Additive Spanners for Circle Graphs and Polygonal Graphs[⋆]

Feodor F. Dragan[1], Derek G. Corneil[2], Ekkehard Köhler[3], and Yang Xiang[1]

[1] Algorithmic Research Laboratory, Department of Computer Science,
Kent State University, Kent, OH 44242, USA
dragan@cs.kent.edu, yxiang@cs.kent.edu

[2] Department of Computer Science, University of Toronto, Toronto, Ontario, Canada
dgc@cs.toronto.edu

[3] Mathematisches Institut, Brandenburgische Technische Universität Cottbus,
D-03013 Cottbus, Germany
ekoehler@math.TU-Cottbus.DE

Abstract. A graph $G = (V, E)$ is said to admit a system of μ collective additive tree r-spanners if there is a system $\mathcal{T}(G)$ of at most μ 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 r plus their distance in G. In this paper, we examine the problem of finding "small" systems of collective additive tree r-spanners for small values of r on circle graphs and on polygonal graphs. Among other results, we show that every n-vertex circle graph admits a system of at most $2\log_{\frac{3}{2}} n$ collective additive tree 2-spanners and every n-vertex k-polygonal graph admits a system of at most $2\log_{\frac{3}{2}} k+7$ collective additive tree 2-spanners. Moreover, we show that every n-vertex k-polygonal graph admits an additive $(k + 6)$-spanner with at most $6n - 6$ edges and every n-vertex 3-polygonal graph admits a system of at most 3 collective additive tree 2-spanners and an additive tree 6-spanner. All our collective tree spanners as well as all sparse spanners are constructible in polynomial time.

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 $r \geq 0$, H is called an *additive r-spanner* of G if for any pair of vertices u and v their distance in H is at most r plus their distance in G [19]. If H is a tree then it is called an *additive tree r-spanner* of G [24]. (A similar definition can be given for multiplicative t-spanners [9, 22, 23] and for multiplicative tree t-spanners [6].) In this paper, we continue the approach taken in [10, 12, 13, 14, 18] of studying *collective tree spanners*. We say that a graph $G = (V, E)$ *admits a system of μ collective additive tree r-spanners* if there is a system $\mathcal{T}(G)$ of at most μ spanning trees of G such that for any two vertices u, v of G a spanning tree $T \in \mathcal{T}(G)$ exists

[⋆] This work was supported by the European Regional Development Fund (ERDF) and by NSERC.

H. Broersma et al. (Eds.): WG 2008, LNCS 5344, pp. 110–121, 2008.

such that the distance in T between u and v is at most r plus their distance in G (see [14]). We say that system $\mathcal{T}(G)$ collectively $+r$-spans the graph G and that G is (collectively) $+r$-spanned by $\mathcal{T}(G)$. Clearly, if G admits a system of μ collective additive tree r-spanners, then G admits an additive r-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 r-spanner.

Collective tree spanners were investigated for a number of particular graph classes, including planar graphs, bounded chordality graphs, bounded genus graphs, bounded treewidth graphs, AT-free graphs and others (see [10, 12, 13, 14, 18]). Some families of graphs admit a constant number and some admit a logarithmic number of collective additive tree r-spanners, for small values of r.

One of the motivations to introduce this concept stems from the problems of designing compact and efficient distance and routing labeling schemes in graphs. A distance labeling scheme for trees is described in [21] that assigns each vertex of an n-vertex tree an $O(\log^2 n)$-bit label such that, given the labels of two vertices x and y, it is possible to compute in constant time, based solely on these two labels, the distance in the tree between x and y. A shortest path routing labeling scheme for trees is described in [27] that assigns each vertex of an n-vertex tree an $O(\log^2 n/\log\log n)$-bit label such that, 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. Hence, if an n-vertex graph G admits a system of μ collective additive tree r-spanners, then G admits an additive r-approximate distance labeling scheme with the labels of size $O(\mu \log^2 n)$ bits per vertex and an $O(\mu)$ time distance decoder. Furthermore, G admits an additive r-approximate routing labeling scheme with the labels of size $O(\mu \log^2 n/\log\log n)$ bits per vertex. Once computed by the sender in $O(\mu)$ 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 (see [13, 14]).

Other motivations stem from the generic problems of efficient representation of the distances in "complicated" graphs by the tree distances and of algorithmic use of these representations [1, 2, 5, 16]. Approximating a graph distance d_G by simpler distances (in particular, by tree–distances d_T) is useful in many areas such as communication networks, data analysis, motion planning, image processing, network design, and phylogenetic analysis (see [3,4,6,9,19,20,22,23,25,26]). An arbitrary metric space (in particular a finite metric defined by a graph) might not have enough structure to exploit algorithmically.

In this paper, we examine the problem of finding "small" systems of collective additive tree r-spanners for small values of r on *circle graphs* and on *polygonal graphs*. Circle graphs are known as the intersection graphs of chords in a circle [17]. For any fixed integer $k \geq 2$, the class of k-polygonal graphs can be defined as the intersection graphs of chords inside a convex k-polygon, where the endpoints of each chord lie on two different sides [15]. Note that permutation graphs are exactly 2-polygonal graphs and any n-vertex circle graph is a k-polygonal graph for some $k \leq n$. Our results are the following.

- For any constant c, there are circle graphs that cannot be collectively $+c$-spanned by any constant number of spanning trees.
- Every n-vertex circle graph G admits a system of at most $2\log_{\frac{3}{2}} n$ collective additive tree 2-spanners, constructible in polynomial time.
- There are circle graphs on n vertices for which any system of collective additive tree 1-spanners will require $\Omega(n)$ spanning trees.
- Every n-vertex circle graph admits an additive 2-spanner with at most $O(n \log n)$ edges.
- Every n-vertex k-polygonal graph admits a system of at most $2\log_{\frac{3}{2}} k + 7$ collective additive tree 2-spanners, constructible in polynomial time.
- Every n-vertex k-polygonal graph admits an additive $(k+6)$-spanner with at most $6n-6$ edges and an additive $(k/2+8)$-spanner with at most $10n-10$ edges, constructible in polynomial time.
- Every n-vertex 4-polygonal graph admits a system of at most 5 collective additive tree 2-spanners, constructible in linear time.
- Every n-vertex 3-polygonal graph admits a system of at most 3 collective additive tree 2-spanners and an additive tree 6-spanner, constructible in linear time.

2 Preliminaries

All graphs occurring in this paper are connected, finite, undirected, loopless and without multiple edges. In a graph $G = (V, E)$ the *length* of a path from a vertex v to a vertex u 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 path connecting u and v. For a vertex v of G, the sets $N_G(v)$ and $N_G[v] = N_G(v) \cup \{v\}$ are called the *open neighborhood* and the *closed neighborhood* of v, respectively. For a set $S \subseteq V$, by $N_G[S] = \bigcup_{v \in S} N_G[v]$ we denote the closed neighborhood of S and by $G(S)$ the subgraph of G induced by vertices of S. Let also $G \setminus S$ be the graph $G(V \setminus S)$ (which is not necessarily connected).

An graph G is called a *circle graph* if it is the intersection graph of a finite collection of chords of a circle [17] (see Fig. 1 for an illustration). Without loss of generality, we may assume that no two chords share a common endpoint. For any fixed integer $k \geq 3$, the class of *k-polygonal* (or *k-gon*) *graphs* is defined as the intersection graphs of chords inside a convex k-polygon, where the endpoints of each chord lie on two different sides [15] (see Fig. 2 for an illustration). *Permutation graphs* can be considered as 2-gon graphs as they are the intersection graphs of chords between two sides (or sides of a degenerate 2-polygon). Again, without loss of generality, we may assume that no two chords share a common endpoint. Clearly, if a graph G is a k-gon graph, it is also a k'-gon graph with $k' > k$, but the reverse is not necessarily true.

Let $G = (V, E)$ be a permutation graph with a given permutation model Π. Let L' and L'' be the two sides of Π. A vertex s of G is called *extreme* if at least one endpoint of the chord of Π, corresponding to s, is the leftmost or the rightmost endpoint either on L' or on L''. The following result was presented in [13]:

Lemma 1. **[13]** *Let G be a permutation graph and let s be an extreme vertex of G in some permutation model. Then, there exists a $BFS(s)$-tree of G, constructible in linear time, which is an additive tree 2-spanner of G.*

Since an induced cycle on 4 vertices is a permutation graph, permutation graphs cannot have any additive tree r-spanner for $r < 2$. Clearly, given any $BFS(s)$-tree T_s of G, $d_{T_s}(x, s) = d_G(x, s)$ holds for any $x \in V$.

3 Additive Spanners for Circle Graphs

In this section, we show that every n-vertex circle graph G admits a system of at most $2\log_{\frac{3}{2}} n$ collective additive tree 2-spanners. This upper bound result is complemented also with two lower bound results.

We start with the main lemma of this section which is also of independent interest.

Lemma 2. *Every n-vertex ($n \geq 2$) circle graph $G = (V, E)$ has two vertices a and b such that $S = N_G[a, b]$ is a balanced separator of G, i.e. each connected component of $G \setminus S$ has at most $\frac{2}{3}n$ vertices.*

Proof. Consider an intersection model $\phi(G)$ of G and let $\mathcal{C}$ be the circle in that model. Let also $\mathcal{P} := (p_1, p_2, \ldots, p_{2n})$ be the sequence in clockwise order of the $2n$ endpoints of the chords representing the vertices of G in $\phi(G)$. We divide the circle $\mathcal{C}$ into three circular arcs B (bottom), L (left) and R (right) each containing at most $\lceil\frac{2}{3}n\rceil$ consecutive endpoints (see Fig. 1 for an illustration). We say that a chord of $\phi(G)$ is an XY-chord if its endpoints lie on arcs X and Y ($X, Y \in \{B, L, R\}$) of $\mathcal{C}$. If v is an XY-chord then let v_X and v_Y be its endpoints on X and Y, respectively.

Let X be an arc from the set of arcs $\{B, L, R\}$. Since G is a connected graph, for any X, there must exist a chord in $\phi(G)$ with one endpoint in X and the other endpoint not in X. Moreover, since we have three arcs (B, L, R), there must exist an arc X in $\{B, L, R\}$ which has both types of chords: between X and $Y \in \{B, L, R\} \setminus \{X\}$ and between X and $Z \in \{B, L, R\} \setminus \{X, Y\}$. Assume, without loss of generality, that $X = B$. Let p be the point of $\mathcal{C}$ separating arcs L and R (see Fig. 1). Now choose a BL-chord a in $\phi(G)$ with endpoint a_L closest to p and choose a BR-chord b in $\phi(G)$ with endpoint b_R closest to p. By a, b we also denote the vertices of G which correspond to chords a and b.

Points a_B, a_L, b_R and b_B of $\mathcal{C}$ divide $\mathcal{C}$ into four arcs. We name these four arcs A_U, A_R, A_D and A_L. The arc $A_U := (a_L, b_R)$ is formed by all points of $\mathcal{C}$ from a_L to b_R in clockwise order. If chords a and b intersect, then we set $A_R := (b_R, a_B)$, $A_D := (a_B, b_B)$, and $A_L := (b_B, a_L)$ (all arcs begin at the left arc-endpoint and go clockwise to the right arc-endpoint). If chords a and b do not intersect, then

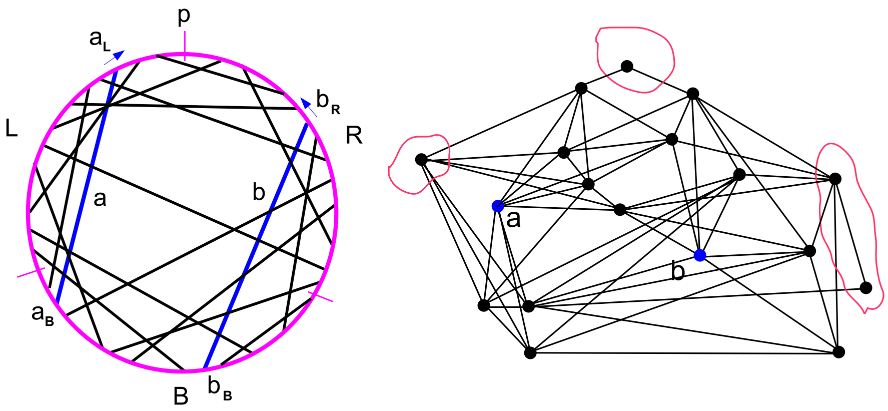


Fig. 1. A circle graph with an intersection model and two special chords a and b. A balanced separator $S = N_G[a, b]$ and the connected components of $G \backslash S$ are also shown.

set $A_R := (b_R, b_B)$, $A_D := (b_B, a_B)$, and $A_L := (a_B, a_L)$. We consider these arcs as open arcs, i.e., the points a_B, a_L, b_R and b_B do not belong to them.

By our choices of a and b, we guarantee that $\phi(G)$ has no chords with one endpoint in A_U and the other one in A_D (regardless of the adjacency of a and b). Denote by V_Y all chords from $\phi(G)$ (vertices of G) whose both endpoints are in A_Y, where Y is either U, or R, or D, or L. Then, it is easy to see that in G, the set $S := N_G[a, b]$ separates vertices of V_Y from vertices of $V_{Y'}$, where $Y, Y' \in \{U, R, D, L\}$, $Y \neq Y'$. Now, since A_L is a sub-arc of arc $B \cup L$, A_U is a sub-arc of arc $L \cup R$, A_R is a sub-arc of arc $R \cup B$, A_D is a sub-arc of arc B, and arcs A_U, A_R, A_D and A_L do not contain points a_B, a_L, b_R and b_B, we conclude that $|A_L \cap \mathcal{P}| \leq \frac{4}{3}n$, $|A_U \cap \mathcal{P}| \leq \frac{4}{3}n$, $|A_R \cap \mathcal{P}| \leq \frac{4}{3}n$ and $|A_D \cap \mathcal{P}| \leq \frac{2}{3}n$. Hence, the number of arcs in $\phi(G)$ whose both endpoints are in A_L (respectively, in A_U, A_R, A_D), and therefore the number of vertices in V_L ((respectively, in V_U, V_R, V_D), is at most $\frac{2}{3}n$. □

In [12], a large class of graphs, called *(α, γ, r)-decomposable graphs*, was defined, and it was proven that any (α, γ, r)-decomposable graph G with n vertices admits a system of at most $\gamma \log_{1/\alpha} n$ collective additive tree $2r$-spanners. Let α be a positive real number smaller than 1, γ be a positive integer and r be a non-negative integer. We say that an n-vertex graph G is (α, γ, r)*–decomposable* if $n \leq \gamma$ or there is a separator $S \subseteq V$ in G, such that the following three conditions hold:

- the removal of S from G leaves no connected component with more than αn vertices;
- there exists a subset $D \subseteq V$ such that $|D| \leq \gamma$ and for any vertex $u \in S$, $d_G(u, D) \leq r$;
- each connected component of $G \setminus S$ is an (α, γ, r)–decomposable graph, too.

Since, any subgraph of a circle graph is also a circle graph, and, by Lemma 2, each n-vertex circle graph $G = (V, E)$ admits a separator $S = N_G[D]$ (where

$D = \{a, b\}$, $a, b \in V$), such that no connected component of $G \setminus S$ has more than $\frac{2}{3}n$ vertices, we immediately conclude.

Corollary 1. *Every circle graph is* $(\frac{2}{3}, 2, 1)-$*decomposable.*

Theorem 1. *Every n-vertex circle graph G admits a system $\mathcal{T}(G)$ of at most* $2\log_{\frac{3}{2}} n$ *collective additive tree 2-spanners.*

Note that such a system of spanning trees $\mathcal{T}(G)$ for a n-vertex m-edge circle graph G, given together with an intersection model $\phi(G)$, can be constructed in $O(m \log n)$ time, since a balanced separator $S = N_G[a, b]$ of G can be found in linear $O(m)$ time (see [12] for details of the construction).

Taking the union of all these spanning trees in $\mathcal{T}(G)$, we also obtain a sparse additive 2-spanner for a circle graph G.

Corollary 2. *Every n-vertex circle graph G admits an additive 2-spanner with at most* $2(n-1)\log_{\frac{3}{2}} n$ *edges.*

We complement our upper bound result with the following lower bounds.

Proposition 1. *There are circle graphs on n vertices for which any system of collective additive tree 1-spanners will require $\Omega(n)$ spanning trees.*

Proof. Since complete bipartite graphs are circle graphs, we can use the lower bound shown in [14] for complete bipartite graphs. It says that any system of collective additive tree 1-spanners will need to have $\Omega(n)$ spanning trees for each complete bipartite graph on n vertices. □

Proposition 2. *For any constant c, there are circle graphs that cannot be collectively $+c$-spanned by any constant number of spanning trees.*

In [7] the authors show that a similar proposition holds for weakly chordal graphs. In fact, the same proof works for circle graphs.

4 Additive Spanners for *k*-Gon and for 3-Gon Graphs

4.1 Additive Spanners for *k*-Gon Graphs

In this section, among other results, we show that every n-vertex k-gon graph G admits a system of at most $2\log_{\frac{3}{2}} k + 7$ collective additive tree 2-spanners, an additive $(k + 6)$-spanner with at most $6n - 6$ edges, and an additive $(k/2 + 8)$-spanner with at most $10n - 10$ edges. We will assume, in what follows, that our k-gon graph G is given together with its intersection model. Let $\mathcal{P}$ be the closed polygonal chain (the boundary) of the k-polygon in that model. The vertices of the k-polygon will be called the *corners*. The idea of the construction here is similar to the one used in Theorem 1. Yet, here we operate with the corners of the model rather than with the endpoints of the chords. More precisely, one can show that there are vertices a and b in the graph, such that $N_G[a, b]$ forms a

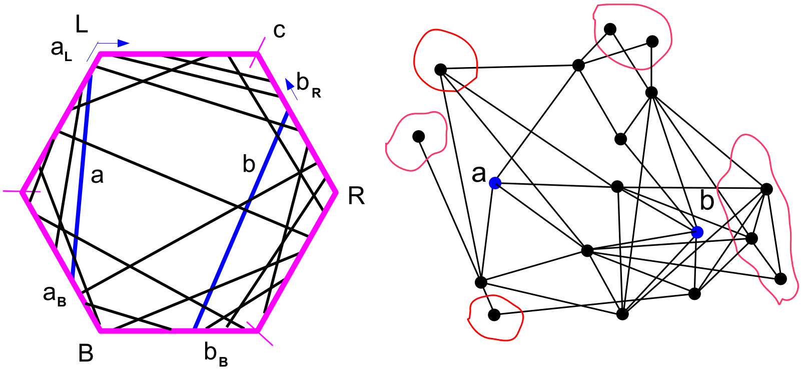


Fig. 2. A 6-gon graph with an intersection model and two special chords a and b. A balanced separator $S = N_G[a, b]$ and the connected components of $G \backslash S$ are also shown.

balanced (with respect to the number of corners) separator of G, similar to the one in Lemma 2 (see Fig. 2 for an illustration).

It is an easy observation that each of the so-created connected components is itself again a k'-gon graph for some $k' < k$. By applying this separator property recursively, we end up with a set of separators and some permutation graphs. By Lemma 1, permutation graphs have good additive tree spanners. We can put these trees together appropriately and can thereby show the following theorem (see [11] for details).

Theorem 2. *Every n-vertex m-edge k-gon graph G admits a system of at most $2\log_{\frac{3}{2}} k + 7$ collective additive tree 2-spanners, constructable in $O(m \log k)$ time. Moreover, every 3-gon graph admits a system of no more than 3 collective additive tree 2-spanners, and every 4-gon graph admits a system of no more than 5 collective additive tree 2-spanners.*

Similar to Corollary 2, we can merge the edges of all the spanning trees to create a single spanning graph.

Corollary 3. *Every n-vertex k-gon graph G admits an additive 2-spanner with at most $(2\log_{\frac{3}{2}} k + 7)(n-1)$ edges. Moreover, every 3-gon graph admits an additive 2-spanner with at most $3(n-1)$ edges, and every 4-gon graph admits an additive 2-spanner with at most $5(n-1)$ edges.*

Instead of using the recursive separation algorithm all the way to permutation graphs, one can also stop at an earlier stage. By using properties of k-gon graphs (see [15]) and some general results on spanners of graphs with bounded length of largest induced cycle (see [8]) we can show the following theorem.

Theorem 3. *Every n-vertex m-edge k-gon graph G admits an additive $(2((\frac{2}{3})^\ell k + 4(1-(\frac{2}{3})^\ell)) + 1)$-spanner with at most $2(\ell+1)(n-1)$ edges, for each $0 \le \ell \le \log_{3/2} k + 3$. Moreover, such a sparse spanner is constructable in $O(m \log k)$ time.*

Finally, when choosing ℓ equal to $0, 1, 2, 3$ or 4 in Theorem 3, we obtain:

Corollary 4. *Every n-vertex k-gon graph G admits an additive $(2k+1)$-spanner with at most $2n-2$ edges, an additive $(\frac{4}{3}k+4)$-spanner with at most $4n-4$ edges, an additive $(\frac{8}{9}k+6)$-spanner with at most $6n-6$ edges, an additive $(\frac{16}{27}k+7)$-spanner with at most $8n-8$ edges, and an additive $(\frac{32}{81}k+8)$-spanner with at most $10n-10$ edges.*

4.2 Additive Tree Spanners for 3-Gon Graphs

In this section, we show that any connected 3-gon graph G admits an additive tree 6-spanner constructible in linear time. Due to space restrictions we have to omit some parts of the proof here, and refer instead the reader to [11] for a complete proof of this result. Note that, since an induced cycle on 6 vertices is a 3-gon graph, 3-gon graphs cannot have any additive tree r-spanner for $r < 4$. The idea of the construction is as follows. The algorithm will identify permutation graphs in each of the 3 corners of the 3-gon and use the algorithm presented in Lemma 1 to construct effective tree spanners of each of these subgraphs. These 3 tree spanners are incorporated into a tree spanner for the entire graph by analyzing the structure in the "center" of the given 3-gon graph.

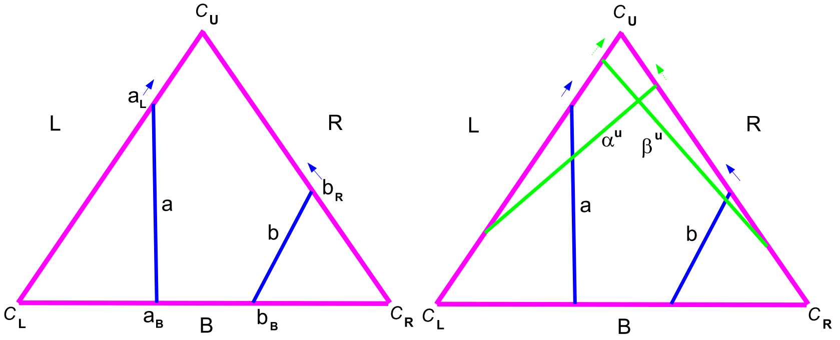


Fig. 3. A 3-gon intersection model Δ with special chords a, b, α^u and β^u

Let $G = (V, E)$ be a connected 3-gon graph. We may assume that G is not a permutation graph. Consider a 3-gon intersection model Δ of G and fix an orientation of Δ. Denote by L (left), R (right) and B (bottom) the corresponding sides of the 3-gon Δ, and by C_L, C_R and C_U the left, right and upper corners of Δ. We say that a chord of Δ is an XY-chord if its endpoints lie on sides X and Y of Δ. If v is an XY-chord then let v_X and v_Y be its endpoints on X and Y, respectively. Since G is not a permutation graph, we must have all three types of chords in Δ: LR-chords, LB-chords and RB-chords. Let a be the LB-chord of G whose endpoint on L is closest to the upper corner C_U of Δ. Let b be the RB-chord of G whose endpoint on R is closest to the upper corner of Δ (see the left 3-gon in Fig. 3 for an illustration). Note that a and b may or may not cross. By a, b we also denote the corresponding vertices of G.

Let V_U be the subset of LR-chords of Δ (of vertices of G) with endpoints in segments (a_L, C_U) and (b_R, C_U). We will add at most two more LR-chords to V_U to form a permutation graph named G_U. Choose (if it exists) an LR-chord α^u in Δ such that α^u_L belongs to segment (C_L, a_L) of L, α^u_R belongs to segment (C_U, b_R) of R and α^u_R is closest to the corner C_U. Clearly, if α^u exists then it must intersect a (but not b). Analogously, choose (if it exists) an LR-chord β^u in Δ such that β^u_R belongs to segment (C_R, b_R) of R, β^u_L belongs to segment (C_U, a_L) of L and β^u_L is closest to the corner C_U. Again, if β^u exists then it must intersect b (but not a). Note that, if $V_U \neq \emptyset$, then at least one of $\{\alpha^u, \beta^u\}$ must exist (since otherwise, G is not connected), and if both chords exist then they must intersect each other. See the right picture in Fig. 3. Now, we define our permutation graph G_U to be the subgraph of G induced by vertices $V_U \cup \{\alpha^u, \beta^u\}$, assuming that $V_U \neq \emptyset$ (see Fig. 4 for an illustration). If $V_U = \emptyset$, then we set G_U to be an empty graph.

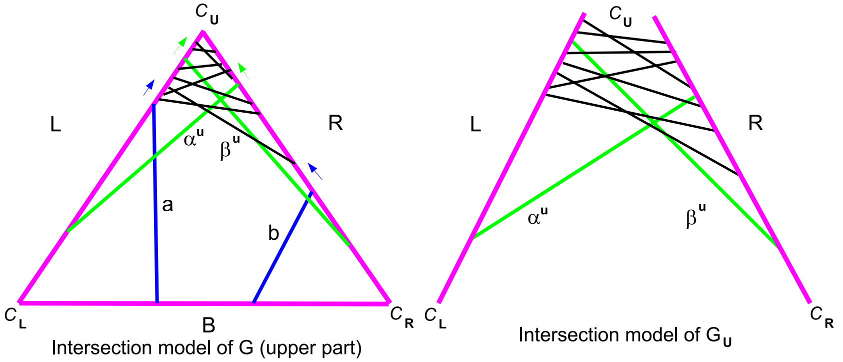
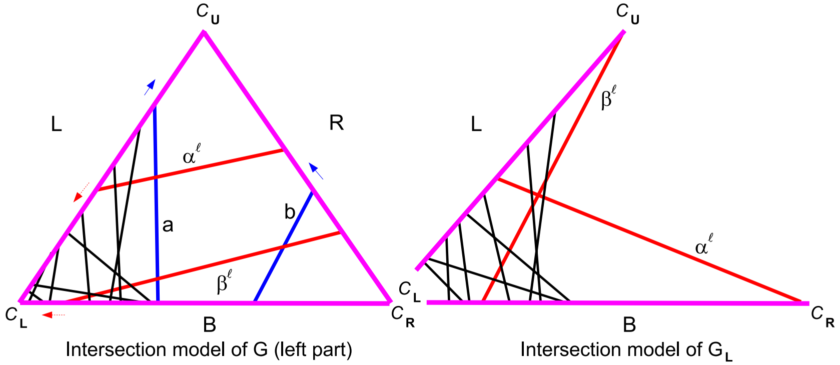


Fig. 4. Permutation graph G_U extracted from G

Define s^u to be a vertex from $\{\alpha^u, \beta^u\}$ as follows: if both α^u and β^u exist, then if α^u has a neighbor in V_U which is not a neighbor of β^u, set $s^u := \alpha^u$; otherwise, set $s^u := \beta^u$. Since G_U is a permutation graph and s^u is extreme, by Lemma 1, G_U has a linear time constructable $BFS(s^u)$-tree T_U such that $d_{T_U}(x, y) \leq d_{G_U}(x, y) + 2$ and $d_{T_U}(x, s^u) = d_{G_U}(x, s^u)$ for any x, y in $V_U \cup \{\alpha^u, \beta^u\}$.

Let V_L be the subset of all chords of Δ (of vertices of G) with endpoints in segments (C_L, a_L) and $(C_L, a_B) \cap (C_L, b_B)$. We will add at most two more chords to V_L to form a permutation graph named G_L. Choose (if it exists) a chord α^ℓ in Δ such that one endpoint of α^ℓ belongs to segment (C_L, a_L) of L, the other endpoint belongs to $R \cup (a_B, C_R) \cup (b_B, C_R)$ and α^ℓ_L is closest to the corner C_L. Equivalently, among all chords of Δ intersecting a or b, α^ℓ is chosen to be the chord with an endpoint α^ℓ_L in (C_L, a_L) closest to C_L. Note that α^ℓ may or may not cross b. Also, choose (if it exists) an RB-chord β^ℓ in Δ such that β^ℓ_R belongs to segment (C_R, b_R) of R, β^ℓ_B belongs to segment $(C_L, a_B) \cap (C_L, b_B)$ of B and

β^ℓ_B is closest to the corner C_L. Notice, if β^ℓ exists then it must intersect both a and b. Furthermore, if $V_L \neq \emptyset$, then at least one chord from $\{\alpha^\ell, \beta^\ell\}$ must exist (since, otherwise, G is not connected). Now, we define our permutation graph G_L. If $V_L = \emptyset$, then set G_L to be an empty graph. Otherwise, G_L is set to be the subgraph of G induced by vertices $V_L \cup \{\alpha^\ell, \beta^\ell\}$ with one extra edge $(\alpha^\ell, \beta^\ell)$ added if it was not already an edge of G (see Fig. 5 for an illustration).



Intersection model of G (left part)

Intersection model of G_L

Fig. 5. Permutation graph G_L obtained from G

Define s^ℓ to be a vertex from $\{\alpha^\ell, \beta^\ell\}$ as follows: if both α^ℓ and β^ℓ exist, then if α^ℓ has a neighbor in V_L which is not a neighbor of β^ℓ, then set $s^\ell := \alpha^\ell$; otherwise, set $s^\ell := \beta^\ell$. Since G_L is a permutation graph, by Lemma 1, G_L has a linear time constructable $BFS(s^\ell)$-tree T_L such that $d_{T_L}(x, y) \leq d_{G_L}(x, y) + 2$ and $d_{T_L}(x, s^\ell) = d_{G_L}(x, s^\ell)$ for any x, y in $V_L \cup \{\alpha^\ell, \beta^\ell\}$.

Taking symmetry into account, similar to α^ℓ, β^ℓ and G_L, we can define for the corner C_R of Δ two specific chords α^r, β^r and a permutation graph G_R. We will have β^r adjacent to both a and b, and α^r adjacent to a or b. Define s^r to be a vertex from $\{\alpha^r, \beta^r\}$, and if both α^r and β^r exist, then if α^r has a neighbor in V_R which is not a neighbor of β^r, then set $s^r := \alpha^r$; otherwise, set $s^r := \beta^r$. Again, by Lemma 1, there is a linear time constructable $BFS(s^r)$-tree T_R of G_R such that $d_{T_R}(x, y) \leq d_G(x, y) + 2$ and $d_{T_R}(x, s^r) = d_G(x, s^r)$ for any x, y in $V_R \cup \{\alpha^r, \beta^r\}$.

Now we create a spanning tree T of G from the trees T_U, T_L and T_R as follows. Initially, T is just the union of T_U, T_L and T_R. We know that $\{\beta^\ell, \beta^r, \alpha^u\} \subseteq N_G(a)$ and $\{\beta^\ell, \beta^r, \beta^u\} \subseteq N_G(b)$. Make vertex a adjacent to α^u and vertex b adjacent to β^u in T, and denote $M := \{\alpha^\ell, \beta^\ell, \alpha^r, \beta^r\}$. If $M \subseteq N_G(a)$, then make vertex a adjacent in T to each vertex in M. If $M \setminus N_G(a) \neq \emptyset$ but $M \subseteq N_G(b)$, then make vertex b adjacent in T to each vertex in M. If neither $M \subseteq N_G(a)$ nor $M \subseteq N_G(b)$, then make vertices α^ℓ, β^ℓ adjacent in T to a common neighbor in $\{a, b\}$ and vertices α^r, β^r adjacent in T to a common neighbor in $\{a, b\}$. Remove from T the edge $(\alpha^\ell, \beta^\ell)$ (it was a part of tree T_L if both α^ℓ and β^ℓ existed) and the edge (α^r, β^r) (it was a part of tree T_R if both α^r and β^r existed).

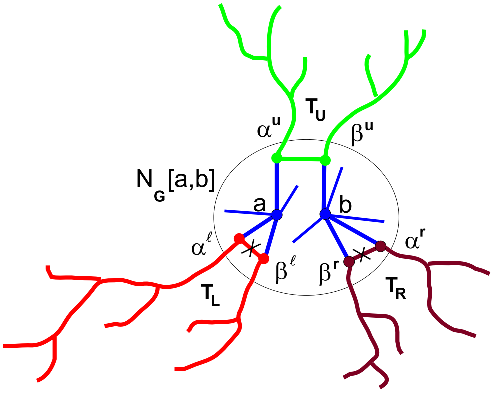


Fig. 6. Trees T_L, T_R and T_U connected via $N_G[a,b]$ to form a tree spanner of G

If a and b are adjacent in G, then add edge (a,b) to T. If a is not adjacent to b in G but $d_G(a,b) = 2$, then choose a common neighbor z of a and b in $N_G[a,b]$ and add edges (a,z) and (b,z) to T. In these cases, i.e., when $d_G(a,b) \leq 2$, remove the possible edge (α^u, β^u) from T (it was a part of the tree T_U if both α^u and β^u existed). If $d_G(a,b) > 2$ then $d_G(a,b) = 3$, chords β^ℓ, β^r do not exist and the edge (α^u, β^u) from T_U goes to T if both chords α^u and β^u exist. If one of these chords does not exist, then there must be two vertices x, y that are adjacent in G, with $x \in N_G(b)$ and $y \in N_G(a)$ (see [11] for details) and we put the edge (x,y) into T. Finally, make all vertices from $N_G(a) \setminus \{\alpha^\ell, \beta^\ell, \alpha^u, \beta^u, \alpha^r, \beta^r, b, z\}$ adjacent to a in T and all remaining vertices from $N_G(b)$ (i.e., those that are not adjacent to a in T) adjacent to b; see Fig. 6 for an illustration. Clearly, T constructed this way is a spanning tree of G. Using a careful analysis, we can show that T is an additive tree 6-spanner of G (see [11] for the complete proof).

Theorem 4. *Any connected 3-gon graph G admits an additive tree 6-spanner constructible in linear time.*

References

1. Bartal, Y.: Probabilistic approximations of metric spaces and its algorithmic applications. In: FOCS 1996, pp. 184–193 (1996)
2. Bartal, Y.: On approximating arbitrary metrices by tree metrics. In: STOC 1998, pp. 161–168 (1998)
3. Barthélemy, J.-P., Guénoche, A.: Trees and Proximity Representations. Wiley, New York (1991)
4. Bhatt, S., Chung, F., Leighton, F., Rosenberg, A.: Optimal simulations of tree machines. In: FOCS 1986, pp. 274–282 (1986)
5. Charikar, M., Chekuri, C., Goel, A., Guha, S., Plotkin, S.: Approximating a Finite Metric by a Small Number of Tree Metrics. In: FOCS 1998, pp. 379–388 (1998)
6. Cai, L., Corneil, D.G.: Tree spanners. SIAM J. Discrete Math. 8, 359–387 (1995)

7. Corneil, D.G., Dragan, F.F., Köhler, E., Xiang, Y.: Lower Bounds for Collective Additive Tree Spanners (in preparation)
8. Chepoi, V.D., Dragan, F.F., Yan, C.: Additive Sparse Spanners for Graphs with Bounded Length of Largest Induced Cycle. Theoretical Computer Science 347, 54–75 (2005)
9. Chew, L.P.: There are planar graphs almost as good as the complete graph. J. of Computer and System Sciences 39, 205–219 (1989)
10. Corneil, D.G., Dragan, F.F., Köhler, E., Yan, C.: Collective tree 1-spanners for interval graphs. In: Kratsch, D. (ed.) WG 2005. LNCS, vol. 3787, pp. 151–162. Springer, Heidelberg (2005)
11. Dragan, F.F., Corneil, D.G., Köhler, E., Xiang, Y.: Additive Spanners for Circle Graphs and Polygonal Graphs (manuscript, 2008), `http://www.cs.kent.edu/~dragan/Coll-Spanners-Circle.pdf`
12. Dragan, F.F., Yan, C.: Collective Tree Spanners in Graphs with Bounded Genus, Chordality, Tree-width, or Clique-width. In: Deng, X., Du, D.-Z. (eds.) ISAAC 2005. LNCS, vol. 3827, pp. 583–592. Springer, Heidelberg (2005)
13. Dragan, F.F., Yan, C., Corneil, D.G.: Collective Tree Spanners and Routing in AT-free Related Graphs. Journal of Graph Algorithms and Applications 10, 97–122 (2006)
14. Dragan, F.F., Yan, C., Lomonosov, I.: Collective tree spanners of graphs. SIAM J. Discrete Math. 20, 241–260 (2006)
15. Elmallah, E.S., Stewart, L.: Polygon Graph Recognition. Journal of Algorithms 26, 101–140 (1998)
16. Fakcharoenphol, J., Rao, S., Talwar, K.: A tight bound on approximating arbitrary metrics by tree metrics. In: STOC 2003, pp. 448–455 (2003)
17. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs, 2nd edn. Elsevier, Amsterdam (2004)
18. Gupta, A., Kumar, A., Rastogi, R.: Traveling with a Pez Dispenser (or, Routing Issues in MPLS). SIAM J. Comput. 34, 453–474 (2005)
19. Liestman, A.L., Shermer, T.: Additive graph spanners. Networks 23, 343–364 (1993)
20. Peleg, D.: Distributed Computing: A Locality-Sensitive Approach. SIAM Monographs on Discrete Math. Appl. SIAM, Philadelphia (2000)
21. Peleg, D.: Proximity-Preserving Labeling Schemes and Their Applications. J. of Graph Theory 33, 167–176 (2000)
22. Peleg, D., Schäffer, A.A.: Graph Spanners. J. Graph Theory 13, 99–116 (1989)
23. Peleg, D., Ullman, J.D.: An optimal synchronizer for the hypercube. In: PODC 1987, pp. 77–85 (1987)
24. Prisner, E.: Distance approximating spanning trees. In: Reischuk, R., Morvan, M. (eds.) STACS 1997. LNCS, vol. 1200, pp. 499–510. Springer, Heidelberg (1997)
25. Sneath, P.H.A., Sokal, R.R.: Numerical Taxonomy. W.H. Freeman, San Francisco (1973)
26. Swofford, D.L., Olsen, G.J.: Phylogeny reconstruction. In: Hillis, D.M., Moritz, C. (eds.) Molecular Systematics, pp. 411–501. Sinauer Associates Inc., Sunderland (1990)
27. Thorup, M., Zwick, U.: Compact routing schemes. In: SPAA 2001, pp. 1–10 (2001)