Additive Spanners for Circle Graphs and Polygonal Graphs*

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

[1] Department of Computer Science, University of Toronto, Toronto, Ontario, Canada
dgc@cs.toronto.edu
[2] Algorithmic Research Laboratory, Department of Computer Science,
Kent State University, Kent, OH 44242, USA
dragan@cs.kent.edu, yxiang@cs.kent.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 [18]. If H is a tree then it is called an *additive tree r-spanner* of G [23]. (A similar definition can be given for multiplicative t-spanners [9, 21, 22] and for multiplicative tree t-spanners [6].) In this paper, we continue the approach taken in [10–13, 17] 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 such that the distance in T between u and v is at most r plus their distance in G (see [13]). We say that system $\mathcal{T}(G)$ collectively c-spans the graph 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 [10–13, 17]). 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.

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

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 [20] 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 [26] 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 [12, 13]).

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, 15]. 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, 18, 19, 21, 22, 24, 25]). 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 [16]. For any fixed integer $k \geq 2$, the class of k-polygon 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 [14]. 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 [16] (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-polygon* (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 [14] (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 [12]:

Lemma 1. **[12]** *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, since T_s is a $BFS(s)$-tree 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.

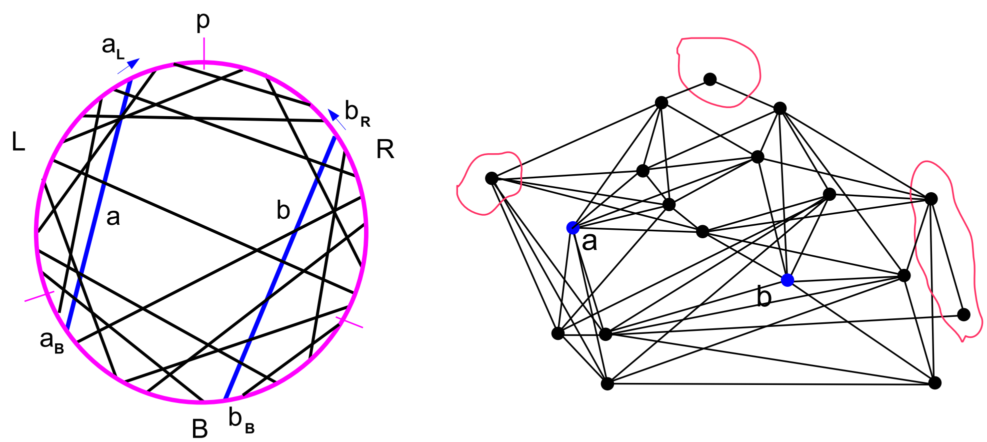

Fig. 1. A circle graph with an intersection model and two special chords a and b. A balanced separator $S = N_G[a, b]$ and the connected components of $G \setminus S$ are also shown.

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 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 $\frac{2}{3}n$. $\square$

In [11], 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 [11] 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 [13] 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 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.

Lemma 3. *Every n-vertex ($n \geq 2$) k-gon graph $G = (V, E)$ has two vertices a and b such that $S = N_G[a, b]$ is a separator of G and each connected component of $G \setminus S$ is a k'-gon graph with $k' \leq 2\lceil \frac{1}{3}k \rceil$, when $k > 5$, and $k' = k - 1$ when $k = 3, 4, 5$.*

Proof. Consider an intersection model $\rho(G)$ of G and let $\mathcal{P}$ be the closed polygonal chain (the boundary) of the k-polygon in that model. The vertices of the k-polygon, in what follows, are called the *corners*. Let $\mathcal{C} := [c_1, c_2, \ldots, c_k]$ be the sequence in the clockwise order of the corners of $\mathcal{P}$. The proof of this lemma is similar to the proof of Lemma 2, but here we operate with the corners rather than with the endpoints of the chords. We divide the closed polygonal chain $\mathcal{P}$ into three polygonal sub-chains $B := (c_1, \ldots, c_{k_1})$, $L := (c_{k_1}, \ldots, c_{k_2})$ and $R := (c_{k_2}, \ldots, c_k, c_1)$ each containing at most $\lceil \frac{1}{3}k \rceil + 1$ consecutive corners (see Fig. 2 for an illustration). We say that a chord of $\rho(G)$ is an XY-chord if its endpoints lie on poly-chains X and Y ($X, Y \in \{B, L, R\}$) of $\mathcal{P}$. If v is an XY-chord then let v_X and v_Y be its endpoints on X and Y, respectively.

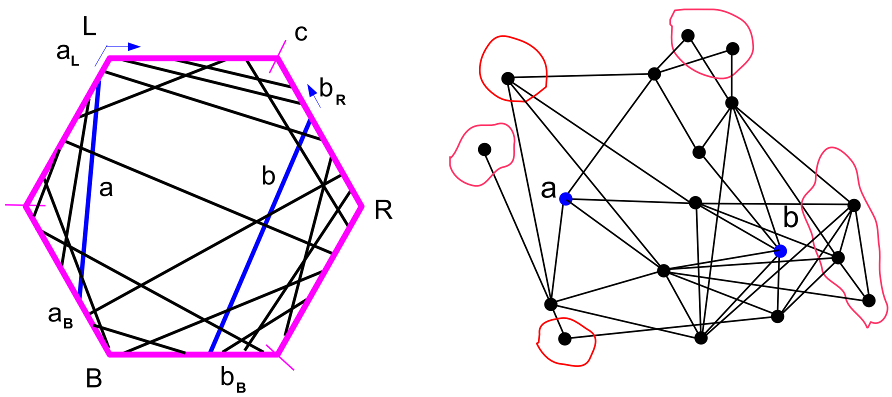

Fig. 2. A 6-gon graph with an intersection model and two special chords a and b. A balanced separator $S = N_G[a, b]$ and the connected components of $G \setminus S$ are also shown.

As in the proof of Lemma 2, we may assume, without loss of generality, that there are both BL- and BR-chords in $\rho(G)$. Choose BL-chord a in $\rho(G)$ with endpoint a_L closest to the corner $c := c_{k_2}$. Choose BR-chord b in $\rho(G)$ with endpoint b_R closest to the corner c (see Fig. 2). By a, b we denote also the vertices of G which correspond to chords a and b. Using the same arguments as in the proof of Lemma 2, we can see that the removal of chords a and b and all the chords intersecting them, divides the remaining chords of $\rho(G)$ into four pairwise disjoint groups V_D, V_L, V_U and V_R; no chord of one group intersects any chord of another group. Hence, each connected component of the graph $G \setminus N_G[a, b]$ has its vertices entirely in either V_D or V_L or V_U or V_R. Since all chords of V_L are between no more that $2\lceil \frac{1}{3}k \rceil$ consecutive sides of $\mathcal{P}$, the induced subgraph $G(V_L)$ of G is a k'-gon graph for $k' \leq 2\lceil \frac{1}{3}k \rceil$ (and $k' = k - 1$ when $k = 3, 4, 5$). The same is true for the induced subgraphs $G(V_U)$, $G(V_R)$ and $G(V_D)$ of G. $\square$

Consider the following procedure which constructs for any k-gon graph G a hierarchy of subgraphs of G and a system of local shortest path trees.

Procedure 1. Construct for a k-gon graph G a system of local shortest path trees and a system of r-gon subgraphs ($r < k$).

Input: A k-gon graph G with an intersection model $\rho(G)$ and a positive integer $r \geq 2$.
Output: A system of local shortest path trees and a system of r-gon subgraphs of G.
Method:

set $i := 0$; $\mathcal{G}_i := \{G\}$; $\mathcal{T} := \emptyset$; $\mathcal{F} := \emptyset$;
while $\mathcal{G}_i \neq \emptyset$ **do**
  set $\mathcal{G}_{i+1} := \emptyset$; $\mathcal{T}_i' := \emptyset$; $\mathcal{T}_i'' := \emptyset$; $\mathcal{F}_i := \emptyset$;
  for each $G' \in \mathcal{G}_i$ **do**
    if G' is an r-gon subgraph of G (i.e., all chords of $\rho(G')$ are between at most r sides of $\rho(G)$)
    then add G' into $\mathcal{F}_i$;
    else
      find special vertices a and b in G' as described in the proof of Lemma 3;
      construct a shortest path tree of G' rooted at a and put it in $\mathcal{T}_i'$;
      construct a shortest path tree of G' rooted at b and put it in $\mathcal{T}_i''$;
      put all the connected components of $G' \setminus N_{G'}[a, b]$ into $\mathcal{G}_{i+1}$;
  set $\mathcal{T} := \mathcal{T} \cup \mathcal{T}_i' \cup \mathcal{T}_i''$ and $\mathcal{F} := \mathcal{F} \cup \mathcal{F}_i$;
  set $i := i + 1$;
return $\mathcal{T}$, $\mathcal{F}$, and $\mathcal{T}_i', \mathcal{T}_i''$ and $\mathcal{F}_i$ for each i.

The following lemma follows from Procedure 1.

Lemma 4. *For any two vertices $x, y \in V(G)$, there exists a local shortest path tree $T \in \mathcal{T}$ such that $d_T(x, y) \leq d_G(x, y) + 2$ or an r-gon subgraph $F \in \mathcal{F}$ of G such that $d_F(x, y) = d_G(x, y)$.*

Proof. Let $\mathcal{G} := \bigcup_i \{\mathcal{G}_i\}$ be the family of all subgraphs of G generated by Procedure 1. Let G' be the smallest (by the number of vertices in it) subgraph from $\mathcal{G}$, containing both vertices x and y together with a shortest path of G connecting them. Denote this shortest path by $P^G(x, y)$. By the choice of G', we know that $d_{G'}(x, y) = d_G(x, y)$. If G' belongs to $\mathcal{F}$, then it is an r-gon graph and therefore we are done. If G' does not belong to $\mathcal{F}$, then any subgraph $G'' \in \mathcal{G}$, properly contained in G' either does not contain both vertices x and y or $d_{G''}(x, y) > d_G(x, y)$.

Consider special vertices a and b of a k'-gon graph G' (see Lemma 3) and let $S := N_{G'}[a, b]$ and $T', T'' \in \mathcal{T}$ be the two shortest path trees of G' rooted at a and b, respectively. From the choice of G', we have $P^G(x, y) \cap S \neq \emptyset$. Let w be a vertex from $P^G(x, y) \cap S$ and assume, without loss of generality, that w belongs to $N_{G'}[a]$. Since T' is a shortest path tree of G' rooted at a, we have $d_{T'}(x, a) = d_{G'}(x, a) \leq d_{G'}(x, w) + 1$ and $d_{T'}(y, a) = d_{G'}(y, a) \leq d_{G'}(y, w) + 1$. Combining these inequalities with $d_{G'}(x, y) = d_{G'}(x, w) + d_{G'}(y, w)$ and $d_{T'}(x, y) \leq d_{T'}(x, a) + d_{T'}(y, a)$, we obtain $d_{T'}(x, y) \leq d_{G'}(x, y) + 2$. Since $d_G(x, y) = d_{G'}(x, y)$, we conclude $d_{T'}(x, y) \leq d_G(x, y) + 2$. □

Now we are ready to show how to construct a system of at most $2 \log_{3/2} k + 7$ collective additive tree 2-spanners for a k-gon graph G. Let $\mathcal{G}_i := \{G_i^1, G_i^2, \ldots, G_i^{p_i}\}$ be the connected graphs of the ith iteration of the while loop ($i = 0, 1, 2, \ldots$). We run Procedure 1 with parameter $r = 2$. Since, in the start iteration, a k-gon graph G is reduced to a set of k_1-gon graphs with $k_1 \leq 2\lceil \frac{1}{3}k \rceil \leq \frac{2}{3}k + \frac{4}{3}$, and generally, at iteration $i - 1$, any k_{i-1}-gon graph is reduced to a set of k_i-gon graphs with $k_i \leq \frac{2}{3}k_{i-1} + \frac{4}{3}$, we conclude that all graphs of $\mathcal{G}_i$ are k_i-gon graphs for $k_i \leq (\frac{2}{3})^i k + 2((\frac{2}{3})^i + (\frac{2}{3})^{i-1} + \cdots + \frac{2}{3}) = (\frac{2}{3})^i k + 4(1 - (\frac{2}{3})^i)$. Hence, after at most $\log_{2/3} k + 1$ iterations, the input k-gon graph G will be reduced to a set of k'-gon graphs with $k' \leq 4$, and all graphs at the beginning of iteration $\log_{2/3} k + 4$ will be 2-gon graphs (i.e., permutation graphs).

We use T'^j_i, T''^j_i to denote the two local shortest path trees constructed for a graph $G^j_i \notin \mathcal{F}_i$, $1 \leq j \leq p_i$, by Procedure 1. For a permutation graph $G^j_i \in \mathcal{F}_i$, let T^j_i be an additive tree 2-spanner of G^j_i, which exists by Lemma 1, and let $T'^j_i := T''^j_i := T^j_i$. Clearly, for any $j, j' \in \{1, \cdots, p_i\}$, $j \neq j'$, we have $G^j_i \cap G^{j'}_i = \emptyset$. Therefore, $T'^j_i \cap T'^{j'}_i = \emptyset$ and $T''^j_i \cap T''^{j'}_i = \emptyset$ hold. We can extend in linear $O(|E|)$ time the forest $T'^1_i, T'^2_i, \ldots, T'^{p_i}_i$ of G to a single spanning tree T'_i of G (using, for example, a variant of the Kruskal's Spanning Tree Algorithm). Similarly, we can extend the forest $T''^1_i, T''^2_i, \ldots, T''^{p_i}_i$ to another single spanning tree T''_i of G. We call these trees T'_i, T''_i the *spanning trees of G corresponding to the ith iteration of the while loop.* For the last iteration, it is sufficient to consider only one spanning tree T_{last} (an extension of the forest $T^1_{last}, T^2_{last}, \ldots, T^{p_{last}}_{last}$). Since the while loop has at most $\log_{3/2} k + 4$ iterations, in this way we will construct at most $2 \log_{3/2} k + 7$ spanning trees for G, two for each iteration of the while loop, except for the last iteration, where we have only one spanning tree. Denote the collection of these spanning trees by $\mathcal{ST}(G)$. By Lemma 4, it is rather straightforward to show that for any two vertices x and y of G, there exists a spanning tree $T \in \mathcal{ST}(G)$ such that $d_T(x, y) \leq d_G(x, y) + 2$. Thus, we have

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 have

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.*

We can state also the following result.

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 $0 \leq \ell \leq \log_{3/2} k + 3$. Moreover, such a sparse spanner is constructable in $O(m \log k)$ time.*

Proof. We run Procedure 1 with parameter $r := k_\ell = (\frac{2}{3})^\ell k + 4(1 - (\frac{2}{3})^\ell)$. In this case, we will have only $\ell + 1$ iterations of the while loop of Procedure 1. We use also the fact that in any k'-gon graph the length of a largest induced cycle is at most $2k'$ (see [14]). In [8], it was shown that if the length of largest induced cycle of a graph G' is c, then G' admits an additive $(c+1)$-spanner with at most $2|V(G')| - 2$ edges, and such a sparse spanner for G' can be constructed in $O(|E(G')|)$ time. Using this, for each k_ℓ-gon graph $G^j_i \in \mathcal{F}$ we can construct an additive $(2k_\ell + 1)$-spanner H^j_i with at most $2|V(G^j_i)| - 2$ edges.

Now, a spanning subgraph $H = (V, F)$ of $G = (V, E)$ can be defined as follows. The edge-set F of H is empty initially. For each iteration i ($0 \leq i \leq \ell$), if $G^j_i \in \mathcal{G}_i$ belongs to $\mathcal{F}_i$, then add all edges $E(H^j_i)$ into F, else add into F all edges of local shortest path trees T'^j_i and T''^j_i. Since for each iteration we add into F at most $2n - 2$ edges of G, the final edge-set F will have no more that $(2n - 2)(\ell + 1)$ edges. Using Lemma 4, it is easy to see also that for any two vertices x and y of G, $d_H(x, y) \leq d_G(x, y) + 2k_\ell + 1$ holds. □

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.*

5 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. 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 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.

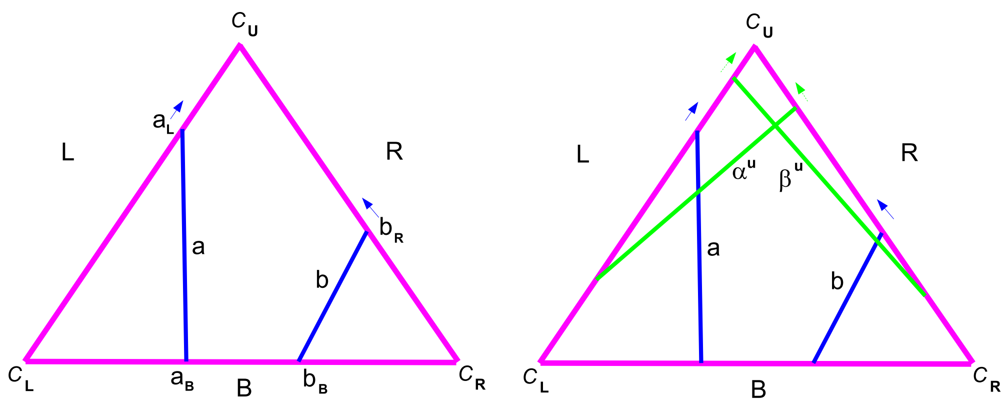

Fig. 3. A 3-gon intersection model Δ with special chords a, b, α^u and β^u.

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.

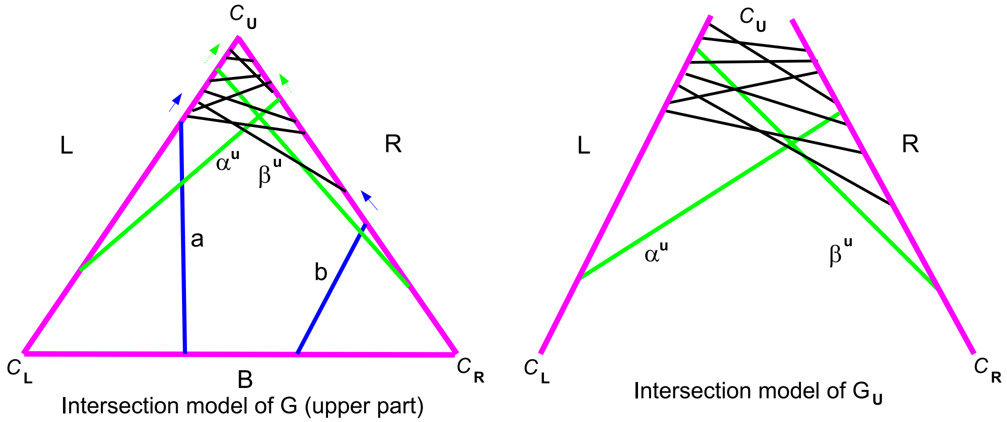

Fig. 4. Permutation graph G_U extracted from G.

The following two propositions hold for G_U.

Proposition 3. *For every $x, y \in V_U \cup \{\alpha^u, \beta^u\}$, $d_{G_U}(x, y) = d_G(x, y)$.*

Proof. Clearly if both α^u and β^u exist, then $d_G(\alpha^u, \beta^u) = d_{G_U}(\alpha^u, \beta^u) = 1$. Let $P_G(x, y)$ be a shortest path in G between $x, y \in V_U$. If $P_G(x, y)$ has no vertices outside V_U, then this path is in G_U, too, and therefore $d_{G_U}(x, y) = d_G(x, y)$. Assume now that $P_G(x, y)$ contains vertices from $V \setminus V_U$. Consider such a vertex x' closest to x and such a vertex y' closest to y. Let x'' be the neighbor of x' on $P_G(x, y)$ closer to x, and y'' be the neighbor of y' on $P_G(x, y)$ closer to y. Necessarily, $x'', y'' \in V_U$, x', y' belong to $N_G[a, b]$ and because of the maximality of a_L and b_R, the corresponding chords x', y' are between (C_L, a_L) and (C_U, b_R) or between (C_R, b_R) and (C_U, a_L). If $x' \neq y'$, then a simple geometric consideration shows that x'' must be adjacent to y' or y'' must be adjacent to x' or x'', y'' are adjacent. Since that is impossible in a shortest path $P_G(x, y)$, we conclude $x' = y'$. Assume, without loss of generality, that the chord $x' = y'$ is between (C_L, a_L) and (C_U, b_R). In $P_G(x, y)$, by replacing vertex x' with vertex α^u (note that chord α^u crosses both x'' and y''), one can obtain a shortest (x, y)-path of G completely contained in G_U.

Now let $P_G(x, \alpha^u)$ be a shortest path in G between $x \in V_U$ and α^u. If $P_G(x, \alpha^u)$ has no vertices outside $V_U \cup \{\alpha^u\}$, then this path is in G_U, too, and therefore $d_{G_U}(x, \alpha^u) = d_G(x, \alpha^u)$. Assume that $P_G(x, \alpha^u)$ contains vertices from $V \setminus (V_U \cup \{\alpha^u\})$. Consider such a vertex x' closest to x and let x'' be the neighbor of x' on $P_G(x, \alpha^u)$ closer to x. Since β^u and α^u are adjacent, $x'' \neq \beta^u$. Necessarily, $x'' \in V_U$, x' belongs to $N_G[a, b]$ and the corresponding chord x' is between (C_L, a_L) and (C_U, b_R) or between (C_R, b_R) and (C_U, a_L). A simple geometric consideration shows that x'' must be adjacent to β^u. In $P_G(x, \alpha^u)$, replacing vertex x' with vertex β^u (note that chord β^u crosses both x'' and α^u), one can obtain a shortest (x, α^u)-path of G completely contained in G_U, i.e., $d_G(x, \alpha^u) = d_{G_U}(x, \alpha^u)$. Similarly, we can show that $d_G(x, \beta^u) = d_{G_U}(x, \beta^u)$ for every $x \in V_U$. □

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 $s^u := \beta^u$.

Proposition 4. *There is a linear time constructable $BFS(s^u)$-tree T_U of G_U such that $d_G(x,y) \leq d_{T_U}(x,y) \leq d_G(x,y) + 2$ and $d_{T_U}(x,s^u) = d_G(x,s^u)$ for any x, y in $V_U \cup \{\alpha^u, \beta^u\}$.*

Proof. Since G_U is a permutation graph and s^u is extreme, by Lemma 1, there is in G_U 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 $V_U \cup \{\alpha^u, \beta^u\}$. Moreover, since T_U is a subgraph of G, $d_G(x,y) \leq d_{T_U}(x,y)$ for all $x, y \in V_U \cup \{\alpha^u, \beta^u\}$. Hence, by Proposition 3, we are done. □

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).

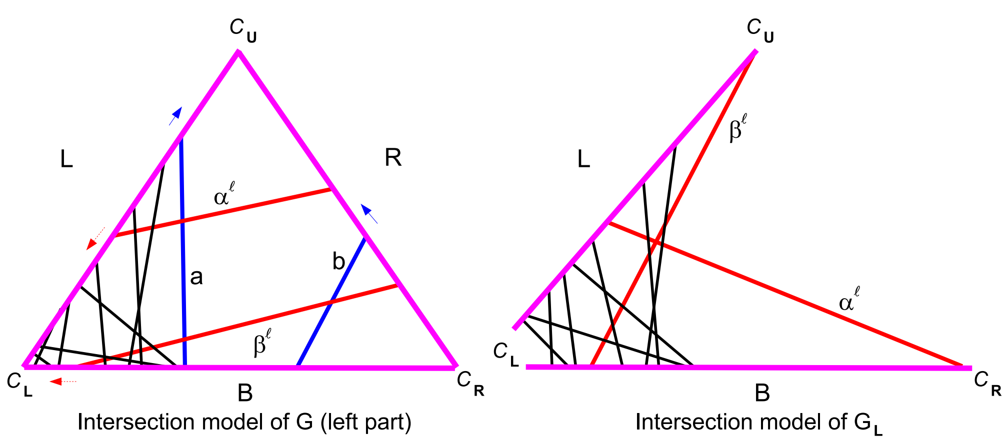

Fig. 5. Permutation graph G_L obtained from G.

The following three propositions hold for G_L.

Proposition 5. *Let both α^ℓ and β^ℓ exist. Then, there is no shortest path in G_L between any $x, y \in V_L$, which uses the edge $(\alpha^\ell, \beta^\ell)$. Moreover, for any vertex $x \in V_L$ and $s \in \{\alpha^\ell, \beta^\ell\}$, there is a shortest path $P_{G_L}(x,s)$ of G_L which does not use the edge $(\alpha^\ell, \beta^\ell)$, whenever $(N_G(s) \setminus N_G(\{\alpha^\ell, \beta^\ell\} \setminus \{s\})) \cap V_L \neq \emptyset$.*

Proof. Let $P_{G_L}(x,y)$ be a shortest path of G_L between x and y ($x,y \in V_L$) using the edge $(\alpha^\ell, \beta^\ell)$. Consider the neighbors f and t ($f,t \in V_L$) in $P_{G_L}(x,y)$ of α^ℓ and β^ℓ, respectively. Since $f \in N_G(\alpha^\ell) \setminus N_G(\beta^\ell)$, $t \in N_G(\beta^\ell) \setminus N_G(\alpha^\ell)$ and $f,t \in V_L$, a simple geometric consideration shows that f and t must be adjacent in G (and hence in G_L), thereby contradicting $P_{G_L}(x,y)$ being a shortest (x,y)-path in G_L.

Now let $P_{G_L}(x,s)$ be a shortest path of G_L between $x \in V_L$ and $s \in \{\alpha^\ell, \beta^\ell\}$ using the edge $(\alpha^\ell, \beta^\ell)$, and assume that s has a neighbor f in V_L which is not adjacent to $g := \{\alpha^\ell, \beta^\ell\} \setminus \{s\}$. Consider the neighbor $t \in V_L$ in $P_{G_L}(x,s)$ of g. Since $t \in N_G(g) \setminus N_G(s)$, $f \in N_G(s) \setminus N_G(g)$ and $f,t \in V_L$, a simple geometric consideration shows that f and t must be adjacent in G (and hence in G_L). Replacing vertex g in $P_{G_L}(x,s)$ with vertex f, we obtain a new shortest (x,s)-path in G_L, which does not use the edge $(\alpha^\ell, \beta^\ell)$. □

Note that, since G_L may have edge $(\alpha^\ell, \beta^\ell)$ which may not be an edge of G, some distances in G_L can be smaller than in G.

Proposition 6. *For every $x,y \in V_L$, $d_{G_L}(x,y) = d_G(x,y)$. Moreover, for each $s \in \{\alpha^\ell, \beta^\ell\}$, $d_{G_L}(x,s) \le d_G(x,s)$ holds for all $x \in V_L$, and if $d_{G_L}(x,s) < d_G(x,s)$ for some $x \in V_L$, then $d_{G_L}(x,s) = d_G(x,s) - 1$ and every neighbor of s in V_L is a neighbor of $g := \{\alpha^\ell, \beta^\ell\} \setminus \{s\}$.*

Proof. Let $P_G(x,y)$ be a shortest path in G between $x,y \in V_L$. If $P_G(x,y)$ has no vertices outside V_L, then this path is in G_L, too, and therefore $d_{G_L}(x,y) \le d_G(x,y)$. Hence, by Proposition 5, $d_{G_L}(x,y) = d_G(x,y)$. Assume now that $P_G(x,y)$ contains vertices from $V \setminus V_L$. Consider such a vertex x' closest to x and such a vertex y' closest to y. Let x'' be the neighbor of x' on $P_G(x,y)$ closer to x, and y'' be the neighbor of y' on $P_G(x,y)$ closer to y. Necessarily, $x'', y'' \in V_L$, x', y' belong to $N_G[a,b]$ and the corresponding chords x', y' are between $(C_L, a_B) \cap (C_L, b_B)$ and (C_R, b_R) or between (C_L, a_L) and $R \cup (a_B, C_R) \cup (b_B, C_R)$. If $x' \neq y'$, then a simple geometric consideration shows that x'' must be adjacent to y' or y'' must be adjacent to x' or x'', y'' are adjacent. Since that is impossible in a shortest path $P_G(x,y)$, we conclude $x' = y'$. If the chord $x' = y'$ is between $(C_L, a_B) \cap (C_L, b_B)$ and (C_R, b_R), then in $P_G(x,y)$ we can replace vertex x' with vertex β^ℓ (since chord β^ℓ crosses both x'' and y''). If the chord $x' = y'$ is between (C_L, a_L) and $R \cup (a_B, C_R) \cup (b_B, C_R)$, then in $P_G(x,y)$ we can replace vertex x' with vertex α^ℓ (since chord α^ℓ crosses both x'' and y''). In both cases, we obtain a shortest (x,y)-path of G completely contained in G_L. Hence, $d_{G_L}(x,y) \le d_G(x,y)$, implying $d_{G_L}(x,y) = d_G(x,y)$, by Proposition 5.

Consider now a shortest path $P_G(x,\alpha^\ell)$ in G between $x \in V_L$ and α^ℓ. If $P_G(x,\alpha^\ell)$ has no vertices outside V_L, except α^ℓ itself, then this path is in G_L, too, and therefore $d_{G_L}(x,\alpha^\ell) \le d_G(x,\alpha^\ell)$. We will have $d_{G_L}(x,\alpha^\ell) < d_G(x,\alpha^\ell)$ only if there is a path P in G_L shorter than $P_G(x,\alpha^\ell)$ where P uses the edge $(\alpha^\ell,\beta^\ell) \in E(G_L) \setminus E(G)$. But then, by Proposition 5, any neighbor of α^ℓ in V_L is a neighbor of β^ℓ, too, thereby contradicting the existence of P. Assume now that $P_G(x,\alpha^\ell)$ contains vertices from $V \setminus (V_L \cup \{\alpha^\ell\})$. Consider such a vertex x' closest to x and let x'' be the neighbor of x' on $P_G(x,\alpha^\ell)$ closer to x. Necessarily, $x'' \in V_L$, x' belongs to $N_G[a,b]$ and the corresponding chord x' is between $(C_L, a_B) \cap (C_L, b_B)$ and (C_R, b_R) or between (C_L, a_L) and $R \cup (a_B, C_R) \cup (b_B, C_R)$. A simple geometric consideration shows that x'' is either adjacent to α^ℓ (which contradicts $P_G(x,\alpha^\ell)$ being a shortest path) or to β^ℓ. Since β^ℓ is adjacent in G_L to α^ℓ as well, we get a (x,α^ℓ)-path completely contained in G_L and of length at most the length of $P_G(x,\alpha^\ell)$. Hence, $d_G(x,\alpha^\ell) \ge d_{G_L}(x,\alpha^\ell)$. Again, $d_{G_L}(x,\alpha^\ell) < d_G(x,\alpha^\ell)$ can hold only if there is no shortest path between x and α^ℓ in G_L not using the edge (α^ℓ,β^ℓ). But then, by Proposition 5, any neighbor of α^ℓ in V_L is a neighbor of β^ℓ, too. Similarly, we can show that

$d_{G_L}(x,\beta^\ell) \le d_G(x,\beta^\ell)$ holds for all $x \in V_L$, and if $d_{G_L}(x,\beta^\ell) < d_G(x,\beta^\ell)$ for some $x \in V_L$, then every neighbor of β^ℓ in V_L is a neighbor of α^ℓ.

To show that $d_{G_L}(x,\alpha^\ell) < d_G(x,\alpha^\ell)$ implies $d_{G_L}(x,\alpha^\ell) = d_G(x,\alpha^\ell)-1$, first note that vertex a is adjacent in G to both α^ℓ and β^ℓ, and that $d_G(x,\beta^\ell) = d_{G_L}(x,\beta^\ell)$, since β^ℓ is adjacent to x'' in V_L. Hence, $d_G(x,\alpha^\ell) \le d_G(x,\beta^\ell)+2 = d_{G_L}(x,\beta^\ell)+2$. On the other hand, we have $d_G(x,\alpha^\ell) \ge d_{G_L}(x,\alpha^\ell)+1 = d_{G_L}(x,\beta^\ell)+2$. From these two inequalities, $d_{G_L}(x,\alpha^\ell) = d_G(x,\alpha^\ell)-1$ follows. Similarly, one can show that $d_{G_L}(x,\beta^\ell) < d_G(x,\beta^\ell)$ implies $d_{G_L}(x,\beta^\ell) = d_G(x,\beta^\ell)-1$. □

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 $s^\ell := \beta^\ell$.

Corollary 5. *$d_{G_L}(x,s^\ell) = d_G(x,s^\ell)$ for every $x \in V_L$.*

Proof. If $s^\ell = \alpha^\ell$, i.e., there is a neighbor of α^ℓ in V_L which is not a neighbor of β^ℓ, then, by Proposition 6, $d_{G_L}(x,\alpha^\ell) = d_G(x,\alpha^\ell)$. Assume now that $s^\ell = \beta^\ell$. If there is a neighbor of β^ℓ in V_L which is not a neighbor of α^ℓ, then again, by Proposition 6, $d_{G_L}(x,\beta^\ell) = d_G(x,\beta^\ell)$. Hence, we may assume that α^ℓ and β^ℓ have the same neighborhood in V_L. In this case, $d_{G_L}(x,\alpha^\ell) = d_G(x,\alpha^\ell) = d_{G_L}(x,\beta^\ell) = d_G(x,\beta^\ell)$, since edge (α^ℓ,β^ℓ) is not part of any shortest path of G_L between $x \in V_L$ and $v \in \{\alpha^\ell,\beta^\ell\}$. □

Proposition 7. *There is a linear time constructable $BFS(s^\ell)$-tree T_L of G_L such that $d_G(x,y)-1 \le d_{T_L}(x,y) \le d_G(x,y)+2$, for any x,y in $V_L \cup \{\alpha^\ell,\beta^\ell\}$, and $d_{T_L}(x,s^\ell) = d_G(x,s^\ell)$, for all $x \in V_L$. Moreover, $d_G(x,y) \le d_{T_L}(x,y)$ for all $x,y \in V_L$.*

Proof. Since G_L is a permutation graph, by Lemma 1, there is in G_L a linear time constructable $BFS(s^\ell)$-tree T_L such that $d_{G_L}(x,y) \le d_{T_L}(x,y) \le d_{G_L}(x,y)+2$ and $d_{T_L}(x,s^\ell) = d_{G_L}(x,s^\ell)$ for any x,y in $V_L \cup \{\alpha^\ell,\beta^\ell\}$. Hence, $d_{T_L}(x,s^\ell) = d_G(x,s^\ell)$ for every $x \in V_L$, by Corollary 5, and $d_G(x,y)-1 \le d_{T_L}(x,y) \le d_G(x,y)+2$ for every $x,y \in V_L \cup \{\alpha^\ell,\beta^\ell\}$ (with $d_G(x,y) \le d_{T_L}(x,y)$ for all $x,y \in V_L$), by Proposition 6. Clearly, $d_G(\alpha^\ell,\beta^\ell) \le 2$ (since a is adjacent to both α^ℓ, β^ℓ) and $d_{T_L}(\alpha^\ell,\beta^\ell) = d_{G_L}(\alpha^\ell,\beta^\ell) = 1$. □

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 $s^r := \beta^r$. We can state.

Proposition 8. *There is a linear time constructable $BFS(s^r)$-tree T_R of G_R such that $d_G(x,y)-1 \le d_{T_R}(x,y) \le d_G(x,y)+2$, for any x,y in $V_R \cup \{\alpha^r,\beta^r\}$, and $d_{T_R}(x,s^r) = d_G(x,s^r)$, for all $x \in V_R$. Moreover, $d_G(x,y) \le d_{T_R}(x,y)$ for all $x,y \in V_R$.*

We will need also the following straightforward facts.

Proposition 9. *We have $d_G(x,s^\ell) \le d_G(x,\{\alpha^\ell,\beta^\ell\})+1$ for every $x \in V_L$, and $d_G(x,s^r) \le d_G(x,\{\alpha^r,\beta^r\})+1$ for every $x \in V_R$.*

Proof. We will prove only the first part. The proof of the second part is similar. Let $g := \{\alpha^\ell,\beta^\ell\}) \setminus \{s^\ell\}$, and assume $d_G(x,g) \le d_G(x,s^\ell)-2$ for some vertex $x \in V_L$. Consider a shortest path $P_G(x,g)$ in G between x and g and the neighbor g' of g in $P_G(x,g)$. Since a is adjacent to both α^ℓ and β^ℓ, $d_G(x,g) = d_G(x,s^\ell) - 2$ thereby implying that s^ℓ has no neighbors in $P_G(x,g)$ and a has only g as a neighbor in $P_G(x,g)$. Then, necessarily, g' belongs to V_L and, by the choice of s^ℓ, there must exist a neighbor f of s^ℓ in V_L which is not adjacent to g. A simple geometric consideration then shows that f and g' have to be adjacent in G. The latter is impossible since $d_G(x,s^\ell) = d_G(x,g) + 2$. □

Proposition 10. *We have $V = V_U \cup V_L \cup V_R \cup N_G[a,b]$ and $V_U, V_L, V_R, N_G[a,b]$ are disjoint sets. Moreover, $N_G[a,b]$ separates vertices of V_X from vertices of V_Y for every $X, Y \in \{L, R, U\}$, $X \neq Y$.*

Proposition 11. *We have $d_G(a,b) \leq 3$.*

Proof. If β^r or β^ℓ exist (say, without loss of generality, that β^r exists), then $d_G(a,b) \leq 2$, since β^r crosses both a and b. Assume now that neither β^r nor β^ℓ exists. Then both α^r and α^ℓ must exist. If $d_G(a,b) > 2$ and both α^u and β^u exist, then $(a,\alpha^u), (\alpha^u,\beta^u), (\beta^u,b) \in E(G)$ and hence $d_G(a,b) = 3$. Assume now that $d_G(a,b) > 2$ and, without loss of generality, α^u exists and β^u does not exist. Choose a BR-chord x such that x_R belongs to segment (C_R, b_R), x_B belongs to segment (a_B, b_B) and x_B is closest to C_L. Analogously, choose a BL-chord y such that y_L belongs to segment (C_L, a_L), y_B belongs to segment (a_B, b_B) and y_B is closest to C_R. Since β^u does not exist, but there must be a connection in G between vertices of V_L and vertices of V_R, the chords x and y must exist and have to cross each other. Hence, $d_G(a,b) = d_G(a,y)+d_G(y,x)+d_G(x,b) = 3$. □

Now we are ready to state the main result of this section.

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

Proof. We will create a spanning tree T of G from the trees T_U, T_L and T_R described in Proposition 4, Proposition 7 and Proposition 8 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. 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).

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, by the proof of Proposition 11, $d_G(a,b) = 3$, 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 the proof of Proposition 11), 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. It is possible that $\alpha^\ell = \alpha^u$, $\alpha^\ell = \beta^r$, $\alpha^r = \beta^u$, $\alpha^r = \beta^\ell$ and $\alpha^\ell = \alpha^r$, but our assignment of vertices $\alpha^\ell, \beta^\ell, \alpha^u, \beta^u, \alpha^r, \beta^r$ to a and b in T agrees with that since vertices of each pair are assigned to the same vertex from $\{a, b\}$. Clearly, T constructed this way is a spanning tree of G. In what follows we show that T is an additive tree 6-spanner of G.

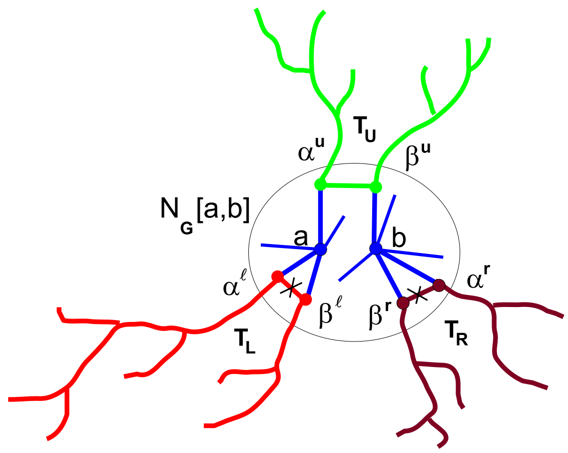

Fig. 6. Trees T_L, T_R and T_U connected via $N_G[a, b]$ to form a tree spanner of G.

Consider any vertices $x, y \in V_L$. We have $d_T(x, y) \leq d_{T_L}(x, y) + 1$, by construction of T, and $d_{T_L}(x, y) \leq d_G(x, y) + 2$, by Proposition 7. Hence, $d_T(x, y) \leq d_G(x, y) + 3$. The same inequality holds for $x, y \in V_R$.

Consider any vertices $x, y \in V_U$. We have $d_T(x, y) \leq d_{T_U}(x, y) + 3$, by construction of T, and $d_{T_U}(x, y) \leq d_G(x, y) + 2$, by Proposition 4. Hence, $d_T(x, y) \leq d_G(x, y) + 5$.

For any two vertices $x, y \in N_G(a, b)$, clearly $d_T(x, y) \leq 1 + d_T(a, b) + 1 \leq 5$ by Proposition 11 and thus $d_T(x, y) \leq d_G(x, y) + 4$.

Now consider arbitrary vertices $x \in V_L$ and $y \in V_R$. It is easy to see that $d_G(x, N_G[a, b]) = d_G(x, \{\alpha^\ell, \beta^\ell\})$ and, hence, $d_G(x, a) = d_G(x, \{\alpha^\ell, \beta^\ell\}) + 1$. Assuming without loss of generality, that α^ℓ and β^ℓ are attached in T to a we know, by the construction of T, that $d_T(x, a)$ is equal either to $d_{T_L}(x, s^\ell)$ or to $d_{T_L}(x, s^\ell) + 1$. Hence, by Propositions 7 and 9, $d_T(x, a) \leq d_{T_L}(x, s^\ell) + 1 = d_G(x, s^\ell) + 1 \leq d_G(x, \{\alpha^\ell, \beta^\ell\}) + 2 = d_G(x, N_G[a, b]) + 2$. Moreover, $d_T(x, a) = d_G(x, N_G[a, b]) + 2$ only if both α^ℓ and β^ℓ exist (i.e., $d_G(a, b) \leq 2$). Similarly, assuming without loss of generality, that α^r and β^r are attached in T to b, we see that $d_T(y, b) \leq d_{T_R}(y, s^r) + 1 = d_G(y, s^r) + 1 \leq d_G(y, \{\alpha^r, \beta^r\}) + 2 = d_G(y, N_G[a, b]) + 2$. Now, $d_T(x, y) \leq d_T(x, a) + d_T(a, b) + d_T(y, b) \leq d_G(x, N_G[a, b]) + 2 + d_T(a, b) + d_G(y, N_G[a, b]) + 2 \leq d_G(x, y) + 6$, since $N_G[a, b]$ separates V_L from V_R and $d_T(a, b) \leq 2$ if $d_T(x, a) = d_G(x, N_G[a, b]) + 2$.

For vertices $x \in V_L$ and $y \in N_G[a, b]$, $d_T(x, y) \leq d_T(x, a) + d_T(a, y) \leq d_G(x, N_G[a, b]) + 2 + d_T(a, y) \leq d_G(x, y) + 5$, since $d_G(x, y) \geq d_G(x, N_G[a, b])$ and $d_T(a, y) \leq 3$ when $d_T(x, a) = d_G(x, N_G[a, b]) + 2$ (i.e., when both α^ℓ and β^ℓ exist and hence $d_G(a, b) \leq 2$ holds). Similarly, for vertices $x \in V_R$ and $y \in N_G[a, b]$, we have $d_T(x, y) \leq d_G(x, y) + 5$.

Finally, consider arbitrary vertices $x \in V_L$ and $y \in V_U$ (the case when $x \in V_R$ and $y \in V_U$ is similar). We know that $d_T(x, a) \leq d_G(x, N_G[a, b]) + 2$. Let w be a vertex from $\{\alpha^u, \beta^u\}$ such that $d_{T_U}(y, w) = d_{T_U}(y, \{\alpha^u, \beta^u\})$. We have, $d_T(y, w) = d_{T_U}(y, w) \leq d_{T_U}(y, s^u) = d_G(y, s^u)$, by Proposition 4. Since vertices α^u and β^u, if both exist, are adjacent in G, we also have

$d_G(y, s^u) \leq d_G(y, \{\alpha^u, \beta^u\}) + 1 = d_G(y, N_G[x, y]) + 1$. Now, $d_T(x, y) \leq d_T(x, a) + d_T(a, w) + d_T(y, w) \leq d_G(x, N_G[a, b]) + 2 + d_T(a, w) + d_G(y, N_G[a, b]) + 1 \leq d_G(x, y) + 6$, since $N_G[a, b]$ separates V_L from V_U and $d_T(a, w) \leq 3$.

For vertices $x \in N_G[a, b]$ and $y \in V_U$, using w as defined above, we see that $d_T(x, y) \leq d_T(y, w) + d_T(w, x) \leq d_G(y, N_G[a, b]) + 1 + d_T(w, x) \leq d_G(x, y) + 5$, since $d_G(x, y) \geq d_G(y, N_G[a, b])$ and $d_T(w, x) \leq 4$. □

References

1. Y. Bartal, Probabilistic approximations of metric spaces and its algorithmic applications, *In IEEE Symposium on Foundations of Computer Science*, pp. 184–193, 1996.
2. Y. Bartal, On approximating arbitrary metrices by tree metrics, Proceedings of the *13th Annual ACM Symposium on Theory of Computing,* pp. 161-168, 1998.
3. J.-P. Barthélemy and A. Guénoche, Trees and Proximity Representations, *Wiley,* New York, 1991.
4. S. Bhatt, F. Chung, F. Leighton, and A. Rosenberg, Optimal simulations of tree machines, *In 27th IEEE Foundations of Computer Science,* Toronto, 1986, 274–282.
5. M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin, Approximating a Finite Metric by a Small Number of Tree Metrics, Proceedings of the *39th Annual Symposium on Foundations of Computer Science*, pp. 379–388, 1998.
6. L. Cai and D.G. Corneil, Tree spanners, *SIAM J. Discrete. Math.* 8 (1995) 359–387.
7. D.G. Corneil, F.F. Dragan, E. Köhler and Y. Xiang, Lower Bounds for Collective Additive Tree Spanners, *In preparation.*
8. V.D. Chepoi, F.F. Dragan and C. Yan, Additive Sparse Spanners for Graphs with Bounded Length of Largest Induced Cycle, *Theoretical Computer Science* 347 (2005), 54–75.
9. L.P. Chew, There are planar graphs almost as good as the complete graph, *J. of Computer and System Sciences,* 39 (1989), 205–219.
10. D.G. Corneil, F.F. Dragan, E. Köhler and C. Yan, Collective tree 1-spanners for interval graphs, In *Proceedings of 31st International Workshop "Graph-Theoretic Concepts in Computer Science" (WG '05)*, June 23-25, 2005, Metz, France, Springer, *Lecture Notes in Computer Science* 3787, pp. 151–162.
11. F.F. Dragan and C. Yan, Collective Tree Spanners in Graphs with Bounded Genus, Chordality, Tree-width, or Clique-width, *In Proceedings of the 16th Annual International Symposium on Algorithms and Computation (ISAAC 2005), Lecture Notes in Computer Science 3827*, 2005, Hainan, China, pp. 583–592.
12. F.F. Dragan, C. Yan and D.G. Corneil, Collective Tree Spanners and Routing in AT-free Related Graphs, *Journal of Graph Algorithms and Applications*, 10 (2006), 97–122.
13. F.F. Dragan, C. Yan and I. Lomonosov, Collective tree spanners of graphs, *SIAM J. Discrete Math*, 20 (2006), 241–260.
14. E.S. Elmallah and L. Stewart, Polygon Graph Recognition, *Journal of Algorithms*, 26 (1998) 101–140.
15. J. Fakcharoenphol, S. Rao, and K. Talwar, A tight bound on approximating arbitrary metrics by tree metrics, Proceedings of the *35th ACM Symposium on Theory of Computing*, pp. 448 - 455, 2003.
16. M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Second Edition, Elsevier, 2004.
17. 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.
18. A.L. Liestman and T. Shermer, Additive graph spanners, *Networks,* 23 (1993), 343-364.
19. D. Peleg, Distributed Computing: A Locality-Sensitive Approach, *SIAM Monographs on Discrete Math. Appl.*, SIAM, Philadelphia, 2000.
20. D. Peleg, Proximity-Preserving Labeling Schemes and Their Applications, *J. of Graph Theory*, 33 (2000), 167–176.
21. D. Peleg and A.A. Schäffer, Graph Spanners, *J. Graph Theory*, 13 (1989), 99-116.
22. 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.
23. E. Prisner, Distance approximating spanning trees, *In Proc. STACS'97*, *Lecture Notes in Computer Science* 1200, Springer, Berlin, 1997, 499–510.
24. P.H.A. Sneath and R.R. Sokal, Numerical Taxonomy, *W.H. Freeman,* San Francisco, California, 1973.
25. D.L. Swofford and G.J. Olsen, Phylogeny reconstruction, *In Molecular Systematics (D.M. Hillis and C. Moritz, editors),* Sinauer Associates Inc., Sunderland, MA., 1990, 411–501.
26. M. Thorup and U. Zwick, Compact routing schemes, *In Proc. 13th Ann. ACM Symp. on Par. Alg. and Arch. (SPAA 2001)*, ACM 2001, pp. 1–10.