Available online at www.sciencedirect.com
SCIENCE DIRECT®

Theoretical Computer Science 347 (2005) 54–75

Theoretical Computer Science

www.elsevier.com/locate/tcs

Additive sparse spanners for graphs with bounded length of largest induced cycle☆

Victor D. Chepoi[a], Feodor F. Dragan[b,*], Chenyu Yan[b]

[a]*Laboratoire d'Informatique Fondamentale, Université Aix-Marseille II, France*
[b]*Department of Computer Science, Kent State University, Kent, OH 44242, U.S.A.*

Received 14 January 2003; received in revised form 18 May 2004; accepted 26 May 2005

Communicated by G.F. Italiano

Abstract

In this paper, we show that every chordal graph with n vertices and m edges admits an additive 4-spanner with at most $2n - 2$ edges and an additive 3-spanner with at most $O(n \log n)$ edges. This significantly improves results of Peleg and Schäffer from [Graph Spanners, *J. Graph Theory* 13 (1989) 99–116]. Our spanners are additive and easier to construct. An additive 4-spanner can be constructed in linear time while an additive 3-spanner is constructable in $O(m \log n)$ time. Furthermore, our method can be extended to graphs with largest induced cycles of length k. Any such graph admits an additive $(k + 1)$-spanner with at most $2n - 2$ edges which is constructable in $O(n k + m)$ time.

Keywords: Additive graph spanners; Efficient algorithms; k-chordal graphs; Distance

1. Introduction

Let $G = (V, E)$ be a connected graph with n vertices and m edges. The *length* of a path from a vertex v to a vertex u in G is the number of edges in the path. The distance $d_G(u, v)$ between vertices u and v is the length of a shortest (u, v)-path of G. We say that

☆ In memory of Prof. Chiril F. Prisacaru.

* Corresponding author. Tel.: +1 330 6729058; fax: +1 330 6727824.
E-mail addresses: chepoi@lim.univ-mrs.fr (V.D. Chepoi), dragan@cs.kent.edu (F.F. Dragan), cyan1@kent.edu (C. Yan).

doi:10.1016/j.tcs.2005.05.017

a graph $H = (V, E')$ is *an additive r-spanner* (*a multiplicative t-spanner*) of G, if $E' \subseteq E$ and $d_H(x, y) - d_G(x, y) \leqslant r$ ($d_H(x, y)/d_G(x, y) \leqslant t$, respectively) holds for any pair of vertices $x, y \in V$ (here, $t \geqslant 1$ and $r \geqslant 0$ are real numbers). We refer to r (to t) as the *additive* (respectively, *multiplicative*) *stretch factor* of H. Clearly, every additive r-spanner of G is a multiplicative $(r + 1)$-spanner of G (but not vice versa).

There are many applications of spanners in various areas; especially, in distributed systems and communication networks. In [26], close relationships were established between the quality of spanners (in terms of stretch factor and the number of spanner edges $|E'|$), and the time and communication complexities of any synchronizer for the network based on this spanner. Also sparse spanners are very useful in message routing in communication networks; in order to maintain succinct routing tables, efficient routing schemes can use only the edges of a sparse spanner [27]. Unfortunately, the problem of determining, for a given graph G and two integers $t, m \geqslant 1$, whether G has a t-spanner with m or fewer edges, is NP-complete (see [25]).

The sparsest spanners are tree spanners. Tree spanners occur in biology [2], and as it was shown in [24], they can be used as models for broadcast operations. Multiplicative tree t-spanners were considered in [9]. It was shown that, for a given graph G, the problem to decide whether G has a multiplicative tree t-spanner is NP-complete for any fixed $t \geqslant 4$ and is linearly solvable for $t = 1, 2$ (the status of the case $t = 3$ is open for general graphs). Also, in [14], NP-completeness results were presented for tree spanners on planar graphs.

Many particular graph classes, such as cographs, complements of bipartite graphs, split graphs, regular bipartite graphs, interval graphs, permutation graphs, convex bipartite graphs, distance–hereditary graphs, directed path graphs, cocomparability graphs, AT-free graphs, strongly chordal graphs and dually chordal graphs admit additive tree r-spanners and/or multiplicative tree t-spanners for sufficiently small r and t (see [4,8,17,18,21,32,28, 29]). We refer also to [1,3,6,8,9,19,22–25,30] for more background information on tree and general sparse spanners.

In this paper we are interested in finding sparse spanners with small additive stretch factors in chordal graphs and their generalizations. A graph G is *chordal* [16] if its largest induced (chordless) cycles are of length 3. A graph is *k-chordal* if its largest induced cycles are of length k.

The class of chordal graphs does not admit good tree spanners. As it was mentioned in [28,29], Le and McKee have independently showed that for every fixed integer t there is a chordal graph without tree t-spanners (additive, as well as multiplicative). Recently, Brandstädt et al. [6] have showed that, for any $t \geqslant 4$, the problem to decide whether a given chordal graph G admits a multiplicative tree t-spanner is NP-complete even when G has the diameter at most $t + 1$ (t is even), respectively, at most $t + 2$ (t is odd). Thus, the only hope for chordal graphs is to get sparse (with $O(n)$ edges) small stretch factor spanners. Peleg and Schäffer have already showed in [25] that any chordal graph admits a multiplicative 5-spanner with at most $2n - 2$ edges and a multiplicative 3-spanner with at most $O(n \log n)$ edges. Both spanners can be constructed in polynomial time.

In this paper we improve those results. We show that every chordal graph admits an additive 4-spanner with at most $2n - 2$ edges and an additive 3-spanner with at most $O(n \log n)$ edges. Our spanners are not only additive but also easier to construct. An additive 4-spanner can be constructed in linear time while an additive 3-spanner is constructable in

$O(m \log n)$ time. Furthermore, our method can be extended to all k-chordal graphs. Any such graph admits an additive $(k+1)$-spanner with at most $2n-2$ edges which is constructable in $O(n\,k+m)$ time. Note that the method from [25] essentially uses the characteristic *clique trees* of chordal graphs and therefore cannot be extended (at least directly) to general k-chordal graphs for $k \geqslant 4$.

In obtaining our results we essentially relayed on ideas developed in papers [4,10,11,25].

2. Preliminaries

All graphs occurring in this paper are connected, finite, undirected, loopless, and without multiple edges. For each integer $l \geqslant 0$, let $B_l(u)$ denote the *ball* of radius l centered at u:

$$B_l(u) = \{v \in V : d_G(u, v) \leqslant l\}.$$

Let $N_l(u)$ denote the *sphere* of radius l centered at u:

$$N_l(u) = \{v \in V : d_G(u, v) = l\}.$$

$N_l(u)$ is also called the lth *neighborhood* of u. A *layering* of G with respect to some vertex u is a partition of V into the spheres $N_l(u), l = 0, 1, \ldots$. By $N(u)$ we denote the *neighborhood* of u, i.e., $N(u) = N_1(u)$. More generally, for a subset $S \subseteq V$ let $N(S) = \bigcup_{u \in S} N(u)$.

Let $\sigma = [v_1, v_2, \ldots, v_n]$ be any ordering of the vertex set of a graph G. We will write $a < b$ whenever in a given ordering σ vertex a has a smaller number than vertex b. Moreover, $\{a_1, \ldots, a_l\} < \{b_1, \ldots, b_k\}$ is an abbreviation for $a_i < b_j$ $(i = 1, \ldots, l;\ j = 1, \ldots, k)$. In this paper, we will use two kind of orderings, namely, BFS-orderings and LexBFS-orderings.

In a *breadth-first search* (BFS), started at vertex u, the vertices of a graph G with n vertices are numbered from n to 1 in decreasing order. The vertex u is numbered by n and put on an initially empty queue of vertices. Then a vertex v at the head of the queue is repeatedly removed, and neighbors of v that are still unnumbered are consequently numbered and placed onto the queue. Clearly, BFS operates by proceeding vertices in layers: the vertices closest to the start vertex are numbered first, and most distant vertices are numbered last. BFS may be seen to generate a rooted tree T with vertex u as the root. We call T the *BFS-tree* of G. A vertex v is the *father* in T of exactly those neighbors in G which are inserted into the queue when v is removed. An ordering σ generated by a BFS will be called a *BFS-ordering* of G. Denote by $f(v)$ the father of a vertex v with respect to σ. The following properties of a BFS-ordering will be used in what follows.

(P1) If $x \in N_i(u)$, $y \in N_j(u)$ and $i < j$, then $x > y$ in σ.

(P2) If $v \in N_q(u) (q > 0)$ then $f(v) \in N_{q-1}(u)$ and $f(v)$ is the vertex from $N(v)$ with the largest number in σ.

(P3) If $x > y$, then either $f(x) > f(y)$ or $f(x) = f(y)$.

Lexicographic breadth-first search (LexBFS), started at a vertex u, orders the vertices of a graph by assigning numbers from n to 1 in the following way. The vertex u gets the number n. Then each next available number k is assigned to a vertex v (as yet unnumbered) which has lexically largest vector $(s_n, s_{n-1}, \ldots, s_{k+1})$, where $s_i = 1$ if v is adjacent to the vertex numbered i, and $s_i = 0$ otherwise. An ordering of the vertex set of a graph generated by LexBFS we will call a *LexBFS-ordering*. Clearly any LexBFS-ordering is a BFS-ordering (but not conversely). Note also that for a given graph G, both a BFS-ordering and a LexBFS-ordering can be generated in linear time [16]. LexBFS-ordering has all the properties of the BFS-ordering. In particular, we can associate a tree T rooted at v_n with every LexBFS-ordering $\sigma = [v_1, v_2, \dots, v_n]$ simply connecting every vertex v ($v \neq v_n$) to its neighbor $f(v)$ with the largest number in σ. We call this tree a *LexBFS-tree* of G rooted at v_n and vertex $f(v)$ *the father* of v in T. Besides the three properties of BFS-ordering, LexBFS-ordering has additionally the following property [16].

(P4) If $a < b < c$ and $ac \in E$ and $bc \notin E$ then there exists a vertex d such that $c < d$, $db \in E$ and $da \notin E$.

3. Spanners for chordal graphs

3.1. Additive 4-spanners with O(n) edges

For a chordal graph $G = (V, E)$ and a vertex $u \in V$, consider a BFS of G started at u and let $q = \max\{d_G(u, v) : v \in V\}$. For a given k, $0 \leqslant k \leqslant q$, let $S_1^k, S_2^k, \dots, S_{p_k}^k$ be the connected components of a subgraph of G induced by the kth neighborhood of u. In [4], there was defined a graph Γ whose vertices are the connected components S_i^k, $k = 0, 1, \dots, q$ and $i = 1, \dots, p_k$. Two vertices S_i^k, S_j^{k-1} are adjacent if and only if, there is an edge of G with one end in S_i^k and another end in S_j^{k-1}. Before we describe our construction of the additive 4-spanner $H = (V, E')$ for a chordal graph G, first we recall two important lemmas.

Lemma 1 (*Brandstädt et al. [4]*). *Let G be a chordal graph. For any connected component S of the subgraph of G induced by $N_k(u)$, the set $N(S) \cap N_{k-1}(u)$ induces a complete subgraph.*

Proof. Consider two arbitrary vertices $x, y \in N(S) \cap N_{k-1}(u)$ and assume that they are not adjacent. Then, since x and y can be connected by an induced path inner vertices of which are outside the ball $B_{k-1}(u)$ and by an induced path inner vertices of which are inside the ball $B_{k-2}(u)$, and both those paths are of length at least 2, we obtain an induced cycle of length at least 4 in G, which is impossible. □

Lemma 2 (*Brandstädt et al. [4]*). *Γ is a tree.*

Now, to construct H, we choose an arbitrary vertex $u \in V$ and perform a BFS in G started at u. Let $\sigma = [v_1, \dots, v_n]$ be a BFS-ordering of G. The construction of H is completed according to the following algorithm (for an illustration see Fig. 1).

Procedure 1. Additive 4-spanners for chordal graphs
Input: A chordal graph $G = (V, E)$ with BFS-ordering σ, and connected components $S_1^k, S_2^k, \dots, S_{p_k}^k$ for any k, $0 \leqslant k \leqslant q$, where $q = \max\{d_G(u, v) : v \in V\}$.

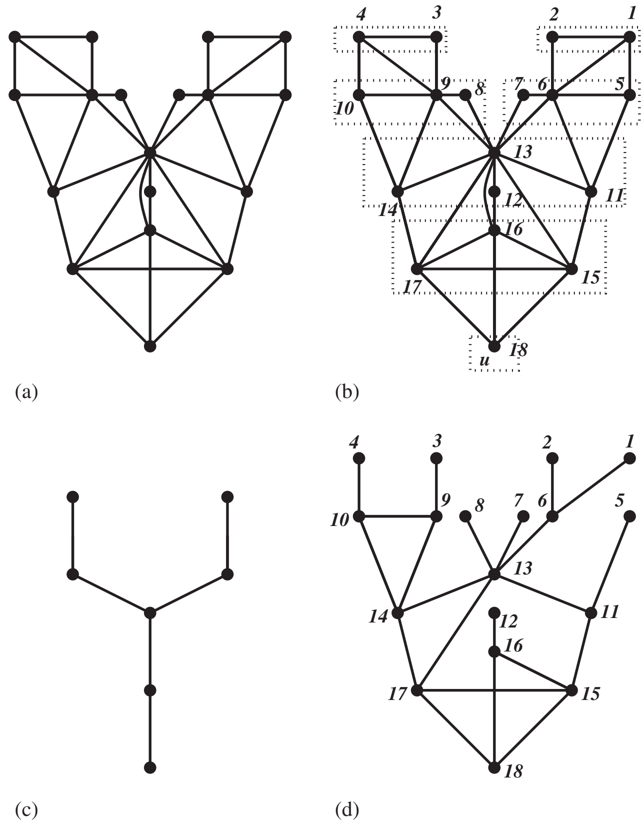

Fig. 1. (a) A chordal graph G. (b) A BFS-ordering σ, BFS-tree T associated with σ and a layering of G. (c) The tree Γ of G associated with that layering. (d) Additive 4-spanner (actually, additive 3-spanner) H of G constructed by Procedure 1 (five edges are added to the BFS-tree T).

Output: A spanner $H = (V, E')$ of G.
Method:

$E' = \emptyset$;
for $k = q$ **downto** 1 **do**
  for $j = 1$ **to** p_k **do**
    $M = \emptyset$;
    for each vertex $v \in S_j^k$ add edge $vf(v)$ to E' and vertex $f(v)$ to M;
    pick vertex $c \in M$ with the minimum number in σ;
    for every vertex $x \in M \setminus \{c\}$ add edge xc to E';
return $H = (V, E')$.

Lemma 3. *H is an additive* 4-*spanner for G.*

Proof. Consider nodes S_i^l and S_j^m of the tree Γ and their lowest common ancestor S_m^p in Γ. For any two vertices $x \in S_i^l$ and $y \in S_j^m$ of G, we have

$$d_G(x, y) \geqslant l - p + m - p,$$

because any path of G connecting x and y must pass S_m^p (since Γ is a tree).

From our construction of H (for every vertex v of G the edge $vf(v)$ is present in H), we can easily show that there exist vertices $x', y' \in S_m^p$ such that

$$d_H(x, x') = l - p,$$

$$d_H(y, y') = m - p.$$

Hence, we only need to show that

$$d_H(x', y') \leqslant 4.$$

If $x' = y'$ then we are done. If vertices x' and y' are distinct, then by Lemma 1, $N(S_m^p) \cap N_{p-1}(u)$ is a clique of G. According to Procedure 1, fathers of both vertices x' and y' are in M and they are connected in H by a path of length at most 2 via vertex c of M. Therefore, $d_H(x', y') \leqslant d_H(x', f(x')) + d_H(f(x'), f(y')) + d_H(f(y'), y') \leqslant 1 + 2 + 1 = 4$. This concludes our proof. □

We can easily show that the bound given in Lemma 3 is tight. For a chordal graph presented in Fig. 2, we have $d_G(y, b) = 1$. The spanner H of G constructed by our method is shown with bold edges. In H we have $d_H(y, b) = 5$. Therefore, $d_H(y, b) - d_G(y, b) = 4$.

Lemma 4. *If G has n vertices, then H contains at most* $2n - 2$ *edges.*

Proof. The edge set of H consists of two sets E_1 and E_2, where E_1 are those edges connecting two vertices between two different layers (edges of type $vf(v)$) and E_2 are those edges which have been used to build a star for a clique M inside a layer (edges of type $cf(v)$). Obviously, E_1 has exactly $n - 1$ edges; actually, they are the edges of the BFS-tree of G. For each connected component S_i^l of size s, we have at most s vertices in M. Therefore, while proceeding component S_i^l, at most $s - 1$ edges are added to E_2. The total size of all the connected components is at most n, so E_2 contains at most $n - 1$ edges. Hence, the graph H contains at most $2n - 2$ edges. □

Lemma 5. *H can be constructed in linear* $O(n + m)$ *time.*

Proof. A BFS-tree of G can be constructed in linear time. To construct H, we only need to find the connected components of the kth neighborhood of u $(1 \leqslant k \leqslant q)$ and for each component compute the set M and build a star on M. Since the size of M is not larger than the size of that connected component, all these can be done in $O(n + m)$ total time (for all ks). Hence, the construction of H can easily be done in total $O(n + m)$ time. □

Combining Lemmas 3–5 we get the following result.

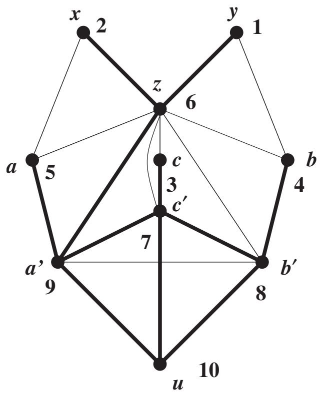

Fig. 2. A chordal graph with a BFS-ordering which shows that the bound given in Lemma 3 is tight. We have $d_H(y,b) - d_G(y,b) = 4$ and $d_H(y,b)/d_G(y,b) = 5$.

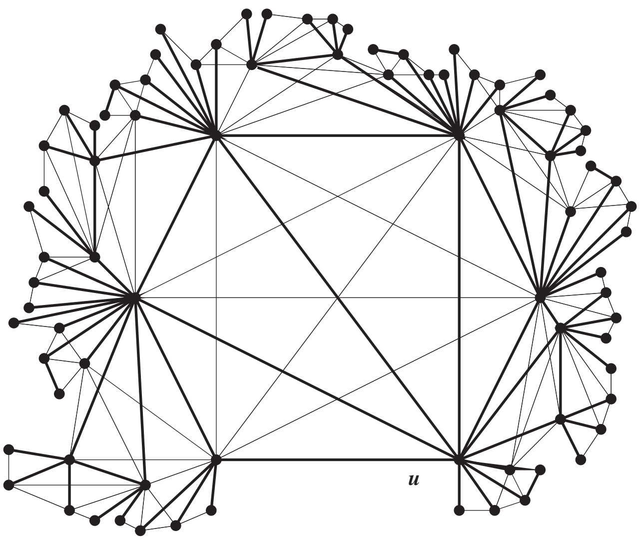

Fig. 3. A chordal graph and its additive 3-spanner constructed by Procedure 1. The spanner is shown with dark edges, it has 80 vertices and 90 edges.

Theorem 1. *Every n-vertex chordal graph $G = (V, E)$ admits an additive 4-spanner with at most $2n - 2$ edges. Moreover, such a sparse spanner of G can be constructed in linear time.*

Fig. 3 presents a chordal graph with its additive 3-spanner constructed according to Procedure 1.

Note that any additive 4-spanner is a multiplicative 5-spanner. As we mentioned earlier, the existence of multiplicative 5-spanners with at most $2n - 2$ edges in chordal graphs was

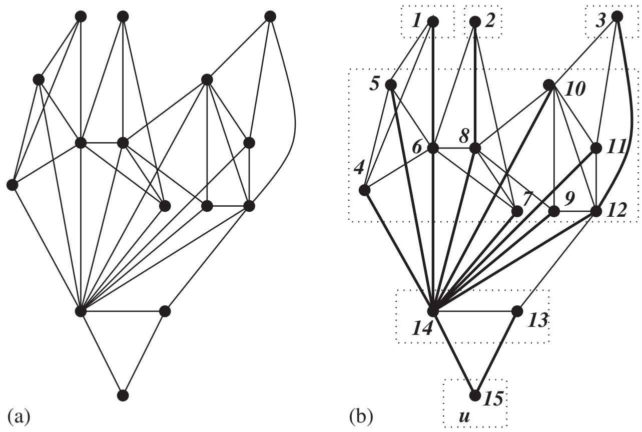

Fig. 4. (a) A chordal graph G. (b) A LexBFS-ordering σ, LexBFS-tree associated with σ and a layering of G.

already shown in [25], but their method of constructing such spanners is more complicated than ours.

3.2. *Additive 3-spanners with* $O(n \log n)$ *edges*

To construct an additive 3-spanner for a chordal graph $G = (V, E)$, first we get a LexBFS-ordering σ of the vertices of G (see Fig. 4). Then, we construct an additive 4-spanner $H = (V, E_1 \cup E_2)$ for G using the algorithm from Section 3.1. Finally, we update H by adding some more edges. In what follows, we will need the following known result.

Theorem 2 (*Gilbert et al. [15]*). *Every n-vertex chordal graph G contains a maximal clique C such that if the vertices in C are deleted from G, every connected component in the graph induced by any remaining vertices is of size at most* $n/2$.

An $O(n + m)$ algorithm for finding such a separating clique C is also given in [15].

As before, for a given k, $0 \leqslant k \leqslant q$, let $S_1^k, S_2^k, \ldots, S_{p_k}^k$ be the connected components of a subgraph of G induced by the kth neighborhood of u. For each connected component S_i^k (which is obviously a chordal graph), we run the following algorithm which is similar to the algorithm in [25] (see also [24]), where a method for construction of a multiplicative 3-spanner for a chordal graph is described. The only difference is that we run that algorithm on every connected component from each layer of G instead of on the whole graph G. For the purpose of completeness, we present the algorithm here (for an example see Fig. 5).

Procedure 2. A balanced clique tree for a connected component S_i^k
Input: A subgraph Q of G induced by a connected component S_i^k.
Output: A balanced clique tree for Q.
Method:
if Q is a clique then the balanced clique tree $T(Q)$ is a one-node tree

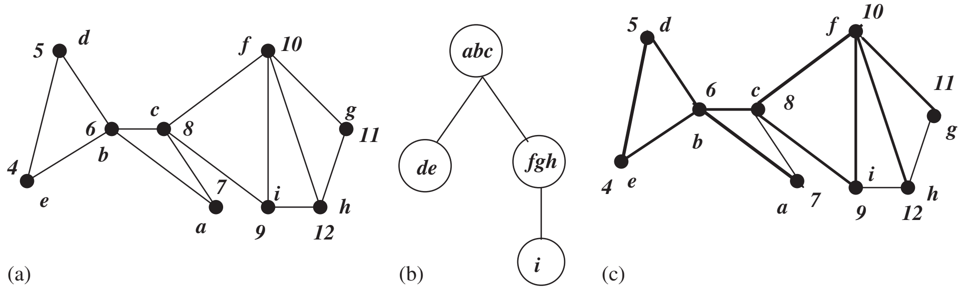

Fig. 5. (a) A chordal graph induced by set S_1^2 of the graph G presented in Fig. 4, (b) its balanced clique tree and (c) edges of $E_3(S_1^2) \cup E_4(S_1^2)$.

else
find a maximum separating clique C of the graph Q as prescribed in Theorem 2;
suppose C partitions the rest of Q into connected components $\{Q_1, \ldots, Q_r\}$;
for each Q_i, construct a balanced clique tree $T(Q_i)$ recursively;
construct $T(Q)$ by taking C to be the root and connecting the root of each tree $T(Q_i)$ as a child of C.

The nodes of the final balanced tree for S_i^k (denote it by $T(S_i^k)$) represent a certain collection of disjoint cliques $\{C_i^k(1), \ldots, C_i^k(s_i^k)\}$ that cover entire set S_i^k (see Fig. 5 for an illustration). For each clique $C_i^k(j)$ $(1 \leqslant j \leqslant s_i^k)$ we build a star centered at its vertex with the minimum number in LexBFS-ordering σ. We use $E_3(i,k)$ to denote this set of star edges. Evidently, $|E_3(i,k)| \leqslant |S_i^k| - 1$. Consider a clique $C_i^k(j)$ in S_i^k. For each vertex v of $C_i^k(j)$ and each clique $C_i^k(j')$ on the path of balanced clique tree $T(S_i^k)$ connecting node $C_i^k(j)$ with the root, if v has a neighbor in $C_i^k(j')$ (i.e., there exists an edge of G between v and a vertex of $C_j^k(j')$), then select one such neighbor w and put the edge vw of G into set $E_4(i,k)$ (initially $E_4(i,k)$ is empty). We do this for every clique $C_i^k(j)$, $j \in \{1, \ldots, s_i^k\}$. Since the depth of the tree $T(S_i^k)$ is at most $\log_2 |S_i^k| + 1$ (see [25,24]), any vertex v from S_i^k may contribute at most $\log_2 |S_i^k|$ edges to $E_4(i,k)$. Therefore, $|E_4(i,k)| \leqslant |S_i^k| \log_2 |S_i^k|$.

Define now two sets of edges in G, namely,

$$E_3 = \bigcup_{k=1}^{q} \bigcup_{i=1}^{p_k} E_3(i,k)$$

and

$$E_4 = \bigcup_{k=1}^{q} \bigcup_{i=1}^{p_k} E_4(i,k),$$

and consider a spanning subgraph $H^* = (V, E_1 \cup E_2 \cup E_3 \cup E_4)$ of G (see Fig. 6). Recall that $E_1 \cup E_2$ is the set of edges of an additive 4-spanner H constructed for G by Procedure 1 (see Section 3.1).

The following lemmas for H^* hold.

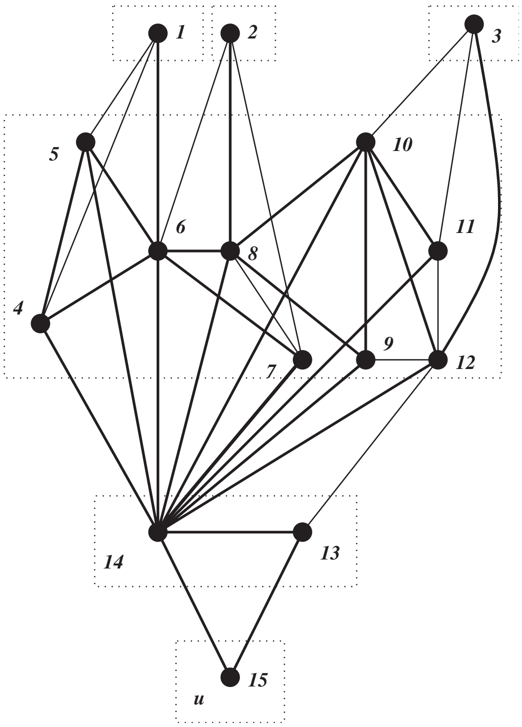

Fig. 6. An additive 3-spanner H^* of graph G presented in Fig. 4.

Lemma 6. *If G has n vertices, then* H^* *has at most* $O(n \log n)$ *edges.*

Proof. We know already that $|E_1| + |E_2| \leqslant 2n - 2$. Also,

$$|E_3| = \sum_{k=1}^{q} \sum_{i=1}^{p_k} |E_3(i,k)| \leqslant \sum_{k=1}^{q} \sum_{i=1}^{p_k} (|S_i^k| - 1) \leqslant n - 1$$

and

$$|E_4| = \sum_{k=1}^{q} \sum_{i=1}^{p_k} |E_4(i,k)| \leqslant \sum_{k=1}^{q} \sum_{i=1}^{p_k} (|S_i^k| \log_2 |S_i^k|) \leqslant n \log_2 n.$$

Hence, the total number of edges in H^* is at most $O(n \log n)$. $\square$

To prove that H^* is an additive 3-spanner for G, we will need the following auxiliary lemmas.

Lemma 7 (*Golumbic [16]*). *Let G be a chordal graph and σ be a LexBFS-ordering of G. Then, σ is a perfect elimination ordering of G, i.e., for any vertices a, b, c of G such that $a < \{b, c\}$ and $ab, ac \in E(G)$, vertices b and c must be adjacent.*

Note that in Lemma 7 LexBFS-ordering cannot be replaced with simple BFS-ordering. There are chordal graphs (e.g., two triangles sharing a common edge) and their BFS-orderings which do not produce perfect elimination orderings. The special feature of LexBFS-orderings on chordal graphs (stated in Lemma 7) is used in Lemma 9 to guarantee that H^* indeed is an additive 3-spanner for G.

Lemma 8 (*Dragan [11]*). *Let G be an arbitrary graph and $T(G)$ be a BFS-tree of G with the root u. Let also v be a vertex of G and w ($w \neq v$) be an ancestor of v in $T(G)$ from layer $N_i(u)$. Then, for any vertex $x \in N_i(u)$ with $d_G(v, w) = d_G(v, x)$, inequality $x \leqslant w$ holds.*

Lemma 9. *H^* is an additive 3-spanner for G.*

Proof. As in the proof of Lemma 3, consider again nodes S_i^k and S_j^l of the tree Γ and their lowest common ancestor S_m^p in Γ. Then $d_G(x, y) \geqslant k - p + l - p$ holds for any two vertices $x \in S_i^k$ and $y \in S_j^l$ and there must exist vertices $x', y' \in S_m^p$ such that

$$d_{H^*}(x, x') = k - p,$$

$$d_{H^*}(y, y') = l - p.$$

So, to have $d_{H^*}(x, y) - d_G(x, y) \leqslant 3$, we only need to show $d_{H^*}(x', y') \leqslant 3$. We may assume $x' \neq y'$.

First note that, since additive 4-spanner H is a subgraph of H^*, $d_{H^*}(x', y') \leqslant 4$ holds (see the proof of Lemma 3). Hence, if $d_G(x, y) > k - p + l - p$, then $d_{H^*}(x, y) - d_G(x, y) \leqslant 3$ and we are done.

We may assume now that $d_G(x, y) = k - p + l - p$ and, therefore, there exists a vertex z in S_m^p such that $d_G(x, z) = k - p$ and $d_G(y, z) = l - p$ (z is a vertex of a shortest path connecting x and y in G). Let S_t^{p+1}, S_r^{p+1} be the two connected components on the paths of Γ between S_m^p, S_i^k and S_m^p, S_j^l, respectively. From $x', z \in N(S_t^{p+1}) \cap N_p(u)$, we conclude that x' and z either coincide or are adjacent in G (see Lemma 1). Similarly, $d_G(y', z) \leqslant 1$. Since $x', y', z \in N_p(u)$, $d_G(x, x') = d_G(x, z) = k - p$, $d_G(y, y') = d_G(y, z) = l - p$, and x' is an ancestor of x and y' is an ancestor of y in the LexBFS-tree associated with σ, by Lemma 8, $z \leqslant \{x', y'\}$.

We claim that x' and y' are adjacent in G. Indeed, if z coincides with x' or y', then $x'y' \in E(G)$ because $d_G(z, x') \leqslant 1$ and $d_G(z, y') \leqslant 1$. If z is distinct from both x' and y', then the inequality $z < \{x', y'\}$ together with $zx', zy' \in E(G)$ imply $x'y' \in E(G)$ (by Lemma 7). Thus, $x'y' \in E(G)$.

Now, if x' and y' are in one clique $C_m^p(t)$ (for some t), then, since in H^* vertices of $C_m^p(t)$ are connected by a star, $d_{H^*}(x', y') \leqslant 2$ must hold.

If x' and y' are in cliques $C_m^p(t)$ and $C_m^p(r)$, respectively (for some t, r), then one of these cliques is descendent of the other in the balanced clique tree $T(S_m^p)$. This is because for two cliques $C_m^p(t)$ and $C_m^p(r)$ that have no descendence relationship in the tree $T(S_m^p)$, the clique $C_m^p(h)$ that is their lowest common ancestor in the tree separates the vertices of $C_m^p(t)$ and $C_m^p(r)$, hence no edge is possible between them.

Assuming, without loss of generality, that $C_m^p(t)$ is an ancestor of $C_m^p(r)$ in the tree $T(S_m^p)$, y' is connected in H^* to some vertex x'' of $C_m^p(t)$. Since vertices of $C_m^p(t)$ are connected by a star in H^*, we get $d_{H^*}(x', y') \leqslant d_{H^*}(x', x'') + d_{H^*}(x'', y') \leqslant 2 + 1 = 3$.

Thus, in any cases we have $d_{H^*}(x', y') \leqslant 3$. □

Lemma 10. *If a chordal graph G has n vertices and m edges, then its additive* 3*-spanner* H^* *can be constructed in* $O(m \log n)$ *time.*

Proof. As it was shown in Section 3.1, the additive 4-spanner can be constructed in $O(n+m)$ time. For each connected component S_i^k with $n_{i,k}$ vertices and $m_{i,k}$ edges, its balanced clique tree can be constructed in $O(m_{i,k} \log n_{i,k})$ time. To build a star on each clique $C_i^k(j)$ and find all edges of $E_4(i,k)$, we need at most $\sum_{v \in S_i^k} (\deg_G(v) \log n_{i,k})$ time.

So, the total time needed to construct H^* is

$$O(n+m) + \sum_{k=1}^{q} \sum_{i=1}^{p_k} (O(m_{i,k} \log n_{i,k}) + \sum_{v \in S_i^k} (\deg_G(v) \log n_{i,k}))$$
$$= O(m \log n).$$

This concludes the proof. □

The main result of this subsection is the following.

Theorem 3. *Every chordal graph* $G = (V, E)$ *with n vertices and m edges admits an additive* 3*-spanner with at most* $O(n \log n)$ *edges. Moreover, such a sparse spanner of G can be constructed in* $O(m \log n)$ *time.*

In [25], it was shown that any chordal graph admits a multiplicative 3-spanner H' with at most $O(n \log n)$ edges which is constructable in $O(m \log n)$ time. But, it is worth to note that the spanner H' gives a better than H^* approximation of distances only for adjacent in G vertices. For pairs $x, y \in V$ at distance at least 2 in G, the multiplicative stretch factor given by H^* is at most 2.5 which is better than the multiplicative stretch factor of at most 3 given by H'.

4. Spanners for *k*-chordal graphs

Let u be an arbitrary vertex of a k-chordal graph $G = (V, E)$, σ be a BFS-ordering of G and T be the BFS-tree associated with σ. For each $l \geqslant 0$ define a graph Q^l with the lth sphere $N_l(u)$ as a vertex set. Two vertices $x, y \in N_l(u)$ $(l \geqslant 1)$ are adjacent in Q^l if and only if they can be connected by a path outside the ball $B_{l-1}(u)$. Let $Q_1^l, \ldots, Q_{p_l}^l$ be all the connected components of Q^l. Similar to chordal graphs and as shown in [10] we define a graph Γ whose vertex-set is the collection of all connected components of the graphs Q^l,

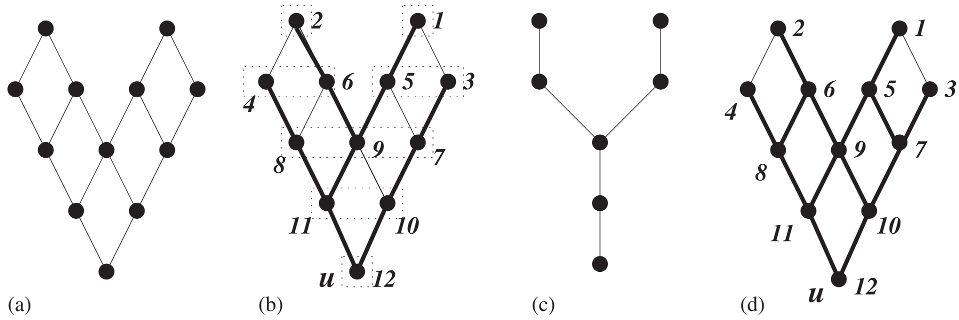

Fig. 7. (a) A 4-chordal graph G. (b) A BFS-ordering σ, BFS-tree associated with σ and a layering of G. (c) The tree Γ of G associated with that layering. (d) Additive 5-spanner (actually, additive 2-spanner) H of G constructed by Procedure 3.

$l = 0, 1, \ldots,$ and two vertices are adjacent in Γ if and only if there is an edge of G between the corresponding components. The following lemma holds.

Lemma 11 (*Chepoi and Dragan [10]*). *Γ is a tree*.

To construct our spanner H for G, we use the following procedure (for an illustration see Fig. 7).

Procedure 3. Additive $(k+1)$-spanners for k-chordal graphs
Input: A k-chordal graph $G = (V, E)$ with a BFS-ordering σ, and connected components $Q_1^l, Q_2^l, \ldots, Q_{p_l}^l$ for any l, $0 \leqslant l \leqslant q$, where $q = \max\{d_G(u, v) : v \in V\}$.
Output: A spanner $H = (V, E')$ of G.
Method:

$E' = \emptyset$;
for $l = q$ **downto** 1 **do**
  for $j = 1$ **to** p_l **do**
    for each vertex $v \in Q_j^l$ add $vf(v)$ to E';
    pick vertex c in Q_j^l with the minimum number in σ;
    for each $v \in Q_j^l \setminus \{c\}$ **do**
      connected = FALSE;
      while connected = FALSE **do**
      /* this while loop works at most $\lfloor k/2 \rfloor$ times for each v */
        if $vc \in E(G)$ **then**
          add vc to E';
          connected = TRUE;
        else if $vf(c) \in E(G)$ **then**
            add $vf(c)$ to E';
            connected = TRUE;
          else $v = f(v)$, $c = f(c)$
return $H = (V, E')$.

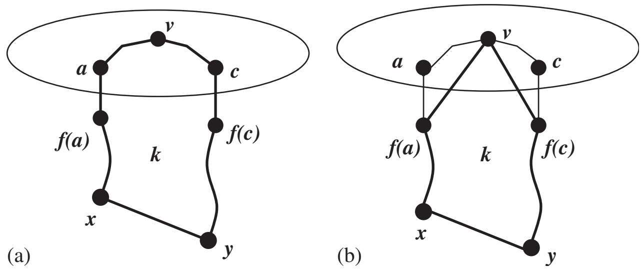

Fig. 8. The distance between a and c is at most k in H.

Clearly, H contains all edges of BFS-tree T because for each $v \in V$ the edge $vf(v)$ is in H. For a vertex v of G, let P_v be the path of BFS-tree T connecting v with the root u. We call it *the maximum neighbor path of* v in G (evidently, P_v is a shortest path of G). Additionally to the edges of T, H contains also some bridging edges connecting vertices from different maximum neighbor paths.

Lemma 12. *Let c be vertex of Q_i^l with the minimum number in σ ($l \in \{1, \ldots, q\}$, $i \in \{1, \ldots, p_l\}$). Then, for any $a \in Q_i^l$, there is a (a, c)-path in H of length at most k consisting of a subpath $(a, \ldots, x)$ of path P_a, edge xy and a subpath $(y, \ldots, c)$ of path P_c. In particular,*

$$d_H(a, c) \leqslant k.$$

Moreover, $0 \leqslant d_G(c, y) - d_G(a, x) \leqslant 1$.

Proof. If a is adjacent to c in G, then by construction of H, $d_H(a, c) = 1$, and we are done. So, assume a is not adjacent to c. Then, since a and c are in the same connected component Q_i^l, there must exist a path of G outside the ball $B_{l-1}(u)$ which connects a and c. Choose an induced subpath P of that path. Consider in G also maximum neighbor paths P_a and P_c, and let x be the vertex of P_a closest to a which has a neighbor in P_c (neighbors are considered in graph G). Denote a neighbor of x in P_c which is closest to c by y. Clearly, path P^* of G formed by (a, x)-subpath of P_a, edge xy and (c, y)-subpath of P_c is induced. Furthermore, only vertices $f(a)$ and $f(c)$ of P^* can have neighbors in P.

Now, if the length of P^* is larger than k, then we can find an induced cycle in G of length at least $k + 1$ by joining paths P and P^* (even if vertices $f(a)$ and $f(c)$ of P^* have a common neighbor in P we still will get an induced cycle of length at least $k + 1$).

So, the length of P^* cannot exceed k (Fig. 8 illustrates this) and we claim that P^* is a path in H, too. Indeed, both maximum neighbor paths P_a and P_c are in H (since they are from BFS-tree T). If we assume that xy is not in E', then by *while* loop of Procedure 3 and the choice of x and y, we must have $x \in N_j(u)$ and $y \in N_{j+1}(u)$ for some $j < l$. Moreover, $x \neq f(y)$ (otherwise, $xy = f(y)y$ is in E'). Let $P_a = (a = a_1, a_2, \ldots, a_{s-1}, a_s = x, \ldots, u)$ and $P_c = (c = c_1, c_2, \ldots, c_{s-1} = y, c_s = f(y), \ldots, u)$. Since $a > c$ and $a_h \neq c_h$ for any $h \in \{1, \ldots, s\}$, by property (P3), we have $a_h > c_h$ ($h \in \{1, \ldots, s\}$). Now, $x \in N(y)$ and $x > f(y)$ contradict with property (P2). Consequently, edge xy must be in E'. From this we also deduce that if $x \in N_j(u)$ for some $j < l$, then y is either in $N_j(u)$ or in $N_{j-1}(u)$.

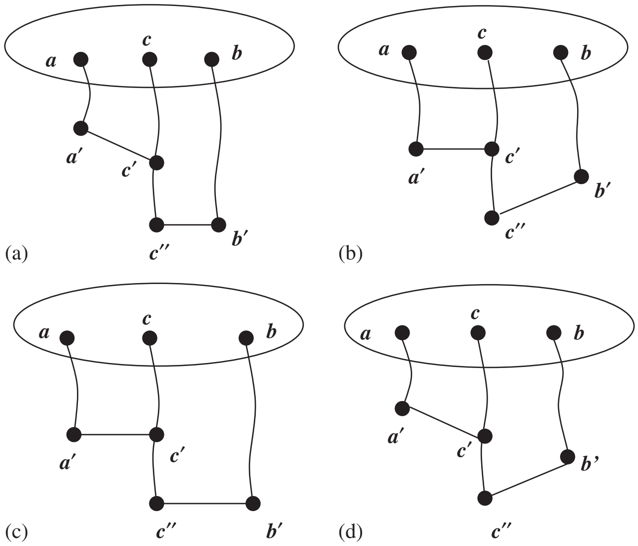

Fig. 9. Four possibilities (up to symmetry) of connection between a and b: (a) a', c' are in neighbor layers, b', c'' are in the same layer; (b) a', c' are in the same layer, b', c'' are in neighbor layers; (c) a', c' are in the same layer, b', c'' are in the same layer; (d) a', c' are in neighbor layers, b', c'' are in neighbor layers.

Thus, P^* is a path of H and therefore, $d_H(a, c) \leqslant k$. □

For any n-vertex k-chordal graph $G = (V, E)$ the following lemma holds.

Lemma 13. *H is an additive $(k + 1)$-spanner of G.*

Proof. Consider vertices $x \in Q_i^l$ and $y \in Q_j^m$, and assume that the lowest common ancestor of Q_i^l and Q_j^m in Γ is Q_s^p. Since Γ is a tree, any path connecting x and y passes Q_s^p. So, we have

$$d_G(x, y) \geqslant l - p + m - p.$$

Since H contains all edges of BFS-tree T, there must exist vertices $a, b \in Q_s^p$ such that $d_H(x, a) = l - p$ and $d_H(y, b) = m - p$. Now, we only need to show that

$$d_H(a, b) \leqslant k + 1.$$

Consider the maximum neighbor paths P_a, P_c and P_b in G. According to Lemma 12, there exists (a, c)-path and (b, c)-path in H both of length at most k such that

- (a, c)-path is formed by a subpath $(a, \ldots, a')$ of path P_a, edge $a'c'$ and a subpath $(c', \ldots, c)$ of path P_c, and
- (b, c)-path is formed by a subpath $(b, \ldots, b')$ of path P_b, edge $b'c''$ and a subpath $(c'', \ldots, c)$ of path P_c.

Moreover, $0 \leqslant d_G(c, c') - d_G(a, a') \leqslant 1$ and $0 \leqslant d_G(c, c'') - d_G(b, b') \leqslant 1$. Hence, up to symmetry, we have only four possible connections in H between vertices a and b that use those (a, c)- and (b, c)-paths (consult with Fig. 9).

In all cases we have

$$\begin{aligned} d_H(b, a) &\leqslant d_H(b, b') + 1 + d_H(c'', c') + 1 + d_H(a', a) \\ &\leqslant d_H(b, b') + 1 + d_H(c'', c') + 1 + d_H(c', c) \\ &= d_H(b, b') + 1 + d_H(c'', c) + 1 \\ &\leqslant k + 1. \end{aligned}$$

Thus, H is an additive $(k + 1)$-spanner of G. □

Lemma 14. *If G has n vertices, then H has at most* $2n - 2$ *edges.*

Proof. For each vertex $v \in V$ distinct from u we add to E' one or two edges. Since for u no such edges have been added to E', the total number of edges in H is at most $2n - 2$. □

Lemma 15. *If G is a k-chordal graph with n vertices and m edges, then H can be constructed in* $O(n\,k + m)$ *time.*

Proof. First, we show that the connected components of the graphs Q^l $(l = 0, 1, \ldots)$ can be found in total linear time. A BFS-tree of G can be constructed in linear time. Having a BFS-tree, we start from the sphere $N_q(u)$ of largest radius, find its connected components and contract each of them into a vertex. Then find the connected components in the graph induced by $N_{q-1}(u)$ and the set of contracted vertices, contract each of them and descend to the lower level, until we reach the vertex u. So, we can assume that all the connected components $Q^l_1, Q^l_2, \ldots, Q^l_{p_l}$ for any l, $0 \leqslant l \leqslant q$ are given.

Now, to get overall $O(n\,k + m)$ time bound, we need only to note that, by Lemma 12, for each vertex v, the *while* loop of Procedure 3 will work at most $\lfloor k/2 \rfloor$ time. □

Summarizing, we have the following result for k-chordal graphs.

Theorem 4. *Every k-chordal graph* $G = (V, E)$ *with n vertices and m edges admits an additive* $(k + 1)$*-spanner with at most* $2n - 2$ *edges. Moreover, such a sparse spanner of G can be constructed in* $O(n\,k + m)$ *time.*

5. Subclasses of 4-chordal graphs

There is an interesting bipartite analog of chordal graphs, so-called *chordal bipartite graphs*. These are bipartite graphs whose largest induced cycles are of length 4. In other words, they are exactly bipartite 4-chordal graphs. By Theorem 4, every such graph with n vertices admits an additive 5-spanner with at most $2n - 2$ edges. It is well-known that this family of graphs has also very close relations to so-called *strongly chordal graphs* (a subclass of chordal graphs). Those relations can be expressed (see [5,7,12,13,20,31]) in terms of

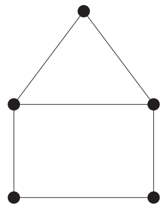

Fig. 10. Forbidden induced subgraph.

Γ-free orderings, totally balancedness of the open/closed neighborhood hypergraphs and in many other ways. As it was shown in [4], unlike general chordal graphs, every strongly chordal graph admits an additive tree 3-spanner. Hence, it is very natural to ask whether chordal bipartite graphs admit additive tree r-spanners for some small integer r. In this section first we note that for every fixed integer l there is a chordal bipartite graph which does not have additive tree $(2l-1)$-spanners and multiplicative tree $2l$-spanners. Then, we show that a slight modification of the Procedure 3 can produce an additive 4-spanner with at most $2n-2$ edges for any chordal bipartite graph. In fact, we will prove a more general result; any 4-chordal graph which does not contain a house as an induced subgraph (see Fig. 10) admits an additive 4-spanner with at most $2n-2$ edges. 4-Chordal graphs without induced houses are known also as *House-Hole-free graphs* or *HH-free graphs* for short.

Our construction of bad chordal bipartite graphs is similar to the one presented in [29] for chordal graphs. It was proved in [29] that for every fixed integer l there is a chordal graph which does not have additive tree l-spanners as well as multiplicative tree $(l+1)$-spanners.

Let G_1 be the induced 4-cycle C_4 (the square), and let G_2 be the graph obtained from G_1 by adding for each edge of G_1 a C_4 which shares that edge with G_1. For any integer $l > 2$ the graph G_l is obtained by adding for every edge in $E(G_{l-1})\backslash E(G_{l-2})$ one new C_4 which shares this edge with G_{l-1} (Fig. 11 shows graphs G_1 and G_3). These graphs G_l ($l > 0$) all are outerplanar and chordal bipartite.

Lemma 16. *No additive tree* $(2l-1)$*-spanner and no multiplicative tree* $2l$*-spanner is possible for* G_l.

Proof. Given a natural plane embedding of the outerplanar graph G_l, let G_l^* be its geometric dual. Namely, in each face of G_l, including the outer face F_0, we pick a point. These points will form the vertex set of G_l^*. Now, for each edge e of G_l we draw a dual edge in G_l^* which crosses e and connects the vertices of G_l^* that correspond to two faces sharing e in G_l. Note that G_l^* is a multigraph (has parallel edges); its vertex F_0 has three edges to each neighbor.

Let T be a spanning tree of G_l. The dual tree T^* contains all the edges in G_l^* which cross the edges of G_l that do not belong to the spanning tree T. See Fig. 11 for an illustration.

Let c be the vertex of T^* corresponding to the central square of G_l. Since T is a tree (a spanning tree of G_l), there must exist a curve on plane which connects a point of square c with a point of face F_0 without crossing any edge of T. Therefore, there must exist a

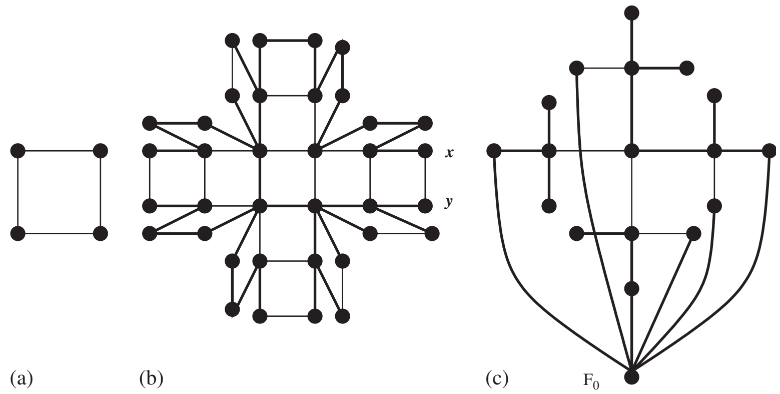

Fig. 11. (a) Chordal bipartite graph G_1. (b) Chordal bipartite graph G_3 with a spanning tree T. (c) The dual graph of G_3 (for simplicity we did not show all edges incident to F_0) and the dual tree of G_3 with respect to T.

path P^* in T^* between c and F_0 such that $P^* \setminus \{F_0\}$ belongs to one connected component of $T^* \setminus \{F_0\}$. It is easy to show by induction on l that the length of P^* is at least l. Let B be a largest connected component of the forest $T^* \setminus \{F_0\}$. Then, B contains at least l vertices.

Let F_1 be the neighbor of F_0 in B and let $xy \in E(G_l)$ be the edge crossed by F_0F_1. Since F_0F_1 is an edge in T^*, x and y are not adjacent in T. Consider an arbitrary edge uv of the path $P^T_{x,y}$ of T connecting vertices x and y, and let F_{uv}, F_{vu} be the two faces of G_l sharing the edge uv. We claim that exactly one of the vertices F_{uv}, F_{vu} is in B. If both F_{uv} and F_{vu} are in B then, since $\{u, v\}$ is a 2-cut of G_l and B is connected, we conclude $F_{uv}F_{vu} \in E(T^*)$ and therefore $uv \notin E(T)$, which is a contradiction. Now assume that neither F_{uv} nor F_{vu} is in B. We know that $P^T_{x,y}$ decomposes the outerplanar graph G_l into two parts and one of them, say G'_l, contains all the faces corresponding to the vertices of B. So, G'_l is a subgraph of G_l bounded by path $P^T_{x,y}$ and edge xy. Since either F_{uv} or F_{vu} belongs to G'_l and B is connected, there must exist two vertices $F' \in B$ and $F'' \notin B$ corresponding to faces of G'_l, such that $F'F''$ is an edge of G^*_l but not an edge of T^*. Let $x'y'$ be the edge of G'_l shared by faces F' and F''. $F'F'' \notin E(T^*)$ implies $x'y' \in E(T)$. On the other hand, since G'_l is outerplanar (as a subgraph of an outerplanar graph G_l), vertices x', y' must belong to the path $P^T_{x,y}$. Thus, two non-consecutive vertices x', y' of $P^T_{x,y}$ are adjacent in T, contradicting with T being a tree.

From the proof above we conclude that $d_T(x, y) + 1$ equals the number of edges in G^*_l that start in B and end outside B. Each vertex $F \in V(B)$ contributes to this number $\deg_{G^*_l}(F) - \deg_B(F)$ units, which is $4 - \deg_B(F)$ since all vertices of G^*_l except F_0 have degree 4. Thus,

$$d_T(x, y) + 1 = \sum_{F \in V(B)} (4 - \deg_B(F)).$$

Since B is a tree and $\sum_{F \in V(B)} \deg_B(F) = 2|E(B)|$, this equals

$$4|V(B)| - 2|E(B)| = 2|V(B)| + 2 \geqslant 2l + 2.$$

Therefore, we have $d_T(x, y) \geqslant 2l + 1$, which means T cannot be an additive $(2l - 1)$-spanner or a multiplicative $2l$-spanner of G_l. □

Let now $G = (V, E)$ be an HH-free graph, σ be a BFS-ordering of G started at a vertex u and T be the BFS-tree of G associated with σ. Assume also that the sets $Q_1^l, Q_2^l, \ldots, Q_{p_l}^l$ for each l, $l > 0$, (see Section 4 for definitions) are already computed. The following algorithm produces an additive 4-spanner for every HH-free graph G.

Procedure 4. Additive 4-spanners for *HH*-free graphs
Input: An *HH*-free graph $G = (V, E)$ with a BFS-ordering σ, and connected components $Q_1^l, Q_2^l, \ldots, Q_{p_l}^l$ for any l, $0 \leqslant l \leqslant q$, where $q = \max\{d_G(u, v) : v \in V\}$.
Output: A spanner $H = (V, E')$ of G.
Method:

```
E' = ∅;
for l = q downto 1 do
   for j = 1 to p_l do
      for each vertex v ∈ Q_j^l add vf(v) to E';
      pick vertex c in Q_j^l with the minimum number in σ;
      for each v ∈ Q_j^l \ {c} do
         connected = FALSE;
         while connected = FALSE do
         /* this while loop works at most twice for each v */
            if vf(c) ∈ E(G) then              /* here this method differs from the */
               add vf(c) to E';               /* one of Section 4; we first check */
               connected = TRUE;              /* adjacency between v and f(c) */
            else if vc ∈ E(G) then            /* and then between v and c. */
                  add vc to E';
                  connected = TRUE;
               else v = f(v), c = f(c)
return H = (V, E').
```

Since, clearly, H has at most $2n - 2$ edges and can be constructed in linear time, we need to prove only that H is an additive 4-spanner of G. The following auxiliary lemma for HH-free graph G will be needed in that proof.

Lemma 17. *For any $l > 0$ and any two adjacent vertices a, c from $N_l(u)$ such that $a > c$ in σ, a must be adjacent to $f(c)$.*

Proof. Let l be the smallest integer such that $af(c) \notin E(G)$ for $a, c \in N_l(u)$, $ac \in E(G)$, $a > c$. Since $a > c$ and $af(c) \notin E(G)$, by properties of BFS-orderings, we have

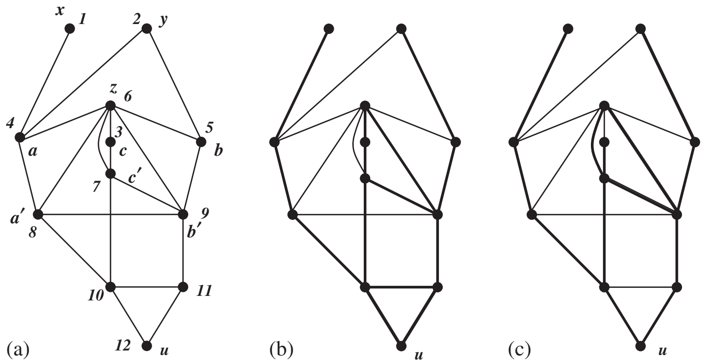

Fig. 12. (a) A weakly chordal graph with a LexBFS-ordering, (b) An additive 5-spanner generated by Procedure 3. (c) An additive 5-spanner generated by Procedure 4.

$f(a) > f(c)$ and $f(a)c \notin E(G)$. To avoid a forbidden induced cycle of length at least 5 in G formed by $a, c, f(a), f(c)$ and some vertices from $B_{l-2}(u)$, vertices $f(a)$ and $f(c)$ must be adjacent. By minimality of l, the father of $f(c)$ has to be adjacent to $f(a)$. But then, vertices $a, c, f(a), f(c), f(f(c))$ induce a house, which is impossible. □

Lemma 18. *H is an additive* 4*-spanner for G.*

Proof. Similar to the proof of Lemma 13, we only need to show that $d_H(a,b) \leqslant 4$ holds for any two vertices a, b from Q_s^p. Let c be the vertex of Q_s^p with the minimum number in σ. By the proof of Lemma 12, at least one of the following edges $ac, af(c)$, $f(a)f(c)$, $f(a)f(f(c))$ must exist in G. But by Lemma 17, if $ac \in E(G)$ then $af(c) \in E(G)$ and if $f(a)f(c) \in E(G)$ then $f(a)f(f(c)) \in E(G)$. Therefore, by Procedure 4, either edge $af(c)$ or edge $f(a)f(f(c))$ will go to $E(H)$. Symmetrically, either edge $bf(c)$ or edge $f(b)f(f(c))$ will go to $E(H)$. In any of four possible cases we have $d_H(a,b) \leqslant 4$. Recall that all edges of BFS-tree T are in H. □

Our final result is the following.

Theorem 5. *Every HH-free graph $G = (V, E)$ with n vertices and m edges admits an additive* 4*-spanner with at most* $2n - 2$ *edges. Moreover, such a sparse spanner of G can be constructed in* $O(n + m)$ *time.*

It is interesting to note that both procedures (3 and 4) may produce additive 5-spanners for the well-known class of weakly chordal graphs (see Fig. 12). Recall that G is a *weakly chordal graph* if both G and its complement $\overline{G}$ are 4-chordal. This class is a superclass of HH-free graphs and a subclass of 4-chordal graphs. A question whether a weakly chordal graph admits an additive 4-spanner with $O(n)$ edges remains open.

Acknowledgements

The authors are indebted to the referees for many comments improving the presentation.

References

[1] I. Althöfer, G. Das, D. Dobkin, D. Joseph, J. Soares, On sparse spanners of weighted graphs, Discrete Comput. Geom. 9 (1993) 81–100.
[2] H.-J. Bandelt, A. Dress, Reconstructing the shape of a tree from observed dissimilarity data, Adv. Appl. Math. 7 (1986) 309–343.
[3] U. Brandes, D. Handke, NP-completeness results for minimum planar spanners, University of Konstanz, Konstanzer Schriften in Mathematik und Informatik, Nr. 16, Preprint, Oktober 1996.
[4] A. Brandstädt, V. Chepoi, F. Dragan, Distance approximating trees for chordal and dually chordal graphs, J. Algorithms 30 (1999) 166–184.
[5] A. Brandstädt, F. Dragan, V. Chepoi, V. Voloshin, Dually chordal graphs, SIAM J. Discrete Math. 11 (1998) 437–455.
[6] A. Brandstädt, F.F. Dragan, H.-O. Le, V.B. Le, Tree spanners on chordal graphs: complexity, algorithms, open problems, Proc. 13th Internat. Symp. on Algorithms and Computation (ISAAC'02), November 2002, Lecture Notes in Computer Science, Vol. 2518, Springer, Berlin, pp. 163–174.
[7] A.E. Brouwer, P. Duchet, A. Schrijver, Graphs whose neighbourhoods have no special cycles, Discrete Math. 47 (1983) 177–182.
[8] L. Cai, Tree spanners: spanning trees that approximate the distances, Ph.D. Thesis, University of Toronto, 1992.
[9] L. Cai, D.G. Corneil, Tree spanners, SIAM J. Discrete Math. 8 (1995) 359–387.
[10] V. Chepoi, F. Dragan, A note on distance approximating trees in graphs, European J. Combin. 21 (2000) 761–766.
[11] F.F. Dragan, Estimating all pairs shortest paths in restricted graph families: a unified approach (extended abstract), Proc. 27th Internat. Workshop "Graph-Theoretic Concepts in Computer Science" (WG'01), June 2001, Lecture Notes in Computer Science, Vol. 2204, Springer, Berlin, pp. 103–116.
[12] F.F. Dragan, V. Voloshin, Incidence graphs of biacyclic hypergraphs, Discrete Appl. Math. 68 (1996) 259–266.
[13] M. Farber, Characterizations of strongly chordal graphs, Discrete Math. 43 (1983) 173–189.
[14] S.P. Fekete, J. Kremer, Tree spanners in planar graphs, Discrete Appl. Math. 108 (2001) 85–103.
[15] J.R. Gilbert, D.J. Rose, A. Edenbrandt, A separator theorem for chordal graphs, SIAM J. Algebraic Discrete Methods 5 (1984) 306–313.
[16] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.
[17] Hoàng-Oanh Le, Effiziente Algorithmen für Baumspanner in chordalen Graphen, Diploma Thesis, Department of Mathematics, Technical University of Berlin, 1994.
[18] H.-O. Le, V.B. Le, Optimal tree 3-spanners in directed path graphs, Networks 34 (1999) 81–87.
[19] A.L. Liestman, T. Shermer, Additive graph spanners, Networks 23 (1993) 343–364.
[20] A. Lubiw, Doubly lexical orderings of matrices, SIAM J. Comput. 16 (1987) 854–879.
[21] M.S. Madanlal, G. Venkatesan, C. Pandu Rangan, Tree 3-spanners on interval, permutation and regular bipartite graphs, Inform. Process. Lett. 59 (1996) 97–102.
[22] I.E. Papoutsakis, Two structure theorems on tree spanners, M.Sc. Thesis, Department of Computer Science, University of Toronto, 1999.
[23] I.E. Papoutsakis, On the union of two tree spanners of a graph, Manuscript, 2001.
[24] D. Peleg, Distributed Computing: A Locality-Sensitive Approach, SIAM Monographs on Discrete Mathematical Applications, SIAM, Philadelphia, 2000.
[25] D. Peleg, A.A. Schäffer, Graph spanners, J. Graph Theory 13 (1989) 99–116.
[26] D. Peleg, J.D. Ullman, An optimal synchronizer for the hypercube, in: Proc. Sixth ACM Symp. on Principles of Distributed Computing, Vancouver, 1987, pp. 77–85.
[27] D. Peleg, E. Upfal, A tradeoff between space and efficiency for routing tables, in: 20th ACM Symp. on the Theory of Computing, Chicago, 1988, pp. 43–52.

[28] E. Prisner, Distance approximating spanning trees, in: R. Reischuk, M. Morvan (Eds.), Proc. of STACS'97, Lecture Notes in Computer Science, Vol. 1200, Springer, Berlin, New York, 1997, pp. 499–510.

[29] E. Prisner, D. Kratsch, H.-O. Le, H. Müller, D. Wagner, Additive tree spanners, SIAM J. Discrete Math. 17 (2003) 332–340.

[30] J. Soares, Graph spanners: a survey, Congr. Numer. 89 (1992) 225–238.

[31] J.P. Spinrad, Doubly lexical ordering of dense 0–1-matrices, Inform. Process. Lett. 45 (1993) 229–235.

[32] G. Venkatesan, U. Rotics, M.S. Madanlal, J.A. Makowsky, C. Pandu Ragan, Restrictions of minimum spanner problems, Inform. Comput. 136 (1997) 143–164.