

# Collective Tree 1-Spanners for Interval Graphs

Derek G. Corneil<sup>1</sup>, Feodor F. Dragan<sup>2</sup>, Ekkehard Köhler<sup>3</sup>, and Chenyu Yan<sup>2</sup>

<sup>1</sup> Department of Computer Science, University of Toronto, Toronto, Ontario, Canada  
dgc@cs.toronto.edu

<sup>2</sup> Department of Computer Science, Kent State University, Kent, Ohio, U.S.A  
{dragan, cyan}@cs.kent.edu

<sup>3</sup> Institut für Mathematik, Technische Universität Berlin, Berlin, Germany  
ekoehler@math.TU-Berlin.DE

**Abstract.** In this paper we study the existence of a small set  $\mathcal{T}$  of spanning trees that collectively “1-span” an interval graph  $G$ . In particular, for any pair of vertices  $u, v$  we require a tree  $T \in \mathcal{T}$  such that the distance between  $u$  and  $v$  in  $T$  is at most one more than their distance in  $G$ . We show that:

- there is no constant size set of collective tree 1-spanners for interval graphs (even unit interval graphs),
- interval graph  $G$  has a set of collective tree 1-spanners of size  $O(\log D)$ , where  $D$  is the diameter of  $G$ ,
- interval graphs have a 1-spanner with fewer than  $2n - 2$  edges.

Furthermore, at the end of the paper we state other results on collective tree  $c$ -spanners for  $c > 1$  and other more general graph classes.

## 1 Introduction

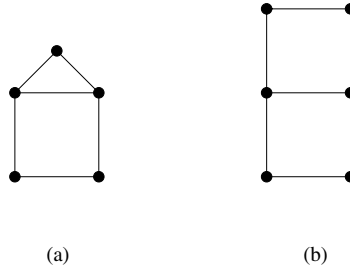
A spanning subgraph  $H$  of  $G$  is called a *spanner* of  $G$  if  $H$  provides a “good” approximation of the distances in  $G$ . More formally, for  $c \geq 1$ ,  $H$  is called an *additive  $c$ -spanner* of  $G$  if for any pair of vertices  $u$  and  $v$  their distance in  $H$  is at most  $c$  plus their distance in  $G$  [10]. (A similar definition can be given for multiplicative  $c$ -spanners [1,14,13]; however since we are only concerned with additive spanners, we will often omit “additive”.) In this paper, we continue the approach taken in [5,4,7] of studying *collective tree spanners*. We say that a graph  $G(V, E)$  *admits a system of  $\mu$  collective additive tree  $c$ -spanners* if there is a system  $\mathcal{T}(G)$  of at most  $\mu$  spanning trees of  $G$  such that for any two vertices  $u, v$  of  $G$  a spanning tree  $T \in \mathcal{T}(G)$  exists such that the distance in  $T$  between  $u$  and  $v$  is at most  $c$  plus their distance in  $G$ . We say that system  $\mathcal{T}(G)$  *collectively  $c$ -spans* the graph  $G$ . Clearly, if  $G$  admits a system of  $\mu$  collective additive tree  $c$ -spanners, then  $G$  admits an additive  $c$ -spanner with at most  $\mu \times (n - 1)$  edges (take the union of all those trees), and if  $\mu = 1$  then  $G$  admits an additive tree  $c$ -spanner. Note also that any graph on  $n$  vertices admits a system of at most  $n - 1$  collective additive tree 0-spanners (take  $n - 1$  Breadth-First-Search-trees (also known as *shortest path trees*) rooted at different vertices of  $G$ ).

One of the motivations to introduce this new concept steams from the problem of designing compact and efficient routing schemes in graphs. In [6,15], a shortest path routing labeling scheme for trees is described that assigns each vertex of an  $n$ -vertex tree a  $O(\log^2 n / \log \log n)$ -bit label. Given the label of a source vertex and the label of a destination, it is possible to compute in constant time, based solely on these two labels, the neighbor of the source that heads in the direction of the destination. Clearly, if an  $n$ -vertex graph  $G$  admits a system of  $\mu$  collective additive tree  $r$ -spanners, then  $G$  admits a routing labeling scheme of deviation (i.e., additive stretch)  $r$  with addresses and routing tables of size  $O(\mu \log^2 n / \log \log n)$  bits per vertex. Once computed by the sender in  $\mu$  time (by choosing for a given destination an appropriate tree from the collection to perform routing), headers of messages never change, and the routing decision is made in constant time per vertex (for details see [4,5]).

Previously, collective tree spanners of particular classes of graphs were considered in [4,5,7]. Paper [5] showed that any chordal graph, chordal bipartite graph or cocomparability graph admits a system of at most  $\log_2 n$  collective additive tree 2-spanners. These results were complemented by lower bounds, which say that any system of collective additive tree 1-spanners must have  $\Omega(\sqrt{n})$  spanning trees for some chordal graphs and  $\Omega(n)$  spanning trees for some chordal bipartite graphs and some cocomparability graphs. Furthermore, it was shown that any  $k$ -chordal graph admits a system of at most  $\log_2 n$  collective additive tree  $(2\lfloor k/2 \rfloor)$ -spanners and any circular-arc graph admits a system of two collective additive tree 2-spanners. Paper [4] showed that any AT-free graph (graph without asteroidal triples) admits a system of two collective additive tree 2-spanners, any graph having a dominating shortest path admits a system of two collective additive tree 3-spanners and a system of five collective additive tree 2-spanners, and any graph with asteroidal number  $\text{an}(G)$  admits a system of  $\text{an}(G)(\text{an}(G) - 1)/2$  collective additive tree 4-spanners and a system of  $\text{an}(G)(\text{an}(G) - 1)$  collective additive tree 3-spanners. Collective multiplicative tree spanners of planar graphs were investigated in [7]. It was shown that any weighted  $n$ -vertex planar graph admits a system of  $O(\sqrt{n})$  collective multiplicative tree 1-spanners (equivalently, additive tree 0-spanners) and a system of at most  $2 \log_{3/2} n$  collective multiplicative tree 3-spanners.

In this paper we study collective tree 1-spanners for interval graphs. In Section 2, we show that no constant number of trees can collectively 1-span interval graphs (even unit interval graphs). Surprisingly there is, as shown in Section 4, an additive 1-spanner that uses fewer than  $2n - 2$  edges, the number of edges required for two disjoint spanning trees. In Section 3, we present a polynomial time algorithm to find a set of  $O(\log D)$  trees that collectively 1-span a given interval graph  $G$ , where  $D$  is the diameter of  $G$ . In the final section we briefly list other results on families of graphs that strictly contain interval graphs. First we present the definitions used in this paper.

**Notation and Definitions:** All graphs occurring in this paper are connected, finite, undirected, loopless and without multiple edges. In a graph  $G(V, E)$  ( $n = |V|, m = |E|$ ) the *length* of a path from a vertex  $v$  to a vertex  $u$  is the number



**Fig. 1.** (a) A house, (b) A domino

of edges in the path. The *distance*  $d_G(u, v)$  between the vertices  $u$  and  $v$  is the length of a shortest path connecting  $u$  and  $v$ . The *eccentricity*  $ecc(v)$  of a vertex  $v$  of  $G$  is  $\max_{u \in V} d_G(u, v)$ . The diameter  $diam(G)$  of  $G$  is  $\max_{v \in V} ecc(v)$ . The  $i$ th *neighborhood* of a vertex  $v$  of  $G$  is the set  $N_i(v) = \{u \in V : d_G(v, u) = i\}$ . For a vertex  $v$  of  $G$ , the sets  $N(v) = N_1(v)$  and  $N[v] = N(v) \cup \{v\}$  are called the *open neighborhood* and the *closed neighborhood* of  $v$ , respectively. For a set  $S \subseteq V$ , by  $N[S] = \bigcup_{v \in S} N[v]$  we denote the *closed neighborhood* of  $S$  and by  $N(S) = N[S] \setminus S$  the *open neighborhood* of  $S$ . A set  $D \subseteq V$  is called a *dominating set* of a graph  $G = (V, E)$  if  $N[D] = V$ .

An independent set of three vertices such that each pair is joined by a path that avoids the neighborhood of the third is called an *asteroidal triple* (AT). A graph  $G$  is an *AT-free graph* if it does not contain any asteroidal triples [2]. A graph is *chordal* if it does not contain any induced cycles of length greater than 3. A graph is an *interval graph* if one can associate with each vertex an interval on the real line such that two vertices are adjacent if and only if the corresponding intervals have a nonempty intersection. Furthermore, an interval graph is a *unit interval graph* if all intervals are of the same length. Unit interval graphs are equivalent to *proper interval graphs* where no interval can properly contain any other interval. It is well known that a graph is an interval graph if and only if it is both a chordal graph and an AT-free graph [9].

A graph is *weakly chordal* (also called *weakly triangulated*) if neither  $G$  nor its complement  $\overline{G}$  contain an induced hole (cycle of size at least 5). A graph  $G$  is *house-hole-domino-free* (HHD-free) if it does not contain the house, the domino, and holes as induced subgraphs (see Fig. 1). Clearly, chordal graphs are strictly contained in both weakly chordal and HHD-free graphs.

## 2 Lower Bound

Independently McKee [12] and Kratsch et al. [8] showed that no single tree can  $c$ -span a chordal graph for any constant  $c$ . We now show a similar result for collectively 1-spanning a unit interval graph.

**Theorem 1.** *No constant number of trees can collectively 1-span a unit interval graph.*

*Proof.* First we will show that two trees do not suffice and then show how to extend this result to any constant number of trees.

The general “gadget” will be a  $K_3$  with two independent universal vertices  $x$  and  $y$  (i.e. we have a  $K_5$  with the edge  $xy$  missing). The vertices of the  $K_3$  will be labelled 1, 2, 3. Now make a sufficiently long chain of these gadgets by identifying the  $y$  vertex of a gadget with the  $x$  vertex of its right neighbor. It is straightforward to confirm that this graph  $G$  is a unit interval graph. Consider two trees  $T_1$  and  $T_2$  that supposedly collectively 1-span  $G$ . By making the chain sufficiently long, by the “pigeonhole principle”, we are guaranteed that there are three gadgets in  $G$  namely,  $A, B$  and  $C$  where  $A$  is left of  $B$  which is left of  $C$  such that:

- $T_1$  restricted to  $A, B$  and  $C$  is exactly the same spanning tree for all three gadgets. Exactly the same means from the labelled vertex point of view,
- $T_2$  restricted to  $A, B$  and  $C$  is also exactly the same spanning tree for all three gadgets. Note that  $T_1$  restricted to  $\{A, B, C\}$  is not necessarily the same as  $T_2$  restricted to  $\{A, B, C\}$ .

The vertices in  $A, B$  and  $C$  will be denoted  $A_x, B_3, C_y$ , where, for example,  $A_x$  refers to the  $x$ -vertex of  $A$ . We say that a tree provides a 1-approximating path between two vertices if the distance between the vertices in the tree is at most 1 more than their distance in  $G$ . We now show that in order for  $T_1$  or  $T_2$  to provide such an approximating path, certain edges of  $G$  must be present in the tree.

*Claim.* Let  $i$  be an element of  $\{1, 2, 3\}$ . If either  $T_1$  or  $T_2$  provides a 1-approximating path between  $A_i$  and  $C_i$ , then it must contain the  $xi$  and  $yi$  edges in all of  $A, B$  and  $C$ .

*Proof.* Without loss of generality, assume that  $T_1$  provides the 1-approximating path between  $A_i$  and  $C_i, i \in \{1, 2, 3\}$ . Such a path requires either  $A_i$  to be adjacent to  $A_y$  and/or  $C_i$  to be adjacent to  $C_x$ . Without loss of generality, assume  $C_i$  is adjacent to  $C_x$ ; thus since  $T_1$  when restricted to  $A, B$  and  $C$  is exactly the same,  $A_i$  is adjacent to  $A_x$  and  $B_i$  is adjacent to  $B_x$  as well. We now show that in all three of  $A, B$  and  $C, i$  is also adjacent to  $y$ . Suppose not; now in each gadget, the distance between  $i$  and  $y$  is at least 2 which means that the tree path between  $A_i$  and  $C_i$  must be at least 2 greater than the distance in  $G$  (since in  $T_1$  the distance between  $B_x$  and  $B_y$  must be at least 3 by following the edge  $B_x B_i$  and the path between  $B_i$  and  $B_y$ ).  $\square$

From the claim, it is clear that each of  $T_1$  and  $T_2$  can provide at most one path between  $A_1, C_1$  or  $A_2, C_2$  or  $A_3, C_3$  and thus at least three trees are required to 1-approximate  $G$ .

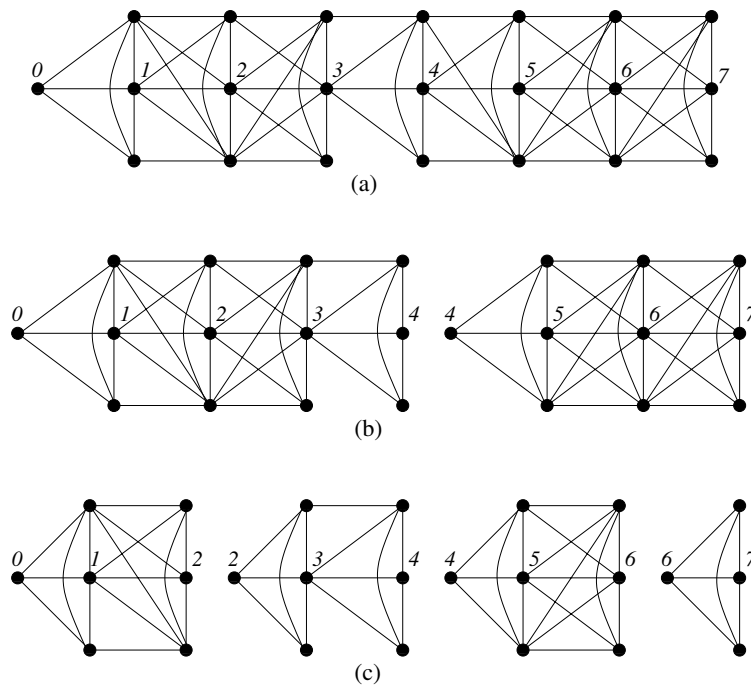
To generalize this argument, i.e. to show that at least  $k$  trees are required, merely replace the  $K_3$  in the gadget by a  $K_k$ . The same use of the claim shows that  $k - 1$  trees are not enough.  $\square$

A straightforward analysis (that will be presented in the journal version of the paper) shows that the size of the collective tree 1-spanners is  $\Omega(\sqrt{\log n})$ .

### 3 Upper Bound

In light of  $\Omega(\sqrt{\log n})$  spanning trees being needed to collectively 1-span an interval graph  $G$ , we now show that  $2 \log_2(D - 1) + 4$  spanning trees suffice, where  $D$  is the diameter of  $G$ .

Let  $P$  be a shortest path of a graph  $G$ . If every vertex  $z$  of  $G$  belongs to the neighborhood  $N[P]$  of  $P$ , then we say that  $P$  is a *dominating shortest path (DS-path)* of  $G$ . It is known that any AT-free graph has a DS-path which can be found in linear time [2]. In what follows we will need a slightly stronger result from [8].



**Fig. 2.** (a) Graph  $G = G_{0,7}$ , (b) graphs  $G_{0,4}$  and  $G_{4,7}$ , (c) graphs  $G_{0,2}$ ,  $G_{2,4}$ ,  $G_{4,6}$  and  $G_{6,7}$ . Graphs  $G_{0,1}$ ,  $G_{1,2}$ ,  $G_{2,3}$ ,  $G_{3,4}$ ,  $G_{4,5}$  and  $G_{5,6}$  are not shown.

**Lemma 1.** [2,8] Any AT-free graph  $G$  admits a DS-path  $(x_0, x_1, \dots, x_{ecc(x_0)})$  such that for every  $i = 1, 2, \dots, ecc(x_0)$ , every vertex  $z \in N_i(x_0)$  is adjacent to  $x_i$  or  $x_{i-1}$ . Moreover, such a DS-path can be constructed in linear time.

Now let  $G$  be an interval graph and let  $(x_0, x_1, \dots, x_{ecc(x_0)})$  be such a DS-path of  $G$  described by Lemma 1. The following lemma is important for our future discussion.

**Lemma 2.** For any two adjacent vertices  $u \in N_i(x_0)$  and  $v \in N_{i+1}(x_0)$ ,  $u, v \in N[x_i]$  or  $u, v \in N[x_{i+1}]$ . Moreover, if  $u \neq x_i$ , then  $ux_i \in E$ .

*Proof.* If  $u = x_i$  or  $v = x_{i+1}$ , then the lemma is trivially true. Hence, we may assume that  $u \neq x_i$  and  $v \neq x_{i+1}$ . If  $ux_i \notin E$  then, by Lemma 1,  $ux_{i-1} \in E$ . If now  $vx_i \in E$ , then  $u, v, x_i, x_{i-1}$  give an induced cycle of length 4 in  $G$ , which is impossible for an interval graph. If  $vx_i \notin E$  then, by Lemma 1,  $vx_{i+1} \in E$ . Then, we obtain either an induced cycle of length 5 or induced cycle of length 4, depending on whether or not  $ux_{i+1}$  is in  $E$ . So, if  $u \neq x_i$ , then  $ux_i$  must be in  $E$ . If now  $vx_{i+1} \in E$  but neither  $ux_{i+1}$  nor  $vx_i$  is in  $E$ , then  $x_i, u, v, x_{i+1}$  form an induced cycle of length 4 in  $G$ , which is impossible.  $\square$

Let  $l$  denote  $\text{ecc}(x_0)$ . For any two integers  $i, j$ ,  $0 \leq i < j \leq l$ , we define  $G_{i,j}$  to be the subgraph of  $G$  induced by vertices  $\{x_i\} \cup N_{i+1}(x_0) \cup \dots \cup N_j(x_0)$  (see Fig. 2 for an illustration). In view of Lemma 1, obviously,  $G_{i,j}$  is connected and  $G = G_{0,l}$ . We use the following procedure to construct a system of *local shortest path trees* of  $G$ .

**PROCEDURE 1. A system of local shortest path trees for an interval graph  $G$ .**

**Input:** An interval graph  $G$ , a DS-path  $(x_0, \dots, x_l)$  and the layering  $\{x_0\}, N_1(x_0), \dots, N_l(x_0)$  of  $G$ .

**Output:** A system of local shortest path trees of  $G$ .

**Method:**

```

set  $k := 0$ ;  $\mathcal{G}_k := \{G_{0,l}\}$ ;  $\mathcal{T} := \emptyset$ ;
while  $\mathcal{G}_k \neq \emptyset$  do
  set  $\mathcal{G}_{k+1} := \emptyset$ ;  $\mathcal{T}'_k := \emptyset$ ;  $\mathcal{T}''_k := \emptyset$ ;
  for each  $G_{i,j} \in \mathcal{G}_k$  do
    if  $j = i + 1$  then
      construct a shortest path tree of  $G_{i,j}$  rooted at  $x_i$  and put it in  $\mathcal{T}'_k$ ;
      construct a shortest path tree of  $G_{i,j}$  rooted at  $x_j$  and put it in  $\mathcal{T}''_k$ ;
    else /* if  $j > i + 1$  */
      set  $s := \lceil (j - i) / 2 \rceil + i + 1$ ;
      construct a shortest path tree of  $G_{i,j}$  rooted at  $x_{s-1}$  and put it in  $\mathcal{T}'_k$ ;
      construct a shortest path tree of  $G_{i,j}$  rooted at  $x_s$  and put it in  $\mathcal{T}''_k$ ;
      set  $\mathcal{G}_{k+1} := \mathcal{G}_{k+1} \cup \{G_{i,s-1}, G_{s-1,j}\}$ ;
  set  $\mathcal{T} := \mathcal{T} \cup \mathcal{T}'_k \cup \mathcal{T}''_k$ ;
  set  $k := k + 1$ ;
return  $\mathcal{T}$ .

```

Note that the while loop in the procedure above will be executed at most  $\log_2(l-1)+2$  times. Let  $G_{i,j}$  be an arbitrary subgraph generated by the procedure with  $j > i + 1$ . Let also  $s = \lceil (j - i) / 2 \rceil + i + 1$  and  $a \in N_r(x_0), b \in N_t(x_0)$  be two arbitrary vertices in  $G_{i,j}$ , where  $r \leq t$  are two integers between  $i$  and  $j$  inclusive. Let  $T_s, T_{s-1} \in \mathcal{T}$  be the two shortest path trees of  $G_{i,j}$  rooted at  $x_s, x_{s-1}$ , respectively. Clearly, both spanning trees span all the vertices of  $G_{i,j}$  and the subgraphs  $G_{i,s-1}$  and  $G_{s-1,j}$  of  $G_{i,j}$  have only one common vertex  $x_{s-1}$ . The following lemmata hold.

**Lemma 3.** *If  $r=t=s$ , then  $d_{T_s}(a,b) \leq d_G(a,b) + 1$  or  $d_{T_{s-1}}(a,b) \leq d_G(a,b) + 1$ .*

*Proof.* Since  $T_s$  and  $T_{s-1}$  are shortest path trees, using Lemma 1, one can easily show that  $d_{T_s}(a,b) \leq 3$  or  $d_{T_{s-1}}(a,b) \leq 3$ . So, if  $ab \notin E$  or  $a, b \in N[x_s]$  or  $a, b \in N[x_{s-1}]$ , then the lemma holds. If now  $ab \in E$  and, without loss of generality,  $ax_s, bx_{s-1} \in E$  and  $bx_s, ax_{s-1} \notin E$ , then the vertices  $a, b, x_{s-1}, x_s$  form an induced cycle of length 4 in  $G$ , which is impossible.  $\square$

In a similar way one can show the following.

**Lemma 4.** *If  $a$  and  $b$  are vertices of a graph  $G_{i,i+1}$  then  $d_{T'}(a,b) \leq d_G(a,b) + 1$  or  $d_{T''}(a,b) \leq d_G(a,b) + 1$ , where  $T', T'' \in \mathcal{T}$  are shortest path trees of  $G_{i,i+1}$  rooted at  $x_i$  and  $x_{i+1}$ , respectively.*

**Lemma 5.** *If  $i \leq r < s \leq t \leq j$ , then  $d_{T_s}(a,b) \leq d_{G_{i,j}}(a,b) + 1$  or  $d_{T_{s-1}}(a,b) \leq d_{G_{i,j}}(a,b) + 1$ .*

*Proof.* Using Lemma 1, it is easy to show that  $d_{T_s}(a,b) \leq t-r+3$  or  $d_{T_{s-1}}(a,b) \leq t-r+3$ . So, when  $d_{G_{i,j}}(a,b) \geq t-r+2$ , the lemma clearly holds. Therefore, we may assume that  $d_{G_{i,j}}(a,b)$  is  $t-r+1$  or  $t-r$ . Let first  $d_{G_{i,j}}(a,b) = t-r$  and  $(z_r = a, z_{r+1}, \dots, z_t = b)$  be a shortest path between  $a$  and  $b$  in  $G_{i,j}$ . Consider vertices  $z_{s-1}$  and  $z_s$ . According to Lemma 2, they both belong to  $N[x_s]$  or to  $N[x_{s-1}]$ . Without loss of generality, assume  $z_s, z_{s-1} \in N[x_s]$ . Since  $T_s$  is a shortest path tree,  $d_{T_s}(x_s, a) \leq s-r$  and  $d_{T_s}(x_s, b) \leq t-s+1$ . So,  $d_{T_s}(a,b) \leq d_{T_s}(x_s, a) + d_{T_s}(x_s, b) \leq t-r+1 = d_{G_{i,j}}(a,b) + 1$ .

Now assume that  $d_{G_{i,j}}(a,b) = t-r+1$ . Let  $z_s z_{s-1} \in E$  be an edge on the shortest path between  $a$  and  $b$  in  $G_{i,j}$  such that  $z_s \in N_s(x_0)$  and  $z_{s-1} \in N_{s-1}(x_0)$ . Obviously, such an edge must exist, and we have  $d_{G_{i,j}}(a,b) = d_{G_{i,j}}(a, z_{s-1}) + d_{G_{i,j}}(z_{s-1}, z_s) + 1$ . According to Lemma 2, both  $z_s$  and  $z_{s-1}$  belong to  $N[x_s]$  or to  $N[x_{s-1}]$ . Without loss of generality, assume they belong to  $N[x_s]$ . Then, since  $T_s$  is a shortest path tree of  $G_{i,j}$ ,  $d_{T_s}(x_s, a) \leq 1 + d_{G_{i,j}}(z_{s-1}, a)$  and  $d_{T_s}(x_s, b) \leq 1 + d_{G_{i,j}}(z_s, b)$ . Hence,  $d_{T_s}(a,b) \leq 2 + d_{G_{i,j}}(z_s, b) + d_{G_{i,j}}(z_{s-1}, a) = 1 + d_{G_{i,j}}(a,b)$ . This concludes our proof.  $\square$

**Lemma 6.** *If  $d_{G_{i,j}}(a,b) \neq d_G(a,b)$ , then  $a \in N_{i+1}(x_0)$  or  $b \in N_{i+1}(x_0)$ .*

*Proof.* Without loss of generality, assume that  $a \in N_r(x_0), b \in N_t(x_0)$  and  $i+1 \leq r \leq t \leq j$ . We claim that there always exists a shortest path  $P^G(a,b)$  between  $a$  and  $b$  in  $G$  such that  $P^G(a,b) \cap N_{j+1}(x_0) = \emptyset$ . If this is not the case, then there must exist vertices  $c, d \in P^G(a,b) \cap N_j(x_0)$  and  $c', d' \in N_{j+1}(x_0) \cap P^G(a,b)$  such that  $cc'$  and  $dd'$  are edges of  $P^G(a,b)$ . Obviously,  $cd \notin E$ . According to Lemma 2(second part),  $cx_j, dx_j \in E$ . Then, if we replace the part of  $P^G(a,b)$  between  $c$  and  $d$  with the path  $(c, x_j, d)$ , obviously we will get a shortest path between  $a$  and  $b$  that does not intersect  $N_{j+1}(x_0)$ . So, we may assume that  $P^G(a,b) \cap N_{j+1}(x_0) = \emptyset$ .

If neither  $a \in N_{i+1}(x_0)$  nor  $b \in N_{i+1}(x_0)$ , then  $i+1 < r \leq t$ . Since  $d_{G_{i,j}}(a,b) \neq d_G(a,b)$ , we must be able to find four vertices  $e, f \in N_{i+2}(x_0) \cap P^G(a,b)$  and  $e', f' \in N_{i+1}(x_0)$  such that  $ee'$  and  $ff'$  are edges of  $P^G(a,b)$ . If

$e'f' \in E$  or  $e' = f'$ , then  $P^G(a, b)$  is in  $G_{i,j}$ , i.e.,  $d_{G_{i,j}}(a, b) = d_G(a, b)$ . Hence, one may assume that  $e'f' \notin E$  and  $e' \neq f'$ . Then, according to Lemma 2(second part),  $e'x_{i+1}, f'x_{i+1} \in E$  and we can choose another shortest path between  $a$  and  $b$  that does not intersect  $N_i(x_0)$  and get  $d_{G_{i,j}}(a, b) = d_G(a, b)$  again. Thus, if neither  $a \in N_{i+1}(x_0)$  nor  $b \in N_{i+1}(x_0)$ , then  $d_{G_{i,j}}(a, b) = d_G(a, b)$ .  $\square$

We are ready to prove the following main lemma of this section.

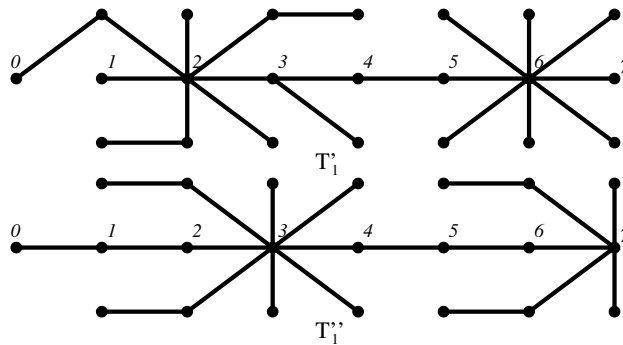
**Lemma 7.** *For any two vertices  $a, b \in V(G)$ , there exists a local shortest path tree  $T \in \mathcal{T}$  such that  $d_T(a, b) \leq d_G(a, b) + 1$ .*

*Proof.* Let  $G_{i,j}$  be a subgraph of  $G$ , generated by Procedure 1, which contains both vertices  $a$  and  $b$  and has the minimum difference  $j - i$ . If  $j - i = 1$  then we are done by Lemma 4. Therefore, in what follows we assume that  $j > i + 1$ , and let  $s = \lceil (j - i)/2 \rceil + i + 1$  and  $a \in N_r(x_0), b \in N_t(x_0)$ , where  $i \leq r \leq t \leq j$ . By minimality of  $j - i$ ,  $r < s \leq t$  (if  $t < s$  then  $G_{i,s-1}$  contains both  $a$  and  $b$ , and if  $r \geq s$  then  $G_{s-1,j}$  contains both  $a$  and  $b$ ).

The case  $i \leq r < s \leq t \leq j$  when  $d_{G_{i,j}}(a, b) = d_G(a, b)$  is handled by Lemma 5. Assume now that  $d_{G_{i,j}}(a, b) \neq d_G(a, b)$ . Let  $P^G(a, b)$  be an arbitrary shortest path between  $a$  and  $b$  in  $G$ . By Lemma 6,  $r = i + 1$ . We claim that  $d_G(a, b) = t - r + 2$ . Indeed, since  $d_{G_{i,j}}(a, b) \leq t - r + 3$  (recall that  $a \in N[x_{i+1}] \cup N(x_i)$  and  $b \in N[x_t] \cup N(x_{t-1})$  by Lemma 1) and  $d_{G_{i,j}}(a, b) \neq d_G(a, b)$ , we must have  $d_G(a, b) \leq t - r + 2$ . On the other hand, if  $d_G(a, b) \leq t - r + 1$ , then we can easily show that all the vertices of  $P^G(a, b)$  are in  $G_{i,j}$ , and thus  $d_{G_{i,j}}(a, b) = d_G(a, b)$ .

Consider now the local shortest path tree  $T_s \in \mathcal{T}$  of  $G_{i,j}$  rooted at  $x_s$ , where  $s = \lceil (j - i)/2 \rceil + i + 1$ . It is easy to show that  $d_{T_s}(x_s, a) \leq s - r + 2$  and  $d_{T_s}(x_s, b) \leq t - s + 1$ . Combining the two inequalities, we get  $d_{T_s}(a, b) \leq t - r + 3$ . Since  $d_G(a, b) = t - r + 2$ , the lemma holds.  $\square$

We can group the local shortest path trees from  $\mathcal{T}$  into at most  $2 \log_2(l-1)+4$  spanning trees of  $G$ . Consider Procedure 1. At the beginning,  $G_{0,l} = G$  and we



**Fig. 3.** Spanning trees  $T'_1$  and  $T''_1$  of an interval graph  $G$  from Fig. 2



construct only two spanning trees of  $G$ , i.e.,  $\mathcal{T}'_0 = \{T'_0\}$ ,  $\mathcal{T}''_0 = \{T''_0\}$ . In the second iteration,  $G$  is decomposed into two subgraphs  $G_{0,s-1}$  and  $G_{s-1,l}$  where  $s = \lceil l/2 \rceil + 1$ . For each of the two subgraphs, the algorithm constructs two local shortest path trees, i.e.,  $\mathcal{T}'_1 = \{T'^1_1, T'^2_1\}$ ,  $\mathcal{T}''_1 = \{T''^1_1, T''^2_1\}$ . Since  $G_{0,s-1}$  and  $G_{s-1,l}$  have only vertex  $x_{s-1}$  in common, we conclude  $T'_1 := T'^1_1 \cup T'^2_1$  and  $T''_1 := T''^1_1 \cup T''^2_1$  are two spanning trees of  $G$  (see Fig. 3). In general, during the iteration  $k$  of Procedure 1, for each of the  $2^{k-1}$  subgraphs  $G_{0,j_1}, G_{j_1,j_2}, \dots, G_{j_{2^{k-1}-1},j}$  of  $G$ , we construct two local shortest path trees, i.e.,  $\mathcal{T}'_k = \{T'^1_k, T'^2_k, \dots, T'^{2^{k-1}}_k\}$ ,  $\mathcal{T}''_k = \{T''^1_k, T''^2_k, \dots, T''^{2^{k-1}}_k\}$ , where  $T'^\gamma_k$  and  $T''^\gamma_k$  are the local shortest path trees constructed for  $G_{j_{\gamma-1},j_\gamma}$  ( $\gamma = 1, \dots, 2^{k-1}$ ). Again, for any  $\gamma = 1, \dots, 2^{k-1} - 1$ ,  $G_{j_{\gamma-1},j_\gamma}$  and  $G_{j_\gamma,j_{\gamma+1}}$  have only vertex  $x_{j_\gamma}$  in common. Therefore,  $T'_k := \bigcup_{1 \leq \gamma \leq 2^{k-1}} T'^\gamma_k$  and  $T''_k := \bigcup_{1 \leq \gamma \leq 2^{k-1}} T''^\gamma_k$  are two spanning trees of  $G$ . Since the number of iterations is bounded by  $\alpha \leq \log_2(l-1) + 2$ , in this way we will create a system  $\mathcal{ST} := \{T'_0, T''_0, T'_1, T''_1, \dots, T'_\alpha, T''_\alpha\}$  of at most  $2\alpha$  spanning trees of  $G$ . Furthermore, each local shortest path tree from  $\mathcal{T}$  will be contained in one of the spanning trees from  $\mathcal{ST}$  as a subtree. Thus, we proved the following result.

**Theorem 2.** *Any interval graph of diameter  $D$  admits a system of  $2 \log_2(D - 1) + 4$  collective additive tree 1-spanners. Moreover, these trees can be constructed in  $O(m \log D)$  total time.*

## 4 Sparse Spanner

Given the result in Theorem 1 that no constant number of trees can collectively 1-span a unit interval graph, it is somewhat surprising that there is a sparse 1-spanner of an interval graph that has fewer than  $2n - 2$  edges (i.e. the number of edges in two disjoint spanning trees). To see this, we first present an algorithm to produce a subgraph  $H$  of interval graph  $G$ . We then show that  $H$  has the required number of edges and is in fact a 1-spanner of  $G$ .

**PROCEDURE 2. Construction of a sparse 1-spanner for an interval graph  $G$ .**

**Input:** An interval graph  $G$ , and an interval ordering  $\prec$  of  $V$  where for all  $x \prec y \prec z$  if  $xz \in E$ , then  $xy \in E$ . Let  $D$  be the diameter of the graph  $G$ .

**Output:** A sparse 1-spanner  $H$  of  $G$ .

**Method:**

let  $x_D$  be the last vertex in the ordering  $\prec$ ; set  $E_H := \emptyset$ ;

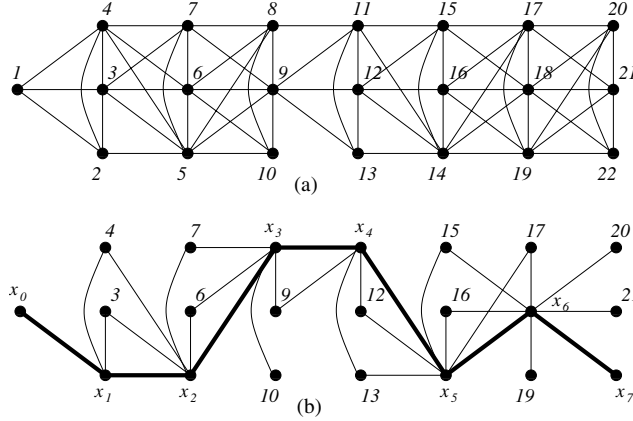
add the edge from  $x_D$  to its leftmost neighbor to  $E_H$ ;

**for**  $i$  **from**  $D$  **downto** 1 **do**

let  $x_{i-1}$  be the left most neighbor of  $x_i$ ;

add to  $E_H$  all edges from  $x_{i-1}$  to vertices to the right of  $x_{i-1}$  up to  $x_i$ ;

**if**  $i > 1$  **then** add to  $E_H$  all edges in  $G$  from  $x_{i-1}$  to vertices to the left of  $x_{i-1}$ .



**Fig. 4.** (a) Graph  $G$  and its interval ordering  $\prec$ . (b) Sparse 1-spanner  $H$  with the edges of  $P$  bold.

As an example of Procedure 2, consider Fig. 4(a) where the interval graph of Fig. 2 is repeated together with an interval numbering. The 1-spanner  $H$  is shown in Fig. 4(b) and the bold edges denote  $P$ , the path induced on  $\{x_i, 0 \leq i \leq D\}$ .

We now show that  $H$  is a sparse 1-spanner of  $G$ .

**Lemma 8.**  $H$  is a 1-spanner of  $G$  with at most  $2n - D - 2$  edges.

*Proof.* First we show that  $H$  has at most  $2n - D - 2$  edges. To see this note:

- all vertices to the right of  $x_{D-1}$  have degree 1 in  $H$  and there is at least one vertex here;
- all vertices to the left of  $x_{D-1}$  that are not on  $P$  have degree at most 2 in  $H$  (by the interval ordering property);
- there are  $D - 1$  edges joining the  $P \setminus \{x_D\}$  vertices.

Thus the total number of edges in  $H$  is at most  $1 + 2(n - (D + 1)) + D - 1 = 2n - D - 2$ , as required.

To see that  $H$  is a 1-spanner, consider arbitrary vertices  $x$  and  $y$  where  $x \prec y$  in the interval ordering. We now show that  $d_H(x, y) \leq d_G(x, y) + 1$ . This is clearly true if  $x$  is in  $P$ , so we assume that  $x$  is not in  $P$ . Now, suppose  $x$  is between  $x_i$  and  $x_{i+1}$  for  $i \geq 0$  and  $y$  satisfies  $x_j \prec y \preceq x_{j+1}$ , where  $i < j$ . (Note that if  $i = j$ , then immediately  $d_H(x, y) \leq 2$ .)

*Claim.*  $d_G(x, y) \geq j - i$ .

*Proof.* Suppose to the contrary that there is a path  $Q$  in  $G$  of length less than  $j - i$ . It is easy to see that the number of  $P$  vertices strictly between  $x$  and  $y$  is  $j - i$  and thus some edge  $uv$  (where  $u \prec v$ ) of  $Q$  surrounds two  $P$  vertices  $x_k$  and  $x_{k+1}$  (i.e.  $u \prec x_k \prec x_{k+1} \prec v$ ). Since  $uv \in E_G$ ,  $ux_{k+1} \in E_G$  contradicting  $x_k$  being the left most neighbor of  $x_{k+1}$ .  $\square$

Now suppose  $d_G(x, y) = j - i$  as witnessed by path  $Q := (q_0 = x, q_1, \dots, q_{j-i} = y)$ . Using the same argument as in the claim, for each  $k \in \{0, 1, \dots, j - 1\}$ ,  $q_k$  must lie between  $x_{i+k}$  and  $x_{i+k+1}$ . Since  $x = q_0$  is adjacent to  $q_1$ , and  $x_{i+1}$  is between  $q_0$  and  $q_1$ , we know that  $x$  is adjacent to  $x_{i+1}$ . Now consider the path in  $H$  from  $x$  to  $x_{i+1}, \dots, x_j, y$ . This path has length  $j - i + 1$ .

Thus we may assume that  $d_G(x, y) > j - i$ . But the path in  $H$  from  $x$  to  $x_i, \dots, x_j, y$  has length  $j - i + 2$  and we are finished.  $\square$

Thus we have the following result:

**Theorem 3.** *Any interval graph  $G$  of diameter  $D$  admits a sparse additive 1-spanner with at most  $2n - D - 2$  edges. Moreover, this spanner can be constructed in  $O(n + m)$  time.*

*Proof.* Given Lemma 4, we only have to establish the time complexity. There are many linear time interval graph recognition algorithms that can be used to determine an interval ordering of the given graph (for example see [3]). Using this ordering, a straightforward implementation of Procedure 2 can be achieved in linear time.  $\square$

Furthermore, in the journal version of the paper we will show that the sparse spanner returned by Procedure 2 can be used for efficient routing.

## 5 Concluding Remarks

The most obvious open question in this paper is to tighten the gap between the lower and upper bounds for the size of a collective tree 1-spanner for interval graphs.

The results stated in this paper also raise questions about additive  $c$ -spanners for  $c > 1$  for graph classes containing interval graphs. (Recall that interval graphs have a single tree that 2-spans the graph [11,8].) In the journal version of the paper, we will present proofs of the following theorems.

**Theorem 4.** *No constant number of trees can collectively additively  $c$ -span chordal graphs for  $c \leq 3$ .*

**Theorem 5.** *No constant number of trees can collectively additively  $c$ -span weakly chordal graphs for all constants  $c$ .*

**Theorem 6.** *Any HHD-free graph admits a system of at most  $2 \log_2 n$  collective additive tree 2-spanners. Moreover, such a set of trees can be constructed in  $O(m \log n)$  time.*

For the proof of Theorem 6 we show an auxiliary result of independent interest that any  $n$ -vertex HHD-free graph  $G$  has a separator  $S \subseteq V$  such that

- any connected component of  $G \setminus S$  has no more than  $n/2$  vertices and
- $S \subseteq (N[x] \cup N[y])$  for some vertices  $x, y \in S$ .

Moreover,  $S$  and such two vertices  $x$  and  $y$  can be found in linear time.

**Acknowledgements.** DGC wishes to thank the Natural Sciences and Engineering Research Council of Canada for financial assistance in the support of this research.

## References

1. L.P. CHEW, There are planar graphs almost as good as the complete graph, *J. of Computer and System Sciences*, 39 (1989), 205–219.
2. D.G. CORNEIL, S. OLARIU and L. STEWART, Asteroidal Triple-free Graphs, *SIAM J. Discrete Math.*, 10 (1997), 399–430.
3. D.G. CORNEIL, S. OLARIU and L. STEWART, The LBFS structure and recognition of interval graphs, under revision.
4. F.F. DRAGAN, C. YAN and D.G. CORNEIL, Collective Tree Spanners and Routing in AT-free Related Graphs (Extended Abstract), Proceedings of *30th International Workshop Graph-Theoretic Concepts in Computer Science (WG '04)*, June 2004, Bad Honnef, Germany, Springer, *Lecture Notes in Computer Science* 3353, 68-80.
5. F.F. DRAGAN, C. YAN and I. LOMONOSOV, Collective tree spanners of graphs, Proc. of the *9th Scandinavian Workshop on Algorithm Theory (SWAT'04)*, 8-10 July, 2004, Humlebæk, Denmark, Springer, *Lecture Notes in Computer Science* 3111, pp. 64-76.
6. P. FRAIGNIAUD and C. GAVOILLE, Routing in Trees, *Proceedings of the 28th Int. Colloquium on Automata, Languages and Programming (ICALP 2001)*, Lecture Notes in Computer Science 2076, 2001, pp. 757–772.
7. A. GUPTA, A. KUMAR and R. RASTOGI, Traveling with a Pez Dispenser (or, Routing Issues in MPLS), *SIAM J. Comput.*, 34 (2005), pp. 453-474.
8. D. KRATSCH, H.-O. LE, H. MÜLLER, E. PRISNER AND D. WAGNER Additive tree spanners *SIAM J. Discrete Math.* 17 (2003), 332-340.
9. C. LEKKERKERKER AND J. BOLAND Representation of a finite graph by a set of intervals on the real line *Fund. Math.*, 51 (1962), 45-64.
10. A.L. LIESTMAN AND T. SHERMER, Additive graph spanners, *Networks*, 23 (1993), 343-364.
11. M.S. MADANLAL, G. VENKATESAN, and C. PANDU RANGAN, Tree 3-spanners on interval, permutation and regular bipartite graphs, *Inform. Process. Lett.*, 59 (1996), 97-102.
12. T.A. MCKEE, personal communication to E. Prisner, 1995.
13. D. PELEG, and A.A. SCHÄFFER, Graph Spanners, *J. Graph Theory*, 13 (1989), 99-116.
14. D. PELEG AND J.D. ULLMAN, An optimal synchronizer for the hypercube, in *Proc. 6th ACM Symposium on Principles of Distributed Computing*, Vancouver, 1987, 77-85.
15. M. THORUP and U. ZWICK, Compact routing schemes, *Proceedings of the 13th Ann. ACM Symp. on Par. Alg. and Arch. (SPAA 2001)*, ACM 2001, pp. 1–10.