2003－AL－90 (9)
2003／5／23

2 部グラフと Probe 区間グラフにおける木スパナー

A. Brandstädt[1], F.F. Dragan[2], H.-O. Le[1], V.B. Le[1], 上原 隆平[3]

グラフ G の全域木 T において，T 上の任意の 2 点間の距離が G 上の距離の高々t 倍で押さえられるとき，T をグラフ G の t-木スパナーと呼ぶ．t-木スパナー問題とは，与えられたグラフが t-木スパナーを持つかどうかを判定する問題である．本稿ではまず 2 部グラフに関して，次の二つの結果を示す．(1)chordal 2 部グラフ上に制限してもこの問題が NP 困難であること．(2)ATE–free 2 部グラフは 3-木スパナーを持ち，それが線形時間で求められること．これらは既知の結果を改善している．次に Probe 区間グラフに関する結果を示す．このグラフクラスは DNA の解析に用いられるモデルで，グラフ理論的には区間グラフを一般化したものである．本稿では Probe 区間グラフは 7-木スパナーを持つことと，それが $O(m \log n)$ 時間で求められることを示す．

Tree Spanners for Bipartite Graphs and Probe Interval Graphs

A. Brandstädt[1], F.F. Dragan[2], H.-O. Le[1], V.B. Le[1], and R. Uehara[3]

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 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 best known before 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.

Keywords: 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 [9, 10], which finds numerous applications in distributed systems and communication networks; for example, it was shown that tree spanners can be used as models for broadcast operations [1] (see also [23]). Moreover, tree spanners were used in the area of biology [2], and approximating the bandwidth of graphs [27]. We refer to [24, 26, 6] for more background information on tree spanners.

The tree t-spanner problem is NP-complete in general [10] 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 [6], strongly chordal graphs are tree 4-spanner admissible [3] (i.e., every strongly chordal graph has a tree 4-spanner), and the following graph classes are tree 3-spanner admissible: interval graphs [18], directed path graphs [17], split graphs [27] (see also [6] 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 wide and important class from

[1]Institut für Theoretische Informatik, Fachbereich Informatik, Universität Rostock, 18051 Rostock, Germany. {ab,hoang-oanh.le,le}@informatik.uni-rostock.de
[2]Dept. of Computer Science, Kent State University, Ohio, USA. dragan@cs.kent.edu
[3]駒澤大学自然科学教室.uehara@komazawa-u.ac.jp

both practical and theoretical points of view. However, the known results for the complexity of the tree t-spanner problem for the bipartite graphs and its subclasses are few comparing to the chordal graphs and its subclasses. The NP-completeness is only known for general bipartite graphs (this result can be deduced from the construction in [10]), 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 [18]; and any convex graph is tree 3-spanner admissible [27]. (The convex graphs were introduced by Brandstädt, Spinrad, and Stewart; see [8] 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 and has applications to nonsymmetric matrices [14, 13]. 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 [22] to characterize interval bigraphs. The class of interval bigraphs is a bipartite analog of interval graphs and was introduced by Harary, Kabell, and McMorris [15].

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 [22, 7]; convex graphs $\subset$ interval bigraphs $\subset$ bip. ATE–free graphs $\subset$ chordal bipartite graphs $\subset$ bipartite graphs.

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 [28, 21, 20, 29] for 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 paper [28, 29], Zhang introduced two variations of probe interval graph. 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, all probe interval graphs are weakly chordal [21], and enhanced probe interval graphs are chordal [28, 29]. In full version of this draft, we show that the class of STS-probe interval graphs is equivalent to the class of convex graphs (hence the class is tree 3-spanner admissible), and the class of the (enhanced) probe interval graphs is incomparable with the classes of strongly chordal graphs and rooted directed path graphs.

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 that 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 in P or N [16], and the time complexity was improved to $O(m \log n)$ time by McConnell and Spinrad [19]. Those recognition algorithms construct within the same time bounds also 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.

Due to space limitation, some proofs are omitted. Full version of this draft is available at `http://www.komazawa-u.ac.jp/~uehara/ps/t-span.pdf`.

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 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 $v_0, v_1, \cdots, v_l$ is a *path*, denoted by $(v_0, v_1, \cdots, v_l)$, 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 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,

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

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

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

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 $N(S) = \cup_{v \in S} N(v)$ denote the *neighborhood* of S.

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 *[10] 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. Thus, for each positive integer k, a tree $2k$-spanner of a bipartite graph G is also a tree $(2k-1)$-spanner. Hence we will consider a tree t-spanner for each odd number t for bipartite graphs in this paper.

We here define graph classes dealt in this paper. See [7] for further details and references.

A graph (V, E) with $V = \{v_1, v_2, \cdots, v_n\}$ is an *interval graph* if there is a set of intervals $\mathcal{I} = \{I_1, I_2, \cdots, I_n\}$ 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 $\mathcal{I}$ *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 $X = \{x_1, x_2, \cdots, x_{n_1}\}$ and $Y = \{y_1, y_2, \cdots, y_{n_2}\}$ is an *interval bigraph* if there are families of intervals $\mathcal{I}_X = \{I_1, I_2, \cdots, I_{n_1}\}$ and $\mathcal{I}_Y = \{J_1, J_2, \cdots, J_{n_2}\}$ 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.

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 [22], where it was also shown that any interval bigraph is an ATE–free graph, and any bipartite ATE–free graph is chordal bipartite.

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 by $G = (P, N, E)$. Note that N is independent set, $G[P]$ is interval graph, and $G[P \cup \{v\}]$ is also 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 of v_1 and v_2 overlap u_1 and u_2. In the case, we can know that intervals I_{u_1} and I_{u_2} have to overlap without an experiment in chemistry. 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 [28, 21, 29] for further details.

3 NP-completeness for Chordal Bipartite Graphs

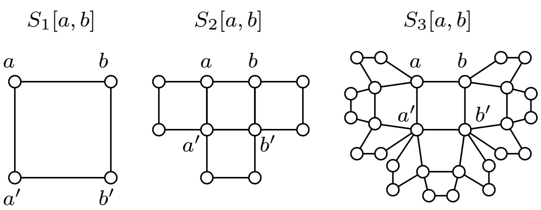

Figure 1: The graph $S_\ell[a, b]$

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 3SAT, 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 one 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 (Figure 1).

We will connect the vertices a and b to the other graphs, and use $S_\ell[a,b]$ as a subgraph of bigger graph. 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 $\{a,b\}$.*

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 $\{a,b\}$. 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 $d_T(a,b) \leq t-2\ell$.*

Observation 3 indicates a way to force an edge $\{x,y\}$ to be a tree edge: Choosing $\ell = \lfloor \frac{t-1}{2} \rfloor$ shows that $\{a,b\}$ must be an edge of 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 $1 \leq i \leq n$.

Definition 4 *In a graph G, an edge $\{a,b\}$ is said to be* forced *by an S_ℓ if $\{a,b\}$ appears in some $S_\ell[a,b]$ (as induced subgraph in G) such that $\{a,b\}$ disconnects $S_\ell[a,b]$ from the rest. We require that each two $S_\ell[a,b]$ and $S_{\ell'}[c,d]$ have at most 2 vertices in $\{a,b,c,d\}$ in common. An edge $\{a,b\}$ is said to be* strongly forced *if it is forced by two $S_k[a,b]$.*

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, and force each edge $\{a,b\} \in \{\{q_i,x_i^j\} : 1 \leq j \leq m\} \cup \{\{r_i,x_i^j\} : 1 \leq j \leq m\} \cup \{\{p_i,\overline{x_i}^j\} : 1 \leq j \leq m\} \cup \{\{s_i,\overline{x_i}^j\} : 1 \leq j \leq m\} \cup \{\{x_i^j,\overline{x_i}^{j'}\} : 1 \leq j,j' \leq m, j \neq j'\}$ 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 2 (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.

Definition 5 *A clause is* positive *(*negative*, respectively) if it contains only variables (negation of variables). A* definite *clause is one that is neither positive nor negative.*

For each clause C_j create the clause gadget $G(C_j)$ as follows. If C_j is a definite clause, $G(C_j)$ is a strongly forced edge $\{c_j^+,c_j^-\}$. If C_j is a positive or a negative clause, $G(C_j)$ is a 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 3.

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: (1) Connect every x_i^j with every $\overline{x_{i'}}^{j'}$ ($i \neq i'$). (Thus, the subgraph induced by the two independent sets $\{x_i^j : 1 \leq i \leq n, 1 \leq j \leq m\} \cup \{p,s\}$, and $\{\overline{x_i}^j : 1 \leq i \leq n, 1 \leq j \leq m\} \cup \{q,r\}$ *plus* the edge $\{p,q\}$ is a complete bipartite graph.) (2) For every definite 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 $S_{k-2}[x_i^j,c_j^+]$. 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 $S_{k-2}[\overline{x_i}^j,c_j^-]$. (3) 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 $S_{k-2}[c_j^-,r]$. (4) 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 $G = G(F)$

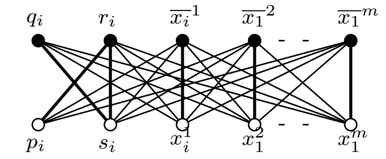

Figure 2: The gadget $G(x_i)$

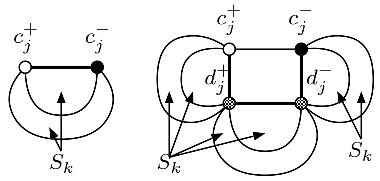

Figure 3: The $G(C_j)$ (definite: left, non-definite: right)

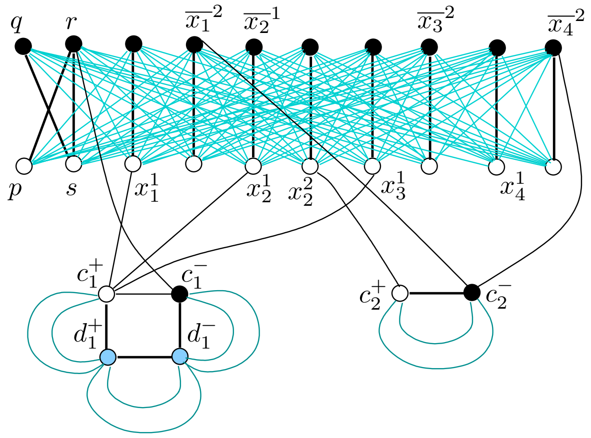

Figure 4: The reduction given $C_1 = (x_1, x_2, x_3)$ and $C_2 = (\overline{x_1}, x_2, \overline{x_4})$

is complete. Clearly, G can be constructed in polynomial time. See Figure 4 for an example.

Lemma 6 *G is chordal bipartite.*

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

Proof.(Outline) 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: (1) $\{p, r\}, \{r, s\}, \{s, q\}$, and $\{x_i^j, \overline{x_i}^j\}$ for $1 \le i \le n, 1 \le j \le m$, (2) $\{c_j^+, c_j^-\}$ for $1 \le j \le m$, where C_j is a definite clause, and (3) $\{c_j^+, d_j^+\}, \{d_j^+, d_j^-\}, \{d_j^-, c_j^-\}$ for $1 \le j \le m$, where C_j is a positive or a negative clause. Then we have the following three claims: (1) For every i and j, $\{q, x_i^j\} \notin E(T)$ and $\{p, \overline{x_i}^j\} \notin E(T)$. (2) For every i and j, exactly one of $\{r, x_i^j\}$ and $\{s, \overline{x_i}^j\}$ belongs to T. (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.

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

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

By (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 (2), $\{s, \overline{x_{i_1}}^j\}, \{s, \overline{x_{i_2}}^j\}$ and $\{s, \overline{x_{i_3}}^j\}$ are edges of T.

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 $d_T(c_j^+, x_{i_1}^j) = 7$. But 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.

Next, consider a definite clause $C_j = (x_{i_1}, x_{i_2}, \overline{x_{i_3}})$ (the other cases of a definite clause can be seen similarly). Recall that c_j^+ is adjacent to $x_{i_1}^j, x_{i_2}^j$ and c_j^- is adjacent to $\overline{x_{i_3}}^j$. Assume to the contrary that $f(C_j) = \texttt{false}$. That is, $\{r, x_{i_1}^j\}$ and $\{r, x_{i_2}^j\}$ do not belong to T and $\{r, x_{i_3}^j\}$ belongs to T.

By (2), $\{s, \overline{x_{i_1}}^j\}, \{s, \overline{x_{i_2}}^j\}$ are edges of T and $\{s, \overline{x_{i_3}}^j\} \notin E(T)$. Recall that the edge $\{c_j^+, c_j^-\}$ is an edge of T.

Now, since T is a tree, exactly one of the edges $\{c_j^+, x_{i_1}^j\}, \{c_j^+, x_{i_2}^j\}$, and $\{c_j^-, \overline{x_{i_3}}^j\}$ belongs to T. If $\{c_j^-, \overline{x_{i_3}}^j\} \in E(T)$ then $(c_j^+, c_j^-, \overline{x_{i_3}}^j, x_{i_3}^j, r, s, \overline{x_{i_1}}^j, x_{i_1}^j)$ is the $(c_j^+, x_{i_1}^j)$-path in T, hence $d_T(c_j^+, x_{i_1}^j) = 7$, contradicting Observation 3. If one of $\{c_j^+, x_{i_1}^j\}, \{c_j^+, x_{i_2}^j\}$ belongs to T, $\{c_j^+, x_{i_1}^j\} \in E(T)$, say, then $(c_j^-, c_j^+, x_{i_1}^j, \overline{x_{i_1}}^j, s, r, x_{i_3}^j, \overline{x_{i_3}}^j)$ is the $(c_j^-, \overline{x_{i_3}}^j)$-path in T, hence $d_T(c_j^-, \overline{x_{i_3}}^j) = 7$, contradicting Observation 3 again. Thus each clause C_j of F is satisfied by the assignment f, proving Lemma 7. ∎

Lemma 8 *Suppose F is satisfiable. Then G admits a tree $(2k+1)$-spanner.*

Theorem 9 *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 Asteroidal-Triple-Edge–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 [5].

Lemma 10 *[5] Any chordal bipartite graph G has a vertex with a maximum neighbor.*

It is easy to deduce from results of [4], [5] and [11] that a vertex with a maximum neighbor of a chordal bipartite graph can be found in linear time by the following procedure.

PROCEDURE 1. 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 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 11 *Let S be a connected component of a subgraph of G induced by set $V \setminus D_{k-1}(u)$ $(k \geq 1)$. Then, there is a vertex $w \in N_{k-1}(u)$ such that $N(w) \supset S \cap N_k(u)$.*

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

PROCEDURE 2. 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 $q := \max\{d_G(u, v) : v \in V\}$;
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 12 *Let $T = (V, E')$ be a spanning tree of a bipartite ATE–free graph $G = (V, E)$ output by PROCEDURE 2. Then, for any $x, y \in V$, we have $d_T(x, y) \leq 3 \cdot d_G(x, y)$ and $d_T(x, y) \leq d_G(x, y) + 2$.*

Since any interval bigraph is a bipartite ATE–free graph, we have the following corollary.

Corollary 13 *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.*

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 [19] in $O(m \log n)$ time, where $n = |P| + |N|$ and $m = |E|$). Let $\mathcal{I} = \{I_x : x \in P\}$ be the intervals in the interval model representing the probes and $\mathcal{J} = \{J_y : y \in N\}$ 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 [25]) that for any interval graph G and its arbitrary vertex u 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 2 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 case, we will 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 is always 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 2 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 13; it is an edge of G', too. 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 13, 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 will 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 $G_0 \cup G_1 \cup \ldots \cup G_i$ $(i < k)$ 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 $G_0 \cup G_1 \cup \ldots \cup G_i \cup G_{i+1}$ by either making all vertices of G_{i+1} adjacent in T^{i+1} to a common neighbor in $G_0 \cup G_1 \cup \ldots \cup G_i$ (if it exists) or by gluing trees T^i and T_{i+1} at a common vertex (see Figure 5 for an illustration). The details are omitted, and can be found in full version of this draft.

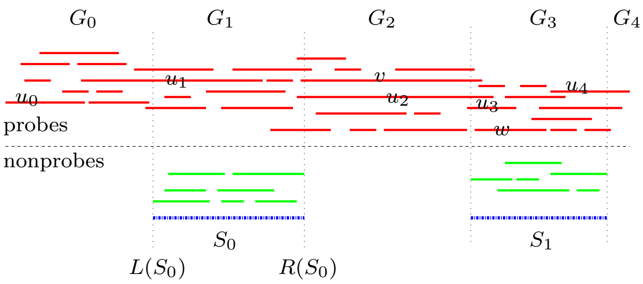

Figure 5: Segments and a decomposition of a probe interval graph

6 Concluding Remarks

In the paper, we have shown that the tree t-spanner problem is NP-complete even for chordal bipartite graphs for $k \geq 5$. 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.

References

[1] B. Awerbuch, A. Baratz, and D. Peleg. Efficient broadcast and light-weighted spanners. manuscript, 1992.

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

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

[4] A. Brandstädt, V.D. Chepoi, and F.F. Dragan. The Algorithmic Use of Hypertree Structure and Maximum Neighbourhood Orderings. *Disc. Appl. Math.*, 82:43–77, 1998.

[5] A. Brandstädt, F. Dragan, V. Chepoi, and V. Voloshin. Dually Chordal Graphs. *SIAM J. Disc. Math.*, 11(3):437–455, 1998.

[6] A. Brandstädt, F.F. Dragan, H.-O. Le, and V.B. Le. Tree Spanners on Chordal Graphs: Complexity, Algorithms, Open Problems. In *ISAAC 2002*, pages 163–174. LNCS Vol. 2518, Springer-Verlag, 2002.

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

[8] A. Brandstädt, J. Spinrad, and L. Stewart. Bipartite Permutation Graphs are Bipartite Tolerance Graphs. *Congr. Numer.*, 58:165–174, 1987.

[9] L. Cai and D.G. Corneil. Tree Spanners: an Overview. *Congr. Numer.*, 88:65–76, 1992.

[10] L. Cai and D.G. Corneil. Tree Spanners. *SIAM J. Disc. Math.*, 8(3):359–387, 1995.

[11] F.F. Dragan and V.I. Voloshin. Incidence Graphs of Biacyclic Hypergraphs. *Disc. Appl. Math.*, 68:259–266, 1996.

[12] M. Farber. Characterization of Strongly Chordal Graphs. *Disc. Math.*, 43:173–189, 1983.

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

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

[15] F. Harary, J.A. Kabell, and F.R. McMorris. Bipartite intersection graphs. *Comment. Math. Univ. Carolin.*, 23:739–745, 1982.

[16] J.L. Johnson and J.P. Spinrad. A Polynomial Time Recognition Algorithm for Probe Interval Graphs. In *12th SODA*, pages 477–486. ACM, 2001.

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

[18] M.S. Madanlal, G. Venkatesan, and C. P. Rangan. Tree 3-Spanners on Interval, Permutation and Regular Bipartite Graphs. *Inf. Proc. Let.*, 59:97–102, 1996.

[19] R.M. McConnell and J.P. Spinrad. Construction of Probe Interval Models. In *13th SODA*, pages 866–875. ACM, 2002.

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

[21] F.R. McMorris, C. Wang, and P. Zhang. On Probe Interval Graphs. *Disc. Appl. Math.*, 88:315–324, 1998.

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

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

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

[25] E. Prisner. Distance approximating spanning trees. In *STACS'97*, pages 499–510. LNCS Vol. 1200, Springer-Verlag, 1997.

[26] J. Soares. Graph Spanners: a Survey. *Congr. Numer.*, 89(225–238):225–238, 1992.

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

[28] P. Zhang. Prove Interval Graphs and Its Applications to Physical Mapping of DNA. manuscript, 1994.

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