a, b\}$. Each of the edges $\{a, a'\}$, $\{a', b'\}$, $\{b, b'\}$ belongs to a unique $S_{\ell-1}$, the *corresponding* $S_{\ell-1}$ in $S_\ell[a, b]$ to $\{a, a'\}$, $\{a', b'\}$, $\{b, b'\}$, respectively.

The following observations collect basic facts on S_ℓ used in the reduction later.

OBSERVATION 2. *For every integer $\ell \geq 0$, $S_\ell[a, b]$ has a tree $(2\ell + 1)$-spanner containing the edge $\{a, b\}$.*

PROOF. By induction on ℓ. The case $\ell = 0$ is clear. Let $\ell > 0$, and let (a, a', b', b, a) be the 4-cycle in $S_\ell[a, b]$ containing the edge $\{a, b\}$. Let L, M, R be the corresponding $S_{\ell-1}$ containing the edge $\{a, a'\}$, $\{a', b'\}$, $\{b, b'\}$, respectively. By the induction hypothesis, each of L, M, R has a tree $(2\ell - 1)$-spanner T_L, T_M, T_R containing the edge $\{a, a'\}$, $\{a', b'\}$, $\{b, b'\}$, respectively.

Let $T_M^{a'}, T_M^{b'}$ be the connected components of $T_M - \{a', b'\}$ with $a' \in T_M^{a'}$ and $b' \in T_M^{b'}$. Then $T_L \cup T_M^{a'}$ and $T_R \cup T_M^{b'}$ are two disjoint trees and $T := (T_L \cup T_M^{a'}) \cup (T_R \cup T_M^{b'}) \cup \{a, b\}$ is a spanning tree of $S_\ell[a, b]$. Moreover, T is a tree $(2\ell + 1)$-spanner of $S_\ell[a, b]$. To see this we need only consider edges $\{x, y\} \in M$ such that $x \in T_M^{a'}$ and $y \in T_M^{b'}$. For such edges we have: The (x, y)-path in T consists of the (x, a')-path in $T_M^{a'}$, the (y, b')-path

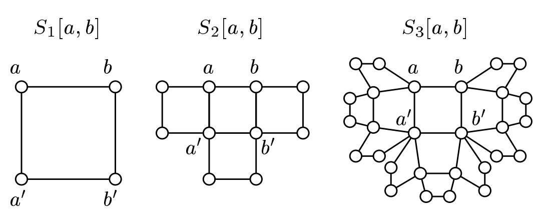

Fig. 1. The graph $S_\ell[a, b]$.

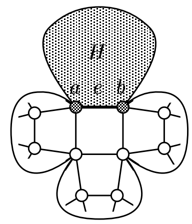

Fig. 2. The graph obtained from H and $S_\ell[a, b]$ by identifying the edge $e = \{a, b\}$.

in $T_M^{b'}$, and the edges $\{a', a\}$, $\{a, b\}$, $\{b, b'\}$. Therefore,

$$d_T(x, y) = d_{T_M}(x, y) - 1 + 3,$$

hence, as T_M is a tree $(2\ell - 1)$-spanner in M,

$$d_T(x, y) \le (2\ell - 1) - 1 + 3 = 2\ell + 1. \qquad \square$$

OBSERVATION 3. *Let H be an arbitrary graph and let e be an arbitrary edge of H. Let K be an $S_\ell[a, b]$ disjoint from H. Let G be the graph obtained from H and K by identifying the edges e and $\{a, b\}$; see Figure 2. Suppose that T is a tree t-spanner in G, $t > 2\ell$, such that the (a, b)-path in T belongs to H. Then $d_T(a, b) \le t - 2\ell$.*

PROOF. By induction on ℓ. For $\ell = 0$, the statement follows directly from the fact that T is a tree t-spanner of G. Let $\ell > 0$, and suppose inductively that the statement is true for arbitrary H and $S_{\ell-1}$.

Let (a, a', b', b, a) be the 4-cycle in K containing the edge $\{a, b\}$, and let L, M, R be the corresponding $S_{\ell-1}$ in K containing the edge $\{a, a'\}$, $\{a', b'\}$, $\{b', b\}$, respectively.

Let P be the (a, b)-path in T. By assumption, $P \subseteq H$. Consider the (a, a')-path Q in T. We distinguish two cases.

Case 1: $Q \not\subseteq L$. In this case, by definition of G, Q belongs to $H \cup R \cup M$ and P is a subpath of Q. The induction hypothesis applied to $H' := H \cup R \cup M$ and L yields

$$d_T(a, a') \le t - 2(\ell - 1),$$

hence

$$d_T(a, b) = d_T(a, a') - d_T(b, a') \le t - 2(\ell - 1) - 2 = t - 2\ell.$$

Case 1 is settled.

Case 2: $Q \subseteq L$. Let P' be the (a', b')-path in T. If $P' \subseteq M$ then $P \cup Q \cup P'$ is the (b, b')-path in T. The induction hypothesis applied to $H' := H \cup M \cup L$ and R yields

$$d_T(b, b') \le t - 2(\ell - 1),$$

hence

$$d_T(a, b) = d_T(a, a') - d_T(b, a') \le t - 2(\ell - 1) - 2 = t - 2\ell.$$

If $P' \not\subseteq M$ then $P' \subseteq L \cup H \cup R$ and $Q \cup P$ is a subpath of P'. The induction hypothesis applied to $H' := H \cup L \cup R$ and M yields

$$d_T(a', b') \leq t - 2(\ell - 1),$$

hence

$$d_T(a, b) = d_T(a', b') - d_T(a', a) - d_T(b', b) \leq t - 2(\ell - 1) - 1 - 1 = t - 2\ell.$$

In either case we are done. □

Observation 3 indicates a way to force an edge $\{x, y\}$ to be a tree edge for given odd t: choosing $\ell = (t-1)/2$ shows that $\{a, b\}$ must be an edge of any tree t-spanner T.

We now describe the reduction. Let $k \geq 2$ be an integer, and let F be a 3SAT formula with m clauses C_j for $1 \leq j \leq m$, over n variables x_i for $1 \leq i \leq n$. We construct a chordal bipartite graph G from F such that G has a tree $(2k+1)$-spanner if and only if F is satisfiable.

DEFINITION 4. In a graph G, an edge $\{a, b\}$ is said to be *forced by an S_ℓ* if G is obtained from two distinct graphs $S_\ell[a, b]$ and the rest by identifying the edges $\{a, b\}$ in $S_\ell[a, b]$ and an edge in the rest. We require that each two $S_\ell[a, b]$ and $S_{\ell'}[c, d]$ have at most two vertices in $\{a, b, c, d\}$ in common. An edge $\{a, b\}$ is said to be *strongly forced* if it is forced by two $S_k[a, b]$.

Hereafter, we omit "by two $S_k[a, b]$" for each strongly forced edge since it is always forced by two $S_k[a, b]$ for the fixed k.

By Observation 3, if G has a tree $(2k+1)$-spanner T every strongly forced edge must belong to T.

For each variable x_i create the gadget $G(x_i)$ as follows:

- Take $2m + 4$ vertices $x_i^1, \ldots, x_i^m, \overline{x_i}^1, \ldots, \overline{x_i}^m, p_i, q_i, r_i, s_i$,
- and add the edges $\{x_i^j, \overline{x_i}^{j'}\}$ for $1 \leq j, j' \leq m$, $\{q_i, x_i^j\}$ for $1 \leq j \leq m$, $\{r_i, x_i^j\}$ for $1 \leq j \leq m$, $\{p_i, \overline{x_i}^j\}$ for $1 \leq j \leq m$, $\{s_i, \overline{x_i}^j\}$ for $1 \leq j \leq m$, and $\{p_i, r_i\}$, $\{r_i, s_i\}$, $\{s_i, q_i\}$.

Furthermore,

- each of the edges $\{p_i, r_i\}$, $\{r_i, s_i\}$, $\{s_i, q_i\}$, and $\{x_i^j, \overline{x_i}^j\}$ with $1 \leq j \leq m$, is a strongly forced edge,
- force each edge $\{a, b\} \in \{\{q_i, x_i^j\}\colon 1 \leq j \leq m\} \cup \{\{r_i, x_i^j\}\colon 1 \leq j \leq m\} \cup \{\{p_i, \overline{x_i}^j\}\colon 1 \leq j \leq m\} \cup \{\{s_i, \overline{x_i}^j\}\colon 1 \leq j \leq m\} \cup \{\{x_i^j, \overline{x_i}^{j'}\}\colon 1 \leq j, j' \leq m,\ j \neq j'\}$ by an $S_{k-1}[a, b]$.

Thus, the subgraph in $G(x_i)$ induced by the two independent sets $\{x_i^1, \ldots, x_i^m\} \cup \{p_i, s_i\}$ and $\{\overline{x_i}^1, \ldots, \overline{x_i}^m\} \cup \{q_i, r_i\}$ *plus* the edge $\{p_i, q_i\}$ is a complete bipartite graph (see Figure 3; in the figure the S_k and S_{k-1} are omitted, and thick edges are strongly forced).

The vertex x_i^j ($\overline{x_i}^j$, respectively) will be connected to the clause gadget of clause C_j if x_i ($\overline{x_i}$, respectively) is a literal in C_j. All edges $\{r_i, x_i^j\}$ $(1 \leq j \leq m)$ or else all edges $\{s_i, \overline{x_i}^j\}$ $(1 \leq j \leq m)$ will belong to any tree $(2k+1)$-spanner (if any) of the graph G which we are going to describe.

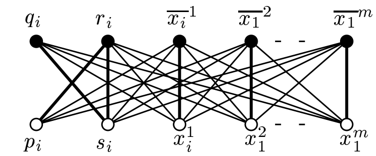

Fig. 3. The gadget $G(x_i)$.

DEFINITION 5. A clause is *positive* (*negative*, respectively) if it contains only variables (negation of variables).

We note that each clause is either positive or negative since the given formula is an instance of Monotone 3SAT. For each clause C_j, $G(C_j)$ is the 4-cycle $(c_j^+, d_j^+, d_j^-, c_j^-, c_j^+)$ where $\{c_j^+, d_j^+\}$, $\{d_j^+, d_j^-\}$, and $\{d_j^-, c_j^-\}$ are strongly forced edges (see Figure 4). Finally, the graph $G = G(F)$ is obtained from all $G(v_i)$ and $G(C_j)$ by identifying all vertices p_i, q_i, r_i, and s_i to a single vertex p, q, r, and s, respectively (thus, $\{p, r\}$, $\{r, s\}$ and $\{s, q\}$ are edges in G), and adding the following additional edges:

- Connect every x_i^j with every $\overline{x_{i'}}^{j'}$ $(i \neq i')$. (Thus, the subgraph induced by the two independent sets $\{x_i^j \colon 1 \leq i \leq n, 1 \leq j \leq m\} \cup \{p, s\}$ and $\{\overline{x_i}^j \colon 1 \leq i \leq n, 1 \leq j \leq m\} \cup \{q, r\}$ *plus* the edge $\{p, q\}$ is a complete bipartite graph.)
- For every positive clause C_j: If x_i is in C_j then connect x_i^j with c_j^+ and force the edge $\{x_i^j, c_j^+\}$ by an $S_{k-2}[x_i^j, c_j^+]$. Connect c_j^- with r and force the edge $\{c_j^-, r\}$ by an $S_{k-2}[c_j^-, r]$.
- For every negative clause C_j: If $\overline{x_i}$ is in C_j then connect $\overline{x_i}^j$ with c_j^- and force the edge $\{\overline{x_i}^j, c_j^-\}$ by an $S_{k-2}[\overline{x_i}^j, c_j^-]$. Connect c_j^+ with s and force the edge $\{c_j^+, s\}$ by an $S_{k-2}[c_j^+, s]$.

The description of the graph $G = G(F)$ is complete. Clearly, G can be constructed in polynomial time. See Figure 5 for an example.

LEMMA 6. *G is chordal bipartite.*

PROOF. First note that each S_ℓ is a chordal bipartite graph. By construction,

$$\{x_i^j \colon 1 \leq i \leq n, 1 \leq j \leq m\} \cup \{p, s\} \cup \{c_j^- \colon 1 \leq j \leq m\} \cup \{d_j^+ \colon 1 \leq j \leq m\}$$

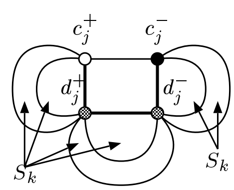

Fig. 4. The gadget $G(C_j)$.

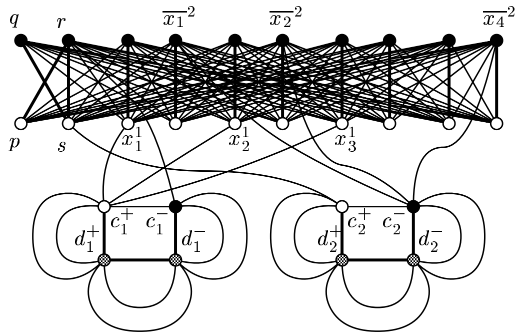

Fig. 5. The reduction given $C_1 = (x_1, x_2, x_3)$ and $C_2 = (\overline{x_1}, \overline{x_2}, \overline{x_4})$.

and

$$\{\overline{x_i}^j\colon\ 1 \le i \le n, 1 \le j \le m\} \cup \{q, r\} \cup \{c_j^+\colon\ 1 \le j \le m\} \cup \{d_j^-\colon\ 1 \le j \le m\}$$

are independent sets. This partition can be extended in a natural way to a bipartition of $V(G)$ into two independent sets. So, G is bipartite.

Next, let G' be the subgraph of G induced by

$$A := \{x_i^j\colon\ 1 \le i \le n, 1 \le j \le m\} \cup \{p, s\}$$

and

$$B := \{\overline{x_i}^j\colon\ 1 \le i \le n, 1 \le j \le m\} \cup \{q, r\}.$$

Since $G' + \{\{p, q\}\}$ is a complete bipartite graph with the bipartition (A, B),

$$G' \text{ is a chordal bipartite graph.}$$

On the other hand, since it is the disjoint union of the clause gadgets,

$$G - G' \text{ is a chordal bipartite graph.}$$

We consider an induced cycle Z in G containing vertices from both G' and $G - G'$. By construction, $Z \cap (G - G') \subseteq \{c_j^+, c_j^-\colon\ 1 \le j \le m\}$.

Let C_j be a positive clause. If $c_j^- \in Z$ then (r, c_j^-, c_j^+) must be a subpath of Z, therefore $Z = (r, c_j^-, c_j^+, x_i^j, r)$ for some i. If $c_j^+ \in Z$ (and $c_j^- \notin Z$) then, for some i_1, i_2, $(x_{i_1}^j, c_j^+, x_{i_2}^j)$ is a subpath of Z. Since c_j^+ is the neighbor outside G' of $x_{i_1}^j$ and of $x_{i_2}^j$, $Z = (v, x_{i_1}^j, c_j^+, x_{i_2}^j, v)$ for a vertex $v \in \{q, r\} \cup \{\overline{x_i}^j\colon\ 1 \le i \le n\}$.

Similarly, Z is a 4-cycle if C_j is a negative clause.

Thus, G is chordal bipartite as claimed. □

LEMMA 7. *Suppose G admits a tree $(2k+1)$-spanner. Then F is satisfiable.*

PROOF. Let T be a tree $(2k+1)$-spanner of G. By construction of G and Observation 3, the following edges of G belong to T:

- $\{p, r\}$, $\{r, s\}$, $\{s, q\}$, and $\{x_i^j, \overline{x_i}^j\}$ for $1 \le i \le n$, $1 \le j \le m$,
- $\{c_j^+, d_j^+\}$, $\{d_j^+, d_j^-\}$, and $\{d_j^-, c_j^-\}$ for $1 \le j \le m$.

CLAIM 1. *For every i and j, $\{q, x_i^j\} \notin E(T)$ and $\{p, \overline{x_i}^j\} \notin E(T)$.*

PROOF OF CLAIM 1. If, for some i, j, $\{q, x_i^j\} \in E(T)$ then $(p, r, s, q, x_i^j, \overline{x_i}^j)$ is the $(p, \overline{x_i}^j)$-path in T, hence $d_T(p, \overline{x_i}^j) = 5$. However, by Observation 3, $d_T(p, \overline{x_i}^j) \le (2k+1) - 2(k-1) = 3$, a contradiction. By symmetry, we have $\{p, \overline{x_i}^j\} \notin E(T)$. □

CLAIM 2. *For every i and j, exactly one of $\{r, x_i^j\}$ and $\{s, \overline{x_i}^j\}$ belongs to T.*

PROOF OF CLAIM 2. Both edges $\{r, x_i^j\}$ and $\{s, \overline{x_i}^j\}$ cannot belong to T, otherwise they would form, together with $\{r, s\}$ and $\{x_i^j, \overline{x_i}^j\}$, a cycle in T.

Now, assume to the contrary that, for some i, j, neither $\{r, x_i^j\}$ nor $\{s, \overline{x_i}^j\}$ belongs to T. Then by Observation 3, $d_T(r, x_i^j) = 3$ and $d_T(s, \overline{x_i}^j) = 3$. Note that by Claim 1, $\{q, x_i^j\} \notin E(T)$ and $\{p, \overline{x_i}^j\} \notin E(T)$. Hence by Observation 3, $d_T(q, x_i^j) = 3$ and $d_T(p, \overline{x_i}^j) = 3$, too. Let P be the (r, x_i^j)-path in T.

If P contains $\{r, s\}$ and $\{x_i^j, \overline{x_i}^j\}$ then clearly $d_T(r, x_i^j) \ge 5$, a contradiction.

If P contains $\{r, s\}$ but not $\{x_i^j, \overline{x_i}^j\}$ then write $P = (r, s, v, x_i^j)$. By assumption, $v \ne \overline{x_i}^j$, and as s and p are nonadjacent, $v \ne p$. Thus, $(p, r, s, v, x_i^j, \overline{x_i}^j)$ is the $(p, \overline{x_i}^j)$-path in T, hence $d_T(p, \overline{x_i}^j) = 5$, a contradiction.

If P contains $\{x_i^j, \overline{x_i}^j\}$ but not $\{r, s\}$ then write $P = (r, v, \overline{x_i}^j, x_i^j)$. By assumption, $v \ne s$, and as q and r are nonadjacent, $v \ne q$. Thus $(q, s, r, v, \overline{x_i}^j, x_i^j)$ is the (q, x_i^j)-path in T, hence $d_T(q, x_i^j) = 5$, a contradiction.

If P does not contain $\{x_i^j, \overline{x_i}^j\}$ and $\{r, s\}$ then write $P = (r, u, v, x_i^j)$. In this case, $u, v \notin \{s, \overline{x_i}^j\}$. Thus $(s, r, u, v, x_i^j, \overline{x_i}^j)$ is the $(s, \overline{x_i}^j)$-path in T, hence $d_T(s, \overline{x_i}^j) = 5$, a contradiction. □

CLAIM 3. *For each i, either all edges $\{r, x_i^j\}$, with $1 \le j \le m$, belong to T, or all edges $\{s, \overline{x_i}^j\}$, with $1 \le j \le m$, belong to T.*

PROOF OF CLAIM 3. Assume to the contrary that there exist $j_1 \ne j_2$ such that $\{r, x_i^{j_1}\} \in E(T)$ but $\{r, x_i^{j_2}\} \notin E(T)$. By Claim 2, $\{s, \overline{x_i}^{j_2}\} \in E(T)$. Thus, $(x_i^{j_2}, \overline{x_i}^{j_2}, s, r, x_i^{j_1}, \overline{x_i}^{j_1})$ is the $(x_i^{j_2}, \overline{x_i}^{j_1})$-path in T, hence $d_T(x_i^{j_2}, \overline{x_i}^{j_1}) = 5$. However, by Observation 3, $d_T(x_i^{j_2}, \overline{x_i}^{j_1}) \le (2k+1) - 2(k-1) = 3$, a contradiction. Thus, all or none of the edges $\{r, x_i^j\}$, with $1 \le j \le m$, belong to T. By symmetry, all or none of the edges $\{s, \overline{x_i}^j\}$ with $1 \le j \le m$, belong to T. Claim 3 follows. □

Now, define a truth assignment f for variables x_i, $1 \le i \le n$, as follows:

$$f(x_i) = \begin{cases} \texttt{true} & \text{if, for some } j, \quad \{r, x_i^j\} \in E(T), \\ \texttt{false} & \text{otherwise.} \end{cases}$$

By Claim 3, f is well-defined. We are going to show that $f(F) = \texttt{true}$.

First, consider a positive clause $C_j = (x_{i_1}, x_{i_2}, x_{i_3})$ and assume to the contrary that $f(x_{i_1}) = f(x_{i_2}) = f(x_{i_3}) = \texttt{false}$. That is, $\{r, x_{i_1}^j\}$, $\{r, x_{i_2}^j\}$, and $\{r, x_{i_3}^j\}$ do not belong to T.

By Claim 2, $\{s, \overline{x_{i_1}}^j\}$, $\{s, \overline{x_{i_2}}^j\}$, and $\{s, \overline{x_{i_3}}^j\}$ are edges of T. Recall that the edges $\{c_j^+, d_j^+\}$, $\{d_j^+, d_j^-\}$, and $\{d_j^-, c_j^+\}$ are edges of T, too.

Now, since T is a tree, exactly one of the edges $\{c_j^+, x_{i_1}^j\}$, $\{c_j^+, x_{i_2}^j\}$, $\{c_j^+, x_{i_3}^j\}$, and $\{c_j^-, r\}$ belongs to T. If $\{c_j^-, r\} \in E(T)$ then $(c_j^+, d_j^+, d_j^-, c_j^-, r, s, \overline{x_{i_1}}^j, x_{i_1}^j)$ is the $(c_j^+, x_{i_1}^j)$-path in T, hence $d_T(c_j^+, x_{i_1}^j) = 7$. However, by Observation 3, $d_T(c_j^+, x_{i_1}^j) \le (2k+1) - 2(k-2) = 5$, a contradiction. If $\{c_j^+, x_i^j\} \in E(T)$ for one $i \in \{i_1, i_2, i_3\}$ then $(c_j^-, d_j^-, d_j^+, c_j^+, x_i^j, \overline{x_i}^j, s, r)$ is the (c_j^-, r)-path in T, hence $d_T(c_j^-, r) = 7$, contradicting Observation 3 again.

Thus, all positive and, similarly, all negative clauses C_j are satisfied by the assignment f.

Thus each clause C_j of F is satisfied by the assignment f, proving Lemma 7. □

DEFINITION 8. If $x_i \in C_j$ ($\overline{x_i} \in C_j$) then we say, for convenience, that the vertex x_i^j ($\overline{x_i}^j$, respectively) is the *corresponding vertex* of the variable x_i (literal $\overline{x_i}$, respectively). Note that the corresponding vertex is not shared by two clauses.

LEMMA 9. *Suppose F is satisfiable. Then G admits a tree $(2k+1)$-spanner.*

PROOF. Let f be a truth assignment for variables x_i that satisfy F. We first construct a spanning tree T' of G', the subgraph of G induced by $p, q, r, s, x_i^j, \overline{x_i}^j$ with $1 \le i \le n$, $1 \le j \le m$, $c_j^+, c_j^-, d_j^+, d_j^-$ with $1 \le j \le m$. Take

- $\{p, r\}, \{r, s\}, \{s, q\}, \{x_i^j, \overline{x_i}^j\}$ with $1 \le i \le n$, $1 \le j \le m$,
- $\{c_j^+, d_j^+\}, \{d_j^+, d_j^-\}, \{d_j^-, c_j^-\}$ with $1 \le j \le m$,
- $\{r, x_i^j\}$ with $1 \le i \le n$, $1 \le j \le m$, where $f(x_i) = \texttt{true}$,
- $\{s, \overline{x_i}^j\}$ with $1 \le i \le n$, $1 \le j \le m$, where $f(x_i) = \texttt{false}$

into T'.

Next, for each clause C_j choose a true literal $l \in C_j$ and let $l_i^j \in \{x_i^j\colon 1 \le i \le n\} \cup \{\overline{x_i}^j\colon 1 \le i \le n\}$ be the corresponding vertex of l. Then take the edge connecting l_i^j and its neighbor in $\{c_j^+, c_j^-\}$ into T'.

So far, T' is a tree. Moreover, by case analysis, the following holds:

CLAIM 1. *T' is a tree 5-spanner of G' such that if $\{a, b\}$ is an edge forced (in G) by an S_{k-1} then $d_{T'}(a, b) = 3$.*

Finally, extend T' at each forced edge using Observation 2 in an obvious way to obtain a spanning tree T of G. More precisely, if $\{a, b\} \in T'$ and is forced by an $S_\ell[a, b]$ ($\ell \in \{k-2, k-1, k\}$) then take a tree $(2\ell + 1)$-spanner T'' in that $S_\ell[a, b]$ containing the edge $\{a, b\}$ into T; such a tree spanner T'' exists by Observation 2. Clearly, after taking T'' into T, T remains a tree.

If $\{a, b\} \notin T'$ and is forced by an $S_\ell[a, b]$ ($\ell \in \{k-2, k-1\}$) then let T'' be a tree $(2\ell + 1)$-spanner in that $S_\ell[a, b]$ containing the edge $\{a, b\}$. Take the two connected components of $T'' - \{a, b\}$ into T. Since there is an (a, b)-path in T', T remains a tree after taking $T'' - \{a, b\}$ into T.

Now we show that T is a tree $(2k + 1)$-spanner of G. As $2k + 1 \geq 5$ and by Claim 1, we only have to check for edges $\{x, y\}$ in an $S_\ell[a, b]$, $\ell \in \{k-2, k-1, k\}$. Let T'' be the tree $(2\ell + 1)$-spanner in that $S_\ell[a, b]$ which has been chosen in extending T' to T. If $\{a, b\} \in T'$, then by definition of T,

$$d_T(x, y) = d_{T''}(x, y) \leq 2\ell + 1 \leq 2k + 1.$$

If $\{a, b\} \notin T'$, then by definition of T', $\ell \neq k$. Write T''_a, T''_b for the connected components of $T'' - \{a, b\}$ containing a, respectively, b. If $\{x, y\} \in T''_a$ (or $\{x, y\} \in T''_b$), we again have

$$d_T(x, y) = d_{T''_a}(x, y) \leq 2\ell + 1 < 2k + 1.$$

Thus, let $x \in T''_a$, $y \in T''_b$, say. Now, the (x, y)-path in T consists of the (x, a)-path in T''_a, the (y, b)-path in T''_b, and the (a, b)-path in T'. Hence

$$d_T(x, y) = d_{T''}(x, y) - 1 + d_{T'}(a, b).$$

As T'' is a $(2\ell + 1)$-spanner in $S_\ell[a, b]$ and by Claim 1, if $\ell = k - 1$ then

$$d_T(x, y) \leq (2(k-1) + 1) - 1 + 3 = 2k + 1,$$

and if $\ell = k - 2$ then

$$d_T(x, y) \leq (2(k-2) + 1) - 1 + 5 = 2k + 1.$$

Thus, T is a tree $(2k + 1)$-spanner of G as claimed. □

Lemmas 6, 7, and 9 immediately imply the main theorem of this section:

THEOREM 10. *For every fixed $k \geq 2$, the tree $(2k+1)$-spanner problem is* NP*-complete for chordal bipartite graphs.*

4. Tree 3-Spanners for Bipartite ATE-Free Graphs. In this section we show that any bipartite ATE-free graph admits a tree 3-spanner.

We say that a vertex u of a graph G has a *maximum neighbor* if there is a vertex w in G such that $N(N(u)) = N(w)$. We will need the following result from [27].

LEMMA 11 [27]. *Any chordal bipartite graph G has a vertex with a maximum neighbor.*

It is easy to deduce from results [28, Lemma 4.4], [27, Corollary 5], [29, Corollary 1] that a vertex with a maximum neighbor of a chordal bipartite graph can be found in linear time by the following procedure:

PROCEDURE NICE-VERTEX. Find a vertex with a maximum neighbor

Input: A chordal bipartite graph $G = (X \cup Y, E)$.
Output: A vertex with a maximum neighbor.
Method:
initially all vertices $v \in X \cup Y$ are unmarked;
repeat
among unmarked vertices of X select a vertex x such that $N(x)$ contains the maximum number of marked vertices;
mark x and all its unmarked neighbors;
until all vertices in Y are marked;
output the vertex of Y marked last.

Now let $G = (V, E)$ be a connected bipartite ATE-free graph and let u be a vertex of G which has a maximum neighbor (recall that G is chordal bipartite and therefore such a vertex u exists).

LEMMA 12. *Let S be a connected component of a subgraph of G induced by set $V \backslash D_{k-1}(u)$ $(k \geq 1)$. Then there is a vertex $w \in N_{k-1}(u)$ such that $N(w) \supset S \cap N_k(u)$.*

PROOF. Since u has a maximum neighbor, we have $N_2(u) \subset N(w)$ for some vertex $w \in N(u)$. Consider now a connected component S of a subgraph of G induced by set $V \backslash D_{k-1}(u)$ $(k \geq 3)$. Let w be a vertex of $N_{k-1}(u)$ such that $|N(w) \cap S \cap N_k(u)|$ is maximal. Assume that there is a vertex x in $S \cap N_k(u)$ which is not adjacent to w. Then, by maximality, for any neighbor z of x in $N_{k-1}(u)$, there must exist a vertex y in $S \cap N_k(u)$ such that $\{y, w\} \in E$ and $\{y, z\} \notin E$. Since vertices x and y both belong to S, they are connected by a path P of G consisting only of vertices from $V \backslash D_{k-1}(u)$. Let y', x' be the neighbors on P of y and x, respectively. Clearly, since G is bipartite, $x', y' \in N_{k+1}(u)$ and $\{y, x\} \notin E$. Consider also shortest paths $P(w, u)$ and $P(z, u)$ of G connecting vertex u with w and z, respectively. Vertex x' cannot be adjacent with y since otherwise a subgraph of G formed by edges $\{y, x'\}$, $\{x', x\}$, $\{y, w\}$, $\{x, z\}$ and paths $P(w, u)$, $P(z, u)$ will contain an induced cycle of length at least 6, which is impossible. Analogously, vertex y' is not adjacent with x. We claim now that edges $a = \{y, y'\}$, $c = \{x, x'\}$, and $e = \{u, v\}$, where v is a neighbor of u on $P(w, u)$, form an ATE in G. Indeed, P avoids the neighborhood of e since $P \subseteq V \backslash D_{k-1}$ and $k > 2$, path $(y, w) \cup P(w, u)$ avoids the neighborhood of c and path $(x, z) \cup P(z, u)$ avoids the neighborhood of a. A contradiction obtained proves that $N(w) \supset S \cap N_k(u)$. □

This lemma suggests the following algorithm for constructing a spanning tree of G:

PROCEDURE SPAN-ATEG. Tree 3-spanners for bipartite ATE-free graphs

Input: A bipartite ATE-free graph $G = (V, E)$ and a vertex u of G with a maximum neighbor.
Output: A spanning tree $T = (V, E')$ of G (rooted at u).
Method:

set $E' := \emptyset$;
set $q := \max\{d_G(u, v)\colon v \in V\}$;
let s_i^q, $i \in \{1, \ldots, p_q\}$, be the vertices of $N_q(u)$;
for every $i \in \{1, \ldots, p_q\}$ **do**
  pick a neighbor w of s_i^q in $N_{q-1}(u)$;
  add edge $\{s_i^q, w\}$ to E';
for $k := q - 1$ **downto** 1 **do**
  compute the connected components $S_1^k, \ldots, S_{p_k}^k$ of $G[N_k(u) \cup \{s_i^{k+1}, i \in \{1, \ldots, p_{k+1}\}\}]$;
  for every $i \in \{1, \ldots, p_k\}$ **do**
    set $S := S_i^k \cap N_k(u)$;
    pick a vertex w in $N_{k-1}(u)$ such that $N(w) \supset S$;
    for each $v \in S$ add the edge $\{v, w\}$ to E';
    shrink component S_i^k to a vertex s_i^k and make s_i^k adjacent in G to all vertices from $N(S_i^k) \cap N_{k-1}(u)$.

It is easy to see that the graph $T = (V, E')$ constructed by this procedure is a spanning tree of G and its construction takes only linear time. Moreover, T is a shortest path tree of G rooted at u since, for any vertex $x \in V$, $d_G(x, u) = d_T(x, u)$ holds.

THEOREM 13. *Let $T = (V, E')$ be a spanning tree of a bipartite ATE-free graph $G = (V, E)$ output by PROCEDURE SPAN-ATEG. Then, for any $x, y \in V$, we have $d_T(x, y) \leq 3 \cdot d_G(x, y)$ and $d_T(x, y) \leq d_G(x, y) + 2$.*

PROOF. First we show that $d_T(x, y) \leq 3$ holds for any edge $\{x, y\}$ of G. Since G is bipartite, $|d_G(x, u) - d_G(y, u)| = 1$ must hold. Without loss of generality, assume that $x \in N_k(u)$ and $y \in N_{k-1}(u)$. Let x' be the father of x in T. If $x' = y$ we are done; $d_T(x, y) = 1$. Otherwise, x' and y are from $N_{k-1}(u)$ and belong to a common connected component of the graph $G[V \backslash D_{k-2}(u)]$. According to the algorithm, x' and y share a common father in T. Hence, $d_T(x, y) = d_T(x', y) + 1 = 3$.

Now consider two arbitrary vertices v and w of G and a shortest (v, w)-path. Applying the inequality $d_T(x, y) \leq 3$ to every edge $\{x, y\}$ of this path, we will get $d_T(v, w) \leq 3 \cdot d_G(v, w)$. That $d_T(x, y) \leq d_G(x, y) + 2$ already follows from the previous part of our proof and from Lemma 1 of [30]. For the sake of completeness, we present another proof here. Since T is a shortest path tree of G rooted at u, the distances in G and T between a vertex and any of its ancestors are the same. We will prove that $d_T(x, y) \leq d_G(x, y) + 2$ by induction on $d_G(v, w)$. If v and w are adjacent, then we are done, because then $d_T(v, w) \leq 3$. Now suppose that $d_G(v, w) = s \geq 2$ and let z be a neighbor of v on a shortest path between v and w. From the induction assumption we have $d_T(z, w) \leq s - 1 + 2 = s + 1$ and $d_T(v, z) \leq 3$. Let $a = \text{nca}(v, z)$ be the nearest common ancestor of v and z in the tree T. Since $d_T(a, v) = d_G(a, v)$, $d_T(a, z) = d_G(a, z)$, and $\{v, z\} \in E$ we obtain that $|d_T(v, a) - d_T(z, a)| = 1$. We can additionally assume that $d_T(z, w) < d_T(v, w) - 1$, since otherwise we immediately conclude $d_T(v, w) \leq d_G(v, w) + 2$. From this and the previous inequality we deduce that the vertex $\text{nca}(w, z)$ lies on the path of T between the vertices a and z. Therefore, a is an ancestor of w, and thus $d_T(a, w) = d_G(a, w)$. Notice that the distance sums $d_T(v, w) + d_T(a, z)$ and $d_T(v, z) + d_T(a, w)$ are equal. Hence, $d_T(v, w) = d_T(a, w) - d_T(a, z) + d_T(v, z) = d_G(a, w) - d_G(a, z) + d_T(v, z) \leq d_G(w, z) + 3 = d_G(v, w) + 2$, concluding the proof. □

Any interval bigraph is a bipartite ATE-free graph, and any convex graph is an interval bigraph. Hence we have the following corollaries:

COROLLARY 14. *Any interval bigraph $G = (V, E)$ admits a spanning tree T such that $d_T(x, y) \leq 3 \cdot d_G(x, y)$ and $d_T(x, y) \leq d_G(x, y) + 2$ hold for any $x, y \in V$. Moreover, such a tree T can be constructed in linear time.*

COROLLARY 15 [6]. *Any convex graph $G = (V, E)$ admits a spanning tree T such that $d_T(x, y) \leq 3 \cdot d_G(x, y)$ and $d_T(x, y) \leq d_G(x, y) + 2$ hold for any $x, y \in V$. Moreover, such a tree T can be constructed in linear time.*

5. Tree 7-Spanners for (Enhanced) Probe Interval Graphs.

In this section we show that any (enhanced) probe interval graph admits a tree 7-spanner.

Let $G = (P, N, E)$ be a connected probe interval graph. We assume that an interval representation of G is given (if not, an interval model for G can be constructed by a method described in [24] in $O(m \log n)$ time, where $n = |P| + |N|$ and $m = |E|$). Let $\mathcal{I} = \{I_x\colon x \in P\}$ be the intervals in the interval model representing the probes and let $\mathcal{J} = \{J_y\colon y \in N\}$ be the intervals representing the nonprobes.

First we discuss two simple special cases. If $N = \emptyset$ then clearly $G = (P, E)$ is an interval graph. It is known (see [30]) that for any interval graph G and any vertex u of G there is a shortest path spanning tree T of G rooted at u such that $d_T(x, y) \leq d_G(x, y) + 2$ holds for any x, y. In fact, a procedure similar to PROCEDURE SPAN-ATEG produces such a spanner in linear time for any interval graph G and any start vertex u. Evidently, T is a tree 3-spanner of G.

To describe other special cases, we need the following notion. A connected probe interval graph $G = (P, N, E)$ is *superconnected* if for any two intersecting intervals $I_v, I_w \in \mathcal{I}$ there always is an interval $J_y \in \mathcal{J}$ such that $I_v \cap I_w \cap J_y \neq \emptyset$. For a superconnected probe interval graph G, a tree 4-spanner can be constructed easily. First we ignore all edges in $G[P]$ to get an interval bigraph $G' = (X = P, Y = N, E')$ and then run PROCEDURE SPAN-ATEG on G'. We claim that a spanning tree T of G', produced by that procedure, is a tree 4-spanner of G. Indeed, for any edge $\{x, y\}$ of G such that $x \in P$ and $y \in N$, $d_T(x, y) \leq 3$ holds by Corollary 14; it is an edge of G', too.

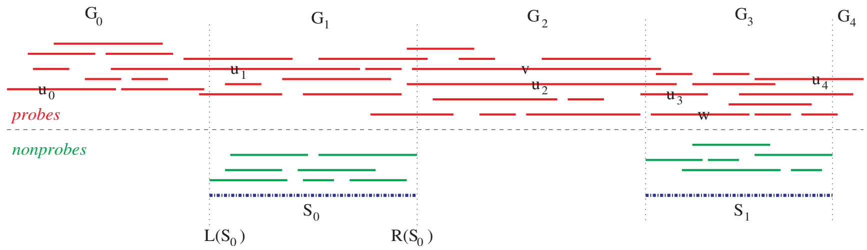

Fig. 6. Segments and a decomposition of a probe interval graph.

Now consider an edge $\{v, w\}$ of G with $v, w \in P$. Since G is superconnected, there is a vertex $y \in N$ such that $I_v \cap I_w \cap J_y \neq \emptyset$, i.e., $d_{G'}(v, w) = 2$. Then, by Corollary 14, we have $d_T(v, w) \leq d_{G'}(v, w) + 2 = 2 + 2 = 4$. Consequently, T is a tree 4-spanner of G.

To get a tree 7-spanner for an arbitrary connected probe interval graph $G = (P, N, E)$, we use the following strategy. First we decompose the graph G into subgraphs $G_0, G_1, \ldots, G_k$ such that G_i and G_j $(i \neq j)$ share at most one common vertex and each G_i is either an interval graph or a superconnected probe interval graph. Then iteratively, given a tree 7-spanner T^i for $G_0 \cup G_1 \cup \cdots \cup G_i$ $(i < k)$ and a tree t-spanner T_{i+1} $(t \leq 4)$ of G_{i+1}, we will extend T^i to a tree 7-spanner T^{i+1} for $G_0 \cup G_1 \cup \cdots \cup G_i \cup G_{i+1}$ by either making all vertices of G_{i+1} adjacent in T^{i+1} to a common neighbor in $G_0 \cup G_1 \cup \cdots \cup G_i$ (if it exists) or by gluing trees T^i and T_{i+1} at a common vertex.

Now we give a formal description of the decomposition algorithm. Let $S_0, S_1, \ldots, S_q$ be segments of the union $\bigcup_{y \in N} J_y$ (see Figure 6 for an illustration).

PROCEDURE DECOMP. A decomposition of a probe interval graph

Input: A probe interval graph G and its interval representation $(\mathcal{I}, \mathcal{J})$.
Output: Subgraphs $G_0, G_1, \ldots, G_{2q+2}$ of G, where G_{2i} $(i \in \{0, \ldots, q+1\})$ is an interval graph and G_{2i+1} $(i \in \{0, \ldots, q\})$ is a superconnected probe interval graph, and special vertices u_j $(j = 1, \ldots, 2q + 2)$, where u_j belongs to G_{j-1} and G_j.
Method:

```
for i = 0 to q do
  /* define an interval graph */
  set 𝒳 := {I_x ∈ ℐ: L(x) ≤ L(S_i)};
  on intervals 𝒳 define an interval graph G_{2i};
  let I* be an interval from 𝒳 with maximum R(·) value;
  set u_{2i+1} := a vertex of G corresponding to I*;
  set ℐ := ℐ\(𝒳\{I*});
  /* define a superconnected probe interval graph */
  set 𝒴 := {I_y ∈ 𝒥: I_y ⊆ S_i};
  set 𝒳 := {I_x ∈ ℐ: L(x) ≤ R(S_i)};
  define a probe interval graph G_{2i+1} with probes 𝒳 and nonprobes 𝒴;
  let I* be an interval from 𝒳 with maximum R(·) value;
  set u_{2i+2} := a vertex of G corresponding to I*;
  set ℐ := ℐ\(𝒳\{I*});
define on ℐ an interval graph G_{2q+2}.
```

Clearly, all probe interval graphs G_{2i+1} $(i = 1, \ldots, q)$ are superconnected and a decomposition of G into $G_0, G_1, \ldots, G_{2q+2}$ can be done in linear time if endpoints of the intervals $\mathcal{I} \cup \mathcal{J}$ are sorted.

LEMMA 16. *For any $i = 2, \ldots, 2q+2$, $R(u_i) \geq R(u_{i-1})$ holds.*

PROOF. When we delete an interval I_v from $\mathcal{I}$, we always leave in $\mathcal{I}$ an interval I_u such that $R(v) \leq R(u)$. □

Now, for an interval graph G_0 (if it is not empty), we can construct a tree 3-spanner $T_0 = T_0(u_0)$ rooted at any vertex u_0 of G_0. For an interval graph G_{2i} $(i = 1, \ldots, q+1)$, we can construct a tree 3-spanner $T_{2i} = T_{2i}(u_{2i})$ rooted at vertex u_{2i} (see PROCEDURE DECOMP). Since all those trees are shortest path trees, the neighborhoods of vertex u_{2i} in G_{2i} and T_{2i} coincide.

Let G^-_{2i+1} be an interval bigraph obtained from a superconnected probe interval graph G_{2i+1} by ignoring all edges between probes and deleting all probes I_v such that $I_v \subset I_{u_{2i+1}}$.

LEMMA 17. *For any $i = 0, \ldots, q$, vertex u_{2i+1} has a maximum neighbor in G^-_{2i+1}.*

PROOF. According to PROCEDURE DECOMP, interval I^* corresponding to u_{2i+1} belongs to $\mathcal{I}$. Let J_y be an interval of $\mathcal{J}$ such that $L(J_y) \leq R(I^*)$ and $R(J_y)$ is maximum. We show that vertex y of G_{2i+1} is a maximum neighbor of u_{2i+1} in G^-_{2i+1}. Consider a vertex w of G_{2i+1} which is at distance 2 from u_{2i+1} in G^-_{2i+1} and assume that intervals J_y, I_w do not intersect. If $R(I_w) < L(J_y)$, then necessarily I_w is a subinterval of I^* and w is not a vertex of G^-_{2i+1}. Hence, we may assume that $R(J_y) < L(I_w)$. However, then, by maximality of $R(J_y)$, there cannot exist an interval in $\mathcal{J}$ which intersects both I^* and I_w. The latter contradicts our assumption that the distance in G^-_{2i+1} between u_{2i+1} and w is 2. □

Let $T^-_{2i+1} = T^-_{2i+1}(u_{2i+1})$ be a tree 3-spanner of an interval bigraph G^-_{2i+1} constructed starting at vertex u_{2i+1}, $i \in \{0, \ldots, q\}$ (see PROCEDURE SPAN-ATEG). Clearly, the neighborhoods of vertex u_{2i+1} in G^-_{2i+1} and T^-_{2i+1} coincide. We can extend tree T^-_{2i+1} to a spanning tree $T_{2i+1} = T_{2i+1}(u_{2i+1})$ of G_{2i+1} by adding, for each probe I_v of G_{2i+1} such that $I_v \subset I_{u_{2i+1}}$, a pendant vertex v adjacent to u_{2i+1}.

LEMMA 18. *$T_{2i+1}(u_{2i+1})$ is a tree 4-spanner for G_{2i+1}, $i \in \{0, \ldots, q\}$. Moreover, for any edge $\{w, u_{2i+1}\}$ of G_{2i+1}, $d_{T_{2i+1}}(w, u_{2i+1}) \leq 2$ holds.*

PROOF. Let $A = \{v\colon v$ is a vertex of G_{2i+1} such that $I_v \subset I_{u_{2i+1}}\}$ and let H be a superconnected probe interval graph obtained from G_{2i+1} by eliminating vertices of A. Since G^-_{2i+1} is the interval bigraph counterpart of H, tree T^-_{2i+1} is a tree 4-spanner for H. Consider now an edge $\{v, w\}$ of G_{2i+1}. We may assume that at least one of these vertices (say, v) is from A. If also $w \in A$ then, by construction, $d_{T_{2i+1}}(v, w) = 2$. If $w \notin A$ then, since I_v intersects I_w and $I_v \subset I_{u_{2i+1}}$, I_w must intersect $I_{u_{2i+1}}$ too. If w is a nonprobe, then

$\{w, u_{2i+1}\}$ is an edge of G^-_{2i+1} and hence of T^-_{2i+1}. If w is a probe, then $d_{G^-_{2i+1}}(u_{2i+1}, w) = 2 = d_{T^-_{2i+1}}(u_{2i+1}, w)$ since T^-_{2i+1} is a shortest path spanning tree (rooted at u_{2i+1}) of G^-_{2i+1}. Consequently, in both cases we have $d_{T_{2i+1}}(v, w) = 1 + d_{T^-_{2i+1}}(u_{2i+1}, w) \leq 3$. □

Now we are ready to construct a spanning tree T for the original probe interval graph $G = (P, N, E)$. We say that a vertex v of G *dominates* a subgraph G_k of G if every vertex of G_k, different from v, is adjacent to v in G.

PROCEDURE SPAN-PIG. Tree 7-spanner for probe interval graphs

Input: A probe interval graph $G = (P, N, E)$, its interval representation $(\mathcal{I}, \mathcal{J})$ and a decomposition of G into graphs $G_0, G_1, \ldots, G_{2q+2}$.
Output: A spanning tree $T = (P \cup N, E')$ of G.
Method:

```
set E' = ∅ and k := 0;
while k ≤ 2q + 2 do
  if there is an index j such that k ≤ j and u_k dominates G_j then do
    find the largest index j with that property;
    for each v in G_k ∪ ... ∪ G_j (v ≠ u_k) add edge {v, u_k} to E';
    set k := j + 1;
  else do
    if k is even then do
      find a tree 3-spanner T_k(u_k) of an interval graph G_k;
      add all edges of T_k(u_k) to E';
    if k is odd then do
      find a tree 4-spanner T_k(u_k) of a superconnected probe interval
      graph G_k;
      add all edges of T_k(u_k) to E';
    set k := k + 1.
```

It is easy to see that the tree T constructed by PROCEDURE SPAN-PIG is a spanning tree of G and its construction takes only linear time.

LEMMA 19. *If for graph G_k ($k \in \{0, \ldots, 2q+2\}$) there exists a vertex $u_i \in \{u_0, \ldots, u_k\}$ which dominates G_k, then there is a vertex $u_s \in \{u_0, \ldots, u_k\}$ such that $d_T(x, u_s) \leq 1$ holds for any x in G_k. Otherwise, if such a vertex u_i does not exist, then for any vertices x, y of G_k, $d_T(x, y) = d_{T_k}(x, y)$ holds.*

PROOF. Assume that such a vertex u_i exists, but for any u_s there is a vertex x in G_k such that $d_T(x, u_s) > 1$. By Lemma 16, $R(u_k) \geq R(u_i)$. Hence, vertex u_k is also adjacent to all vertices of G_k (except itself). On the other hand, since for any u_s there is a vertex x in G_k such that $d_T(x, u_s) > 1$, there was an iteration of the while loop in PROCEDURE SPAN-PIG where the edges of tree $T_k(u_k)$ were added to T. That is, it was detected that vertex u_k does not dominate G_k. A contradiction obtained proves the first part of the lemma. The second part is evident. □

COROLLARY 20. *For any vertices* x, y *of* G_k $(k \in \{0, \dots, 2q+2\})$, $d_T(x, y) \leq \max\{2, d_{T_k}(x, y)\}$ *holds*.

Such a vertex u_s described in Lemma 19 is called the *focus* of G_k in T.

LEMMA 21. T *is a tree* 7*-spanner for* G.

PROOF. Consider an edge $\{v, w\}$ of G. If both vertices v and w belong to the same graph G_k $(k = 0, \dots, 2q+2)$ then either $d_T(v, w) \leq 2$ or $d_T(v, w) = d_{T_k}(v, w) \leq 4$. Hence, we may assume that they are from different graphs. Clearly, v and w cannot both belong to N.

Case 1: $v \in P$ *and* $w \in N$. In this case there is a segment $S_i \in \{S_0, S_1, \dots, S_q\}$ such that $L(S_i) \leq L(w) \leq R(w) \leq R(S_i)$. Clearly, w is a vertex of G_{2i+1} and, by PROCEDURE DECOMP, no neighbor of w different from u_{2i+2} can belong to G_k $(k > 2i+1)$. Hence, $L(v) \leq L(S_i) \leq L(w) \leq R(v)$ must hold. Moreover, since $L(u_{2i+1}) \leq L(S_i) \leq L(w) \leq R(v) \leq R(u_{2i+1})$, vertices w and u_{2i+1} are adjacent in G and therefore in T_{2i+1}. So, $d_{T_{2i+1}}(w, u_{2i+1}) = 1$. The latter means that either vertices w and u_{2i+1} are adjacent in T or they are both adjacent to the focus of G_{2i+1} in T (see Lemma 19).

If v belongs to G_{2i} then, since u_{2i+1} is also in G_{2i} and $d_G(v, u_{2i+1}) = 1$, $d_{T_{2i}}(v, u_{2i+1}) \leq 3$ must hold. Hence, we have $d_T(v, w) \leq d_T(v, u_{2i+1}) + d_T(w, u_{2i+1}) \leq \max\{2, d_{T_{2i}}(v, u_{2i+1})\} + \max\{2, d_{T_{2i+1}}(w, u_{2i+1})\} \leq 3 + 2 = 5$.

Now assume that v belongs to G_j with $j < 2i$. Then vertex u_{j+1} dominates G_{j+1} since $R(u_{j+1}) \geq R(v) \geq L(S_i)$. Let u_s be the focus of G_{j+1} in T and let r be the largest index such that graph G_r is still dominated by u_s. By PROCEDURE SPAN-PIG, u_s is the focus in T of all graphs $G_{j+1}, \dots, G_r$. Therefore, vertices $u_{j+1}, \dots, u_{r+1}$ are all at distance at most 1 from u_s in T. We also have, by Corollary 20, $d_T(v, u_{j+1}) \leq \max\{2, d_{T_j}(v, u_{j+1})\} \leq 4$.

If $d_T(v, u_{j+1}) > 2$, then necessarily $s = j+1$ and $r \geq 2i$ (recall that $R(u_{j+1}) \geq L(S_i)$). Hence, $2 < d_T(v, u_s) \leq 4$, $d_T(u_s, u_{2i+1}) \leq 1$, and vertex w is adjacent in T either to u_s or to u_{2i+1}, depending on whether u_s dominates G_{2i+1} or not. Thus, we have $d_T(v, w) \leq d_T(v, u_s) + d_T(u_s, u_{2i+1}) + 1 \leq 4 + 1 + 1 = 6$.

Let now $d_T(v, u_{j+1}) \leq 2$. If u_s dominates G_{2i}, then again $d_T(u_s, u_{2i+1}) \leq 1$. Otherwise, $r < 2i$ and vertex u_{r+1} dominates G_{2i} since $R(u_{r+1}) \geq R(u_{j+1}) \geq R(v) \geq L(S_i)$. By PROCEDURE SPAN-PIG, u_{r+1} is the focus of G_{2i} in T. Hence, $d_T(u_{r+1}, u_{2i+1}) \leq 1$ and therefore $d_T(u_s, u_{2i+1}) = d_T(u_s, u_{r+1}) + d_T(u_{r+1}, u_{2i+1}) \leq 1 + 1 = 2$. Since w is adjacent in T either to u_{2i+1} or to the focus of G_{2i+1} in T, we get $d_T(v, w) \leq d_T(v, u_{j+1}) + d_T(u_{j+1}, u_s) + d_T(u_s, u_{2i+1}) + 1 \leq 2 + 1 + 2 + 1 = 6$.

Case 2: $v, w \in P$. Since w (as well as v) can be a vertex from $\{u_1, \dots, u_{2q+2}\}$, it can belong to few consecutive graphs $G_i, \dots, G_{i+a}$. Therefore, let i and j be the smallest indices such that w belongs to G_i and v belongs to G_j. Without loss of generality, assume also that $j < i$.

We have $R(v) \geq L(w)$. Since v in G_j is adjacent to w in G_i, vertex u_i must be adjacent to v. Since G_i can be a proper interval graph (if i is odd), by Lemma 18, we have $d_{T_i}(w, u_i) \leq 2$. Recall that, if G_i is an interval graph (i.e., i is even), then we would have $d_{T_i}(w, u_i) \leq 1$.

If $j = i - 1$, then both vertices v and u_i are in G_{i-1} and, therefore, $d_G(v, u_i) = 1$ implies $d_{T_{i-1}}(v, u_i) \leq 4$. Hence, we have $d_T(v, w) \leq d_T(v, u_i) + d_T(w, u_i) \leq \max\{2, d_{T_{i-1}}(v, u_i)\} + \max\{2, d_{T_i}(w, u_i)\} \leq 4 + 2 = 6$.

Now assume that $j < i - 1$. Then vertex u_{j+1} dominates G_{j+1} since $R(u_{j+1}) \geq R(v) \geq L(w)$. Let again u_s be the focus of G_{j+1} in T and let r be the largest index such that graph G_r is still dominated by u_s. Since u_s is the focus in T of all graphs $G_{j+1}, \ldots, G_r$, vertices $u_{j+1}, \ldots, u_{r+1}$ are all at distance at most 1 from u_s in T. By Corollary 20, we also have $d_T(v, u_{j+1}) \leq \max\{2, d_{T_j}(v, u_{j+1})\} \leq 4$.

If $d_T(v, u_{j+1}) > 2$, then again $s = j + 1$ and $r \geq i - 1$. Hence, $2 < d_T(v, u_s) \leq 4$, $d_T(u_s, u_i) \leq 1$ and therefore $d_T(v, w) \leq d_T(v, u_s) + d_T(u_s, u_i) + d_T(u_i, w) \leq 4 + 1 + 2 = 7$.

Let now $d_T(v, u_{j+1}) \leq 2$. If u_s dominates G_{i-1}, then again $d_T(u_s, u_i) \leq 1$. If u_s does not dominate G_{i-1}, then vertex u_{r+1} must dominate it (since $R(u_{r+1}) \geq R(u_{j+1}) \geq R(v)$). Therefore, by PROCEDURE SPAN-PIG, u_{r+1} is the focus of G_{2i} in T and $d_T(u_{r+1}, u_i) \leq 1$ must hold. Thus, $d_T(v, w) \leq d_T(v, u_{j+1}) + d_T(u_{j+1}, u_s) + d_T(u_s, u_i) + d_T(u_i, w) \leq 2 + 1 + 2 + 2 = 7$. □

The main theorem in this section is the following:

THEOREM 22. *Any probe interval graph G admits a tree 7-spanner. Moreover, such a tree 7-spanner can be constructed in $O(m \log n)$ time, or in $O(m + n \log n)$ time if the intersection model of G is given in advance.*

Now let $G = (P, N, E)$ be an enhanced probe interval graph with probes P and nonprobes N.

COROLLARY 23. *Any enhanced probe interval graph $G = (P, N, E)$ admits a tree 7-spanner. Moreover, such a tree spanner can be constructed in $O(m \log n)$ time.*

PROOF. By ignoring in G edges between nonprobes, we get a probe interval graph G'. Let T be a tree 7-spanner of G' constructed by PROCEDURE SPAN-PIG. We show that T is a tree 7-spanner of G, too. One needs to check the distance in T only between nonprobes $x, y \in N$ which are adjacent in G, i.e., $\{x, y\}$ is an enhanced edge. By the definition of an enhanced edge, there must exist two nonintersecting probes v, w such that $\{v, x\}$, $\{v, y\}$, $\{w, x\}$, $\{w, y\}$ are edges in G. Without loss of generality, assume that $R(v) < L(w)$. Then, clearly, vertices x, y, w are all from some superconnected probe interval graph G'_{2i+1} (see PROCEDURE DECOMP). Let G''_{2i+1} be the interval bigraph counterpart of G'_{2i+1} and let T_{2i+1} be a tree 3-spanner of G''_{2i+1}. For edges $\{w, x\}$ and $\{w, y\}$ of graph G''_{2i+1} we have $d_T(w, x) \leq \max\{2, d_{T_{2i+1}}(w, x)\} \leq 3$ and $d_T(w, y) \leq \max\{2, d_{T_{2i+1}}(w, y)\} \leq 3$ (recall that edges $\{w, x\}$ and $\{w, y\}$ are connecting a probe with nonprobes and hence they are both edges of G''_{2i+1}). Thus, $d_T(x, y) \leq d_T(x, w) + d_T(y, w) \leq 3 + 3 = 6$. □

6. Concluding Remarks. In the paper we have shown that the tree t-spanner problem is NP-complete even for chordal bipartite graphs for $t \geq 5$. The complexity of the tree 3-spanner problem is still open. We have also shown that every (enhanced) probe interval graph has a tree 7-spanner. However, it is also open whether the graph classes are tree t-spanner admissible for smaller t.

Appendix. Graph Theoretic Aspects of Probe Interval Graphs. We first introduce the graph classes which appear only in this appendix, and show the relationships between the graph classes related to the probe interval graphs and the other graph classes.

A.1. Definitions. For a cycle $C = (v_1, v_2, \ldots, v_{2k})$ of even length, an *odd chord* $\{v_i, v_j\}$ is a chord such that $i - j$ is odd. A chordal graph is *strongly chordal* if each cycle of even length at least 6 has an odd chord.

Assume that for a graph $G = (V, E)$ there is a tree $\mathcal{T}$ and a set $T = \{T_v | v \in V\}$ of subtrees of $\mathcal{T}$ such that $T_u \cap T_v \neq \emptyset$ if and only if $\{u, v\} \in E$. In this case, $(\mathcal{T}, T)$ is called a *tree model* for G. If $\mathcal{T}$ is a directed tree, i.e., if each edge of $\mathcal{T}$ has a fixed orientation, and if all subtrees are directed paths, then G is called a *directed path graph*. Moreover, if $\mathcal{T}$ can be chosen to be a rooted tree such that all edges are oriented downwards, i.e., in the direction from the root r to the leaves, then G is called a *rooted directed path graph*.

Note that this paper distinguishes between rooted directed path graphs and directed path graphs. The first class is a subclass of strongly chordal graphs, while the second class is not. Sometimes both classes are referred to as directed path graphs, which may lead to potential confusion.

A probe interval graph $G = (P, N, E)$ is an *STS-probe interval graph* if every edge joins a vertex in P and another in N. That is, probes do not overlap in an STS-probe interval graph.

A.2. Relations. The relations shown in this section are summarized in Figure 7. We first show two simple propositions:

PROPOSITION 24. *For any probe interval graph $G = (P, N, E)$, the graph G' obtained by removing all edges joining vertices in P is an interval bigraph.*

PROOF. Trivial by the definitions. □

Although Proposition 24 is trivial, combining it with Müller's result in [16], that interval bigraphs are subset of chordal bipartite graphs, we give a simple proof of Theorem 4.1 in [23], which states that G' in Proposition 24 is chordal bipartite.

PROPOSITION 25. *The class of the STS-probe interval graphs is equivalent to the class of convex graphs.*

PROOF. Let G be an STS-probe interval graph $G = (P, N, E)$ with $P = \{v_1, v_2, \ldots, v_n\}$. Let I_i be the interval corresponding to the vertex v_i for each $i = 1, \ldots, n$. Then we have

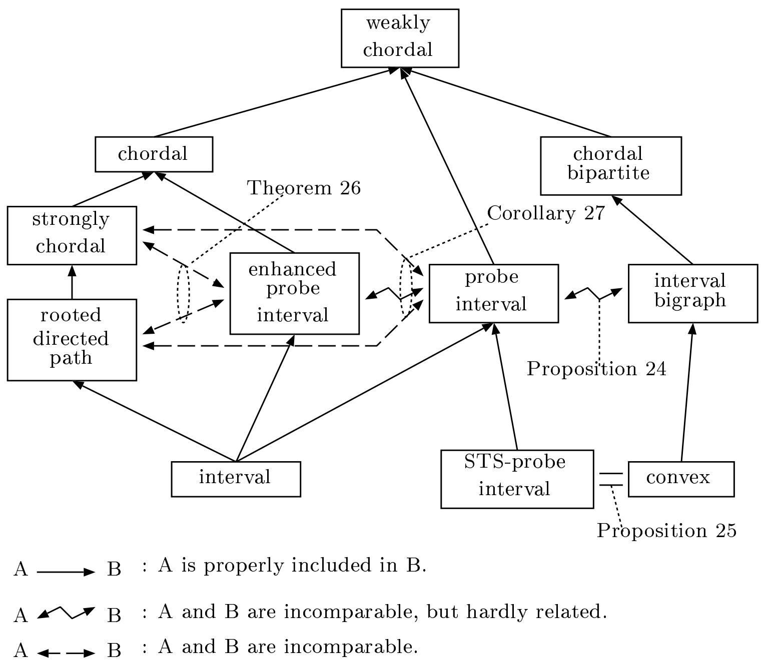

Fig. 7. Hierarchy of graph classes.

$I_i \cap I_j = \emptyset$ for each i, j with $i \neq j$. Thus we can (re)order the vertices such that $L(I_1) < R(I_1) < L(I_2) < R(I_2) < \cdots < L(I_n) < R(I_n)$. Hence for the graph $G = (X, Y, E)$ with $X = P$ and $Y = N$, the ordering of X fulfills the adjacency property. Thus, G is a convex graph. Similarly, for any convex graph $G' = (X', Y', E')$, we immediately have an STS-probe interval graph (P', N', E') letting $P = X'$ and $N = Y'$ if X' has an ordering with adjacency property. □

THEOREM 26. *The class of enhanced probe interval graphs is incomparable with both strongly chordal graphs and rooted directed path graphs.*

Before the proof of Theorem 26, we introduce some graph notions; a *sun* is a chordal graph G on $2n$ vertices for some $n \geq 3$ whose vertex set can be partitioned into two sets, $W = \{w_1, \ldots, w_n\}$, $U = \{u_1, \ldots, u_n\}$, such that W is independent and for each i and j, w_j is adjacent to u_i if and only if $i = j$ or $i \equiv j + 1 \pmod n$. It is known that a graph G is strongly chordal if and only if G is sun-free chordal (see [31] and [18] for further details). Now we are ready to prove the theorem.

PROOF. It is sufficient to show that there are two graphs G_1 and G_2 such that G_1 is an enhanced probe interval graph, and not a strongly chordal graph (hence not a rooted directed path graph), and G_2 is a rooted directed path graph (hence a strongly chordal graph), and not an enhanced probe interval graph.

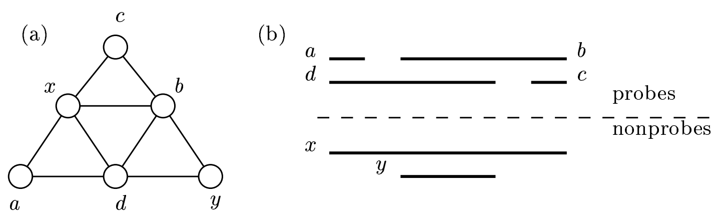

Fig. 8. Enhanced probe interval and not strongly chordal graph.

The graph G_1 in Figure 8(a) is a sun with six vertices, which is not a strongly chordal graph. Letting $P = \{a, b, c, d\}$ and $N = \{x, y\}$, the sun is an enhanced probe interval graph (Figure 8(b)).

To construct the graph G_2, we first consider the graph depicted in Figure 9(a) as an enhanced probe interval graph. A possible representation is shown in Figure 9(b). We claim that the vertex c has to be a nonprobe to represent G_2, and hence x_1, x_2, and x_3 are probes. To show this, we assume that c is a probe. Since each y_i does not intersect c, without loss of generality we can assume that either $R(y_1) < R(y_2) < L(c) < R(c) < L(y_3)$ or $R(y_1) < R(y_2) < R(y_3) < R(c)$. In both cases, x_1 has to intersect y_1 and c without intersecting y_2. Thus, the only possible way is x_1 and y_2 are nonprobes, and x_1 contains the interval $[R(y_1), L(c)]$. Since x_1 is a nonprobe, y_1 is a probe. On the other hand, since y_2 is a nonprobe, x_2 is a probe. Probe x_2 has to intersect both y_2 and c. Thus x_2 contains the interval $[R(y_2), L(c)]$, which is covered by x_1. Since x_2 is a probe, we have an edge $\{x_1, x_2\}$, which is a contradiction.

The graph G_2 is now depicted in Figure 10(a). Assume that G_2 is an enhanced probe interval graph. To represent the graph $G[\{c_1, x_1, y_1, x_2, y_2, c_2, x_3\}]$, c_1 has to be a nonprobe and c_2 has to be a probe by the claim. On the other hand, to represent the graph $G[\{c_2, c_1, x_2, x_3, y_3, x_4, y_4\}]$, c_2 has to be a nonprobe and c_1 has to be a probe, which is a contradiction. Thus, G_2 is not an enhanced probe interval graph. We next show that G_2 is a rooted directed path graph; a tree representation of G_2 is given in Figure 10(b), which completes the proof. $\square$

In the arguments in the proof of Theorem 26, no enhanced edge appears (in fact, an enhanced edge has to be a chord of a C_4). Thus, we immediately have the following:

COROLLARY 27. *The class of probe interval graphs is incomparable with both strongly chordal graphs and rooted directed path graphs.*

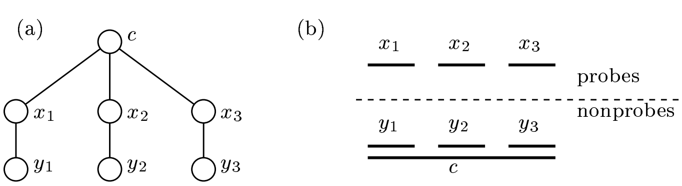

Fig. 9. Probe interval graph with nonprobe c.

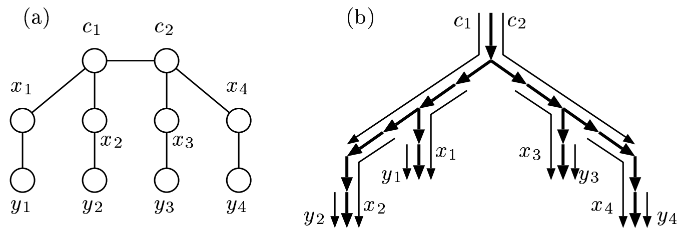

Fig. 10. Rooted directed path, and not enhanced probe interval graph.

References

[1] L. Cai and D.G. Corneil. Tree Spanners: An Overview. *Congressus Numerantium*, 88:65–76, 1992.

[2] L. Cai and D.G. Corneil. Tree Spanners. *SIAM Journal on Discrete Mathematics*, 8(3):359–387, 1995.

[3] B. Awerbuch, A. Baratz, and D. Peleg. Efficient Broadcast and Light-Weighted Spanners. Manuscript, 1992.

[4] D. Peleg. *Distributed Computing*: *A Locally-Sensitive Approach*. Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia, PA, 2000.

[5] H.-J. Bandelt and A. Dress. Reconstructing the Shape of a Tree from Observed Dissimilarity Data. *Advances in Applied Mathematics*, 7:309–343, 1986.

[6] G. Venkatesan, U. Rotics, M.S. Madanlal, J.A. Makowsy, and C.P. Rangan. Restrictions of Minimum Spanner Problems. *Information and Computation*, 136:143–164, 1997.

[7] D. Peleg and A.A. Schäffer. Graph Spanners. *Journal of Graph Theory*, 13(1):99–116, 1989.

[8] J. Soares. Graph Spanners: A Survey. *Congress Numerantium*, 89:225–238, 1992.

[9] A. Brandstädt, F.F. Dragan, H.-O. Le, and V.B. Le. Tree Spanners on Chordal Graphs: Complexity and Algorithms. *Theoretical Computer Science*, 310:329–354, 2004.

[10] A. Brandstädt, V. Chepoi, and F. Dragan. Distance Approximating Trees for Chordal and Dually Chordal Graphs. *Journal of Algorithms*, 30(1):166–184, 1999.

[11] M.S. Madanlal, G. Venkatesan, and C. P. Rangan. Tree 3-Spanners on Interval, Permutation and Regular Bipartite Graphs. *Information Processing Letters*, 59:97–102, 1996.

[12] H.-O. Le and V.B Le. Optimal Tree 3-Spanners in Directed Path Graphs. *Networks*, 34:81–87, 1999.

[13] A. Brandstädt, J. Spinrad, and L. Stewart. Bipartite Permutation Graphs are Bipartite Tolerance Graphs. *Congressus Numerantium*, 58:165–174, 1987.

[14] M.C. Golumbic and C.F. Goss. Perfect Elimination and Chordal Bipartite Graphs. *Journal of Graph Theory*, 2:155–163, 1978.

[15] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.

[16] H. Müller. Recognizing Interval Digraphs and Interval Bigraphs in Polynomial Time. *Discrete Applied Mathematics*, 78:189–205, 1997. Erratum is available at http://www.comp.leeds.ac.uk/hm/pub/node1.html.

[17] F. Harary, J.A. Kabell, and F.R. McMorris. Bipartite Intersection Graphs. *Commentationes Mathematicae Universitatis Carolinae*, 23:739–745, 1982.

[18] A. Brandstädt, V.B. Le, and J.P. Spinrad. *Graph Classes*: *A Survey*. SIAM, Philadelphia, PA, 1999.

[19] P. Zhang. Probe Interval Graphs and Its Applications to Physical Mapping of DNA. Manuscript, 1994.

[20] F.R. McMorris, C. Wang, and P. Zhang. On Probe Interval Graphs. *Discrete Applied Mathematics*, 88:315–324, 1998.

[21] T.A. McKee and F.R. McMorris. *Topics in Intersection Graph Theory*. SIAM, Philadelphia, PA, 1999.

[22] P. Zhang. United States Patent. Method of Mapping DNA Fragments. [Online] Available at http://www.cc.columbia.edu/cu/cie/techlists/patents/5667970.htm, July 3, 2000.

[23] J.L. Johnson and J.P. Spinrad. A Polynomial Time Recognition Algorithm for Probe Interval Graphs. In *Proceedings of the* 12*th Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 477–486. ACM, New York, 2001.

[24] R.M. McConnell and J.P. Spinrad. Construction of Probe Interval Models. In *Proceedings of the* 13*th Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 866–875. ACM, New York, 2002.

[25] D. König. *Theorie der endlichen und unendlichen Graphen* (in German). Akademische Verlagsgesellschaft, 1936.

[26] M.R. Garey and D.S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.

[27] A. Brandstädt, F. Dragan, V. Chepoi, and V. Voloshin. Dually Chordal Graphs. *SIAM Journal on Discrete Mathematics*, 11(3):437–455, 1998.

[28] A. Brandstädt, V.D. Chepoi, and F.F. Dragan. The Algorithmic Use of Hypertree Structure and Maximum Neighbourhood Orderings. *Discrete Applied Mathematics*, 82:43–77, 1998.

[29] F.F. Dragan and V.I. Voloshin. Incidence Graphs of Biacyclic Hypergraphs. *Discrete Applied Mathematics*, 68:259–266, 1996.

[30] E. Prisner. Distance approximating spanning trees. In *Proceedings of STACS '97*, pages 499–510. Lecture Notes in Computer Science, Vol. 1200. Springer-Verlag, Berlin, 1997.

[31] M. Farber. Characterization of Strongly Chordal Graphs. *Discrete Mathematics*, 43:173–189, 1983.