How to Use Spanning Trees to Navigate in Graphs
(Extended Abstract)

Feodor F. Dragan and Yang Xiang

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

Abstract. In this paper, we investigate three strategies of how to use a spanning tree T of a graph G to navigate in G, i.e., to move from a current vertex x towards a destination vertex y via a path that is close to optimal. In each strategy, each vertex v has full knowledge of its neighborhood $N_G[v]$ in G (or, k-neighborhood $D_k(v, G)$, where k is a small integer) and uses a small piece of global information from spanning tree T (e.g., distance or ancestry information in T), available locally at v, to navigate in G. We investigate advantages and limitations of these strategies on particular families of graphs such as graphs with locally connected spanning trees, graphs with bounded length of largest induced cycle, graphs with bounded tree-length, graphs with bounded hyperbolicity. For most of these families of graphs, the ancestry information from a BFS-tree guarantees short enough routing paths. In many cases, the obtained results are optimal up to a constant factor.

1 Introduction

As part of the recent surge of interest in different kind of networks, there has been active research exploring strategies for navigating synthetic and real-world networks (modeled usually as graphs). These strategies specify some rules to be used to advance in a graph from a given vertex towards a target vertex along a path that is close to shortest. Current strategies include (but not limited to): routing with full-tables, interval routing, routing labeling schemes, greedy routing, geographic routing, compass routing, etc. in wired or wireless communication networks and in transportation networks (see [13,15,20,26,31] and papers cited therein); routing through common membership in groups, popularity, and geographic proximity in social networks and e-mail networks (see [9,20,23] and papers cited therein).

Navigation in communication networks is performed using a routing scheme, i.e., a mechanism that can deliver packets of information from any vertex of a network to any other vertex. In most strategies, each vertex v of a graph has full knowledge of its neighborhood and uses a piece of global information

R. Královič and D. Niwiński (Eds.): MFCS 2009, LNCS 5734, pp. 282–294, 2009.

available to it about the graph topology – some "sense of direction" to each destination, stored locally at v. Based only on this information and the address of a destination, vertex v needs to decide whether the packet has reached its destination, and if not, to which neighbor of v to forward the packet.

One of the most popular strategies in wireless (and social) networks is the *geographic routing* (sometimes called also the *greedy geographic routing*), where each vertex forwards the packet to the neighbor geographically closest to the destination (see survey [15] and paper [23]). Each vertex of the network knows its position (e.g., Euclidean coordinates) in the underlying physical space and forwards messages according to the coordinates of the destination and the coordinates of neighbors. Although this greedy method is effective in many cases, packets may get routed to where no neighbor is closer to the destination than the current vertex. Many recovery schemes have been proposed to route around such voids for guaranteed packet delivery as long as a path exists [1,19,22]. These techniques typically exploit planar subgraphs (e.g., Gabriel graph, Relative Neighborhood graph), and packets traverse faces on such graphs using the well-known right-hand rule.

All earlier papers assumed that vertices are aware of their physical location, an assumption which is often violated in practice for various of reasons (see [8,21,27]). In addition, implementations of recovery schemes are either based on non-rigorous heuristics or on complicated planarization procedures. To overcome these shortcomings, recent papers [8,21,27] propose routing algorithms which assign virtual coordinates to vertices in a metric space X and forward messages using geographic routing in X. In [27], the metric space is the Euclidean plane, and virtual coordinates are assigned using a distributed version of Tutte's "rubber band" algorithm for finding convex embeddings of graphs. In [8], the graph is embedded in R^d for some value of d much smaller than the network size, by identifying d beacon vertices and representing each vertex by the vector of distances to those beacons. The distance function on R^d used in [8] is a modification of the ℓ_1 norm. Both [8] and [27] provide substantial experimental support for the efficacy of their proposed embedding techniques – both algorithms are successful in finding a route from the source to the destination more than 95% of the time – but neither of them has a provable guarantee. Unlike embeddings of [8] and [27], the embedding of [21] guarantees that the geographic routing will always be successful in finding a route to the destination, if such a route exists. Algorithm of [21] assigns to each vertex of the network a virtual coordinate in the hyperbolic plane, and performs greedy geographic routing with respect to these virtual coordinates. More precisely, [21] gets virtual coordinates for vertices of a graph G by embedding in the hyperbolic plane a spanning tree of G. The proof that this method guaranties delivery is relied only on the fact that the hyperbolic greedy route is no longer than the spanning tree route between two vertices; even more, it could be much shorter as greedy routes take enough short cuts (edges which are not in the spanning tree) to achieve significant saving in stretch. However, although the experimental results of [21] confirm that the greedy hyperbolic embedding yields routes with low stretch when applied to typical unit-disk graphs, the worst-case stretch is still linear in the network size.

Previous work. Motivated by the work of Robert Kleinberg [21], in paper [6], we initiated exploration of the following strategy in advancing in a graph from a source vertex towards a target vertex. Let $G = (V, E)$ be a (unweighted) graph and T be a spanning tree of G. To route/move in G from a vertex x towards a target vertex y, use the following rule:

TDGR(**T**ree **D**istance **G**reedy **R**outing) strategy: *from a current vertex z (initially $z = x$), unless $z = y$, go to a neighbor of z in G that is closest to y in T.*

In this strategy, each vertex has full knowledge of its neighborhood in G and can use the distances in T to navigate in G. Thus, additionally to standard local information (the neighborhood $N_G(v)$), the only global information that is available to each vertex v is the topology of the spanning tree T. In fact, v can know only a very small piece of information about T and still be able to infer from it the necessary tree-distances. It is known [14,24,25] that the vertices of an n-vertex tree T can be labeled in $O(n \log n)$ total time with labels of up to $O(\log^2 n)$ bits such that given the labels of two vertices v, u of T, it is possible to compute in constant time the distance $d_T(v, u)$, by merely inspecting the labels of u and v. Hence, one may assume that each vertex v of G knows, additionally to its neighborhood in G, only its $O(\log^2 n)$ bit distance label. This distance label can be viewed as a virtual coordinate of v.

For each source vertex x and target vertex y, by this routing strategy, a path, called a *greedy routing path*, is produced (clearly, this routing strategy will always be successful in finding a route to the destination). Denote by $g_{G,T}(x, y)$ the length (i.e., the number of edges) of a longest greedy routing path that can be produced for x and y using this strategy and T. We say that a spanning tree T of a graph G is an *additive r-carcass* for G if $g_{G,T}(x, y) \leq d_G(x, y) + r$ for each ordered pair $x, y \in V$ (in a similar way one can define also a *multiplicative t-carcass* of G, where $g_{G,T}(x, y)/d_G(x, y) \leq t$). Note that this notion differs from the notion of "remote-spanners" introduced recently in [18].

In [6], we investigated the problem, given a graph family $\mathcal{F}$, whether a small integer r exists such that any graph $G \in \mathcal{F}$ admits an additive r-carcass. We showed that rectilinear $p \times q$ grids, hypercubes, distance-hereditary graphs, dually chordal graphs (and, therefore, strongly chordal graphs and interval graphs), all admit additive 0-carcasses. Furthermore, every chordal graph G admits an additive $(\omega(G) + 1)$-carcass (where $\omega(G)$ is the size of a maximum clique of G), each chordal bipartite graph admits an additive 4-carcass. In particular, any k-tree admits an additive $(k + 2)$-carcass. All those carcasses were easy to construct.

This new combinatorial structure, carcass, turned out to be "more attainable" than the well-known structure, tree spanner (a spanning tree T of a graph G is an additive tree r-spanner if for any two vertices x, y of G, $d_T(x, y) \leq d_G(x, y)+r$ holds, and is a multiplicative tree t-spanner if for any two vertices x, y, $d_T(x, y) \leq t\ d_G(x, y)$ holds). It is easy to see that any additive (multiplicative) tree r-spanner is an additive (resp., multiplicative) r-carcass. On the other hand, there is a number of graph families not admitting any tree spanners, yet admitting very good carcasses. For example, any hypercube has an additive 0-carcass (see [6]) but does not have any tree r-spanner (additive or multiplicative) for any constant r. The same holds for 2-trees and chordal bipartite graphs [6].

Results of this paper. All graphs occurring in this paper are connected, finite, undirected, unweighted, 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 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. The *neighborhood* of a vertex v of G is the set $N_G(v) = \{u \in V : uv \in E\}$ and the *closed neighborhood* of v is $N_G[v] = N_G(v) \cup \{v\}$. The *disk* of radius k centered at v is the set of all vertices at distance at most k to v, i.e., $D_k(v, G) = \{u \in V : d_G(u, v) \leq k\}$.

In this paper we continue investigations of how to use spanning trees to navigate in graphs. Spanning trees are very well understood structures in graphs. There are many results available in literature on how to construct (and maintain) different spanning trees in a number of settings; including in distributed way, in self stabilizing and localized way, etc. (see [10,11,12,17] and papers cited therein).

Additionally to TDGR strategy, we propose to investigate two more strategies. Let $G = (V, E)$ be a graph and T be a spanning tree of G rooted at an arbitrary vertex s. Using T, we associate an interval I_v with each vertex v such that, for any two vertices u and v, $I_u \subseteq I_v$ if and only if u is a descendent of v in T. This can be done in the following way (see [29] and Fig. 1). By depth-first search tour of T, starting at root, assign each vertex u of T a depth-first search number $DFS(u)$. Then, label u by interval $[DFS(u), DFS(w)]$, where w is last descendent of u visited by depth-first search. For two intervals $I_a = [a_L, a_R]$ and $I_b = [b_L, b_R]$, $I_a \subseteq I_b$ if and only if $a_L \geq b_L$ and $a_R \leq b_R$. Let xTy denote the (unique) path of T connecting vertices x and y, and let $N_G[xTy] = \{v \in V : v$ belongs to xTy or is adjacent to a vertex of xTy in $G\}$.

IGR (**I**nterval **G**reedy **R**outing) strategy.

To advance in G from a vertex x towards a target vertex y $(y \neq x)$, do:
if there is a neighbor w of x in G such that $y \in I_w$ (i.e., $w \in sTy$),
then go to such a neighbor with smallest (by inclusion) interval;
else (which means $x \notin N_G[sTy]$), go to a neighbor w of x in G
such that $x \in I_w$ and I_w is largest such interval.

IGRF (**I**nterval **G**reedy **R**outing with forwarding to **F**ather) strategy.

To advance in G from a vertex x towards a target vertex y $(y \neq x)$, do:
if there is a neighbor w of x in G such that $y \in I_w$ (i.e., $w \in sTy$),
then go to such a neighbor with smallest (by inclusion) interval;
else (which means $x \notin N_G[sTy]$), go to the father of x in T (i.e., a neighbor
of x in G interval of which contains x and is smallest by inclusion).

Note that both, IGR and IGRF, strategies are simpler and more compact than the TDGR strategy. In IGR and IGRF, each vertex v, additionally to standard local information (the neighborhood $N_G(v)$), needs to know only $2\lceil \log_2 n \rceil$ bits of global information from the topology of T, namely, its interval I_v. Information

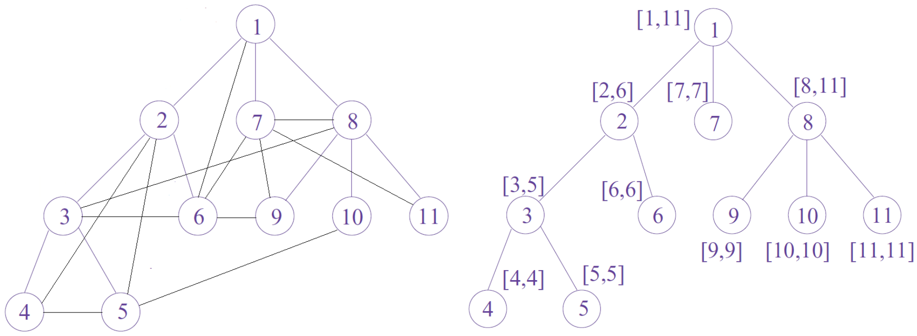

Fig. 1. A graph and its rooted spanning tree with precomputed ancestry intervals. For (ordered) pair of vertices 10 and 4, both IGR and IGRF produce path 10,8,3,4 (TDGR produces 10,5,4). For pair 5 and 8, IGR produces path 5,2,1,8, while IGRF produces path 5,3,8 (TDGR produces 5,10,8). For pair 5 and 7, IGR produces path 5,2,1,7, while IGRF produces path 5,3,2,1,7 (TDGR produces 5,2,1,7).

stored in intervals gives a "sense of direction" in navigation in G (current vertex x either may already know intervals of its neighbors, or it can ask each neighbor w, when needed, whether its interval I_w contains destination y or vertex x itself, and if yes to send I_w to x). On the other hand, as we will show in this paper, routing paths produced by IGR (IGRF) will have, in many cases, almost the same quality as routing paths produced by TDGR. Moreover, in some cases, they will be even shorter than routing paths produced by TDGR.

Let $R_{G,T}(x,y)$ be the routing path produced by IGR strategy (resp., by IGRF strategy) for a source vertex x and a target vertex y in G using T. It will be evident later that this path always exists, i.e., IGR strategy (resp., IGRF strategy) guarantees delivery. Moreover, this path is unique for each ordered pair x, y of vertices (note that, depending on tie breaking rule, TDGR can produce different routing paths for the same ordered pair of vertices). Denote by $g_{G,T}(x,y)$ the length (i.e., the number of edges) of path $R_{G,T}(x,y)$. We say that a spanning tree T of a graph G is an *additive r-frame* (resp., an *additive r-fframe*) for G if the length $g_{G,T}(x,y)$ of the routing path $R_{G,T}(x,y)$ produced by IGR strategy (resp., by IGRF strategy) is at most $d_G(x,y)+r$ for each ordered pair $x,y \in V$. In a similar way one can define also a *multiplicative t-frame* (resp., a *multiplicative t-fframe*) of G, where $g_{G,T}(x,y)/d_G(x,y) \leq t$.

In Sections 2 and 3, we show that each distance-hereditary graph admits an additive 0-frame (0-fframe) and each dually chordal graph (and, hence, each interval graph, each strongly chordal graph) admits an additive 0-frame. In Section 4, we show that each k-chordal graph admits an additive $(k-1)$-frame ($(k-1)$-fframe), each chordal graph (and, hence, each k-tree) admits an additive 1-frame (1-fframe), each AT-free graph admits an additive 2-frame (2-fframe), each chordal bipartite graph admits an additive 0-frame (0-fframe). Definitions of the graph families will be given in appropriate sections (see also [3] for many equivalent definitions of those families of graphs).

To better understand full potentials and limitations of the proposed routing strategies, in Section 5, we investigate also the following generalizations of them. Let G be a (unweighted) graph and T be a (rooted) spanning tree of G.

k-localized TDGR strategy.

To advance in G from a vertex x towards a target vertex y, go, using
a shortest path in G, to a vertex $w \in D_k(x, G)$ that is closest to y in T.

In this strategy, each vertex has full knowledge of its disk $D_k(v, G)$ (e.g., all vertices in $D_k(v, G)$ and how to reach each of them via some shortest path of G) and can use the distances in T to navigate in G. Let $g_{G,T}(x, y)$ be the length of a longest path of G that can be produced for x and y using this strategy and T. We say that a spanning tree T of a graph G is a *k-localized additive r-carcass* for G if $g_{G,T}(x, y) \leq d_G(x, y) + r$ for each ordered pair $x, y \in V$ (in a similar way one can define also a *k-localized multiplicative t-carcass* of G.

k-localized IGR strategy.

To advance in G from a vertex x towards a target vertex y, do:
if there is a vertex $w \in D_k(x, G)$ such that $y \in I_w$ (i.e., $w \in sTy$),
then go, using a shortest path in G, to such a vertex w
with smallest (by inclusion) interval;
else (which means $d_G(x, sTy) > k$),
go, using a shortest path in G, to a vertex $w \in D_k(x, G)$ such
that $x \in I_w$ and I_w is largest such interval.

k-localized IGRF strategy.

To advance in G from a vertex x towards a target vertex y, do:
if there is a vertex $w \in D_k(x, G)$ such that $y \in I_w$ (i.e., $w \in sTy$),
then go, using a shortest path in G, to such a vertex w
with smallest (by inclusion) interval;
else (which means $d_G(x, sTy) > k$), go to the father of x in T.

In these strategies, each vertex has full knowledge of its disk $D_k(v, G)$ (e.g., all vertices in $D_k(v, G)$ and how to reach each of them via some shortest path of G) and can use the DFS intervals I_w to navigate in G. We say that a (rooted) spanning tree T of a graph G is a *k-localized additive r-frame* (resp., a *k-localized additive r-fframe*) for G if the length $g_{G,T}(x, y)$ of the routing path produced by k-localized IGR strategy (resp., k-localized IGRF strategy) is at most $d_G(x, y)+r$ for each ordered pair $x, y \in V$. In a similar way one can define also a *k-localized multiplicative t-frame* (resp., a *k-localized multiplicative t-fframe*) of G.

We show, in Section 5, that any tree-length λ graph admits a λ-localized additive 5λ-fframe (which is also a λ-localized additive 5λ-frame) and any δ-hyperbolic graph admits a 4δ-localized additive 8δ-fframe (which is also a 4δ-localized additive 8δ-frame). Definitions of these graph families will also be given in appropriate sections. Additionally, we show that: for any $\lambda \geq 3$, there exists a tree-length λ graph G with n vertices for which no $(\lambda - 2)$-localized additive $\frac{1}{2}\sqrt{\log \frac{n-1}{\lambda}}$-fframe exists; for any $\lambda \geq 4$, there exists a tree-length λ graph G with n vertices for which no $\lfloor 2(\lambda - 2)/3 \rfloor$-localized additive $\frac{2}{3}\sqrt{\log \frac{3(n-1)}{4\lambda}}$-frame exists; for any $\lambda \geq 6$, there exists a tree-length λ graph G with n vertices for which no $\lfloor(\lambda-2)/4\rfloor$-localized additive $\frac{3}{4}\sqrt{\log\frac{n-1}{\lambda}}$-carcass exists.

Proofs omitted due to space limitation can be found in the journal version of the paper [7].

2 Preliminaries

Let $G=(V,E)$ be a graph and T be a spanning tree of G rooted at an arbitrary vertex s. We assume that T is given together with the precomputed ancestry intervals. The following facts are immediate from the definitions of IGR and IGRF strategies.

Lemma 1. *Any routing path $R_{G,T}(x,y)$ produced by IGR or IGRF, where x is not an ancestor of y in T, is of the form $x_1 \dots x_k y_l \dots y_1$, where $x_1 = x$, $y_1 = y$, x_i is a descendent of x_{i+1} in T, and y_i is an ancestor of y_{i-1} in T. In addition, for any $i \in [1,k]$, x_i is not an ancestor of y, and, for any $i \in [1,k-1]$, x_i is not adjacent in G to any vertex of sTy.*
If x is an ancestor of y in T, then $R_{G,T}(x,y)$ has only part $y_l \dots y_1$ with $x = y_l$, $y = y_1$ and y_i being an ancestor of y_{i-1} in T.

In what follows, any routing path produced by IGR (resp., by IGRF, by TDGR) will be called *IGR routing path* (resp., *IGRF routing path, TDGR routing path*).

Corollary 1. *A tale of any IGR routing path (any IGRF routing path) is also an IGR routing path (IGRF routing path, respectively).*

Corollary 2. *Both IGR and IGRF strategies guarantee delivery.*

Corollary 3. *Let T be a BFS-tree (Breadth-First-Search–tree) of a graph G rooted at an arbitrary vertex s, and let x and y be two vertices of G. Then, IGR and IGRF strategies produce the same routing path $R_{G,T}(x,y)$ from x to y.*

Lemma 2. *For any vertices x and y, the IGR routing path (respectively, the IGRF routing path) $R_{G,T}(x,y)$ is unique.*

Lemma 3. *Any IGR routing path $R_{G,T}(x,y)$ is an induced path of G.*

Note that an IGRF routing path $R_{G,T}(x,y) = x_1 \dots x_k y_l \dots y_1$ may not necessarily be induced in the part $x_1 \dots x_k$. In [6], it was shown that routing paths produced by TDGR strategy are also induced paths.

A graph G is called *distance-hereditary* if any induced path of G is a shortest path (see [3] for this and equivalent definitions). By Lemma 3 and Corollary 3, we conclude.

Theorem 1. *Any spanning tree of a distance-hereditary graph G is an additive 0-frame of G, regardless where it is rooted. Any BFS-tree of a distance-hereditary graph G is an additive 0-fframe of G.*

3 Frames for Dually Chordal Graphs

Let G be a graph. We say that a spanning tree T of G is *locally connected* if the closed neighborhood $N_G[v]$ of any vertex v of G induces a subtree in T (i.e., $T \cap N_G[v]$ is a connected subgraph of T). The following result was proven in [6].

Lemma 4. [6] *If T is a locally connected spanning tree of a graph G, then T is an additive 0-carcass of G.*

Here we prove the following lemma.

Lemma 5. *Let G be a graph with a locally connected spanning tree T, and let x and y be two vertices of G. Then, IGR and TDGR strategies produce the same routing path $R_{G,T}(x, y)$ from x to y (regardless where T is rooted).*

Proof. Assume that we want to route from a vertex x towards a vertex y in G, where $x \neq y$. We may assume that $d_G(x, y) \geq 2$, since otherwise both routing strategies will produce path xy. Let x^* (x') be the neighbor of x in G chosen by IGR strategy (resp., by TDGR strategy) to relay the message. We will show that $x' = x^*$ by considering two possible cases. We root the tree T at an arbitrary vertex s.

First assume that $N_G[x] \cap sTy \neq \emptyset$. By IGR strategy, we will choose a neighbor $x^* \in N_G[x]$ such that $y \in I_{x^*}$ and I_{x^*} is the smallest interval by inclusion, i.e., x^* is a vertex from $N_G[x]$ closest in sTy to y. If $d_T(x', y) < d_T(x^*, y)$, then $x' \notin sTy$ and the nearest common ancestor $NCA_T(x', y)$ of x', y in T must be in x^*Ty. Since $T \cap N_G[x]$ is a connected subgraph of T and $x', x^* \in N_G[x]$, we conclude that $NCA_T(x', y)$ must be in $N_G[x]$, too. Thus, we must have $x' = NCA_T(x', y) = x^*$.

Assume now that $N_G[x] \cap sTy = \emptyset$. By IGR strategy, we will choose a neighbor $x^* \in N_G[x]$ such that $x \in I_{x^*}$ and I_{x^*} is the largest interval by inclusion, i.e., x^* is a vertex from $N_G[x]$ closest in sTx to $NCA_T(x, y)$. Consider the nearest common ancestor $NCA_T(x', x^*)$ of x', x^* in T. Since $T \cap N_G[x]$ is a connected subgraph of T and $x', x^* \in N_G[x]$, we conclude that $NCA_T(x', x^*)$ must be in $N_G[x]$, too. Thus, necessarily, we must have $x' = NCA_T(x', x^*) = x^*$.

From these two cases we conclude, by induction, that IGR and TDGR strategies produce the same routing path $R_{G,T}(x, y)$ from x to y. □

Corollary 4. *If T is a locally connected spanning tree of a graph G, then T is an additive 0-frame of G (regardless where T is rooted).*

It has been shown in [2] that the graphs admitting locally connected spanning trees are precisely the dually chordal graphs. Furthermore, [2] showed that the class of dually chordal graphs contains such known families of graphs as strongly chordal graphs, interval graphs and others. Thus, we have the following result.

Theorem 2. *Every dually chordal graph admits an additive 0-frame. In particular, any strongly chordal graph (any interval graph) admits an additive 0-frame.*

Note that, in [2], it was shown that dually chordal graphs can be recognized in linear time, and if a graph G is dually chordal, then a locally connected spanning tree of G can be efficiently constructed.

4 Frames for k-Chordal Graphs and Subclasses

A graph G is called *k-chordal* if it has no induced cycles of size greater than k, and it is called *chordal* if it has no induced cycle of length greater than 3. Chordal graphs are precisely the 3-chordal graphs.

Theorem 3. *Let $G = (V, E)$ be a k-chordal graph. Any BFS-tree T of G is an additive $(k-1)$-fframe (and, hence, an additive $(k-1)$-frame) of G. If G is a chordal graph (i.e., $k = 3$), then any LexBFS-tree T (a special BFS-tree) of G is an additive 1-fframe (and, hence, an additive 1-frame) of G.*

A graph G is called *chordal bipartite* if it is bipartite and has no induced cycles of size greater than 4. Chordal bipartite graphs are precisely the bipartite 4-chordal graphs. A graph is called *AT–free* if it does not have an *asteroidal triple*, i.e. a set of three vertices such that there is a path between any pair of them avoiding the closed neighborhood of the third. It is known that AT-free graphs form a proper subclass of 5-chordal graphs.

Theorem 4. *Every chordal bipartite graph G admits an additive 0-fframe and an additive 0-frame, constructible in $O(n^2)$ time. Any BFS-tree T of an AT-free graph G is an additive 2-fframe (and, hence, an additive 2-frame) of G.*

5 Localized Frames for Tree-Length λ Graphs and δ-Hyperbolic Graphs

In this section, we show that any tree-length λ graph admits a λ-localized additive 5λ-fframe (which is also a λ-localized additive 5λ-frame) and any δ-hyperbolic graph admits a 4δ-localized additive 8δ-fframe (which is also a 4δ-localized additive 8δ-frame). We complement these results with few lower bounds.

Tree-length λ graphs. The *tree-length* of a graph G is the smallest integer λ for which G admits a tree-decomposition into bags of diameter at most λ. It has been introduced and extensively studied in [5]. Chordal graphs are exactly the graphs of tree-length 1, since a graph is chordal if and only if it has a tree-decomposition into cliques (cf. [3]). AT-free graphs and distance-hereditary graphs are of tree-length 2. More generally, k-chordal graphs have tree-length at most $k/2$. However, there are graphs with bounded tree-length and unbounded chordality, like the wheel (here, the *chordality* is the smallest k such that the graph is k-chordal). So, bounded tree-length graphs is a larger class than bounded chordality graphs.

We now recall the definition of *tree-decomposition* introduced by Robertson and Seymour in their work on graph minors [28]. A tree-decomposition of a graph G is a tree T whose vertices, called *bags*, are subsets of $V(G)$ such that: (1) $\cup_{X \in V(T)} X = V(G)$; (2) for all $uv \in E(G)$, there exists $X \in V(T)$ such that $u, v \in X$; and (3) for all $X, Y, Z \in V(T)$, if Y is on the path from X to Z in T then $X \cap Z \subseteq Y$. The *length* of tree-decomposition T of a graph G is $\max_{X \in V(T)} \max_{u,v \in X} d_G(u, v)$, and the *tree-length of G* is the minimum, over all tree-decompositions T of G, of the length of T.

Theorem 5. *If G has the tree-length λ, then any BFS-tree T of G is a λ-localized additive 5λ-fframe (and, hence, a λ-localized additive 5λ-frame) of G.*

Now, we provide some lower bound results.

Lemma 6. *For any $\lambda \geq 3$, there exists a tree-length λ graph without any $(\lambda-2)$-localized additive (λa)-fframe for any constant $a \geq 1$.*

Corollary 5. *For any $\lambda \geq 3$, there exists a tree-length λ graph G with n vertices for which no $(\lambda - 2)$-localized additive $\frac{1}{2}\sqrt{\log \frac{n-1}{\lambda}}$-fframe exists.*

Lemma 7. *For any $\lambda \geq 4$, there exists a tree-length λ graph without any $\lfloor 2(\lambda - 2)/3 \rfloor$-localized additive (λa)-frame for any constant $a \geq 1$.*

Corollary 6. *For any $\lambda \geq 4$, there exists a tree-length λ graph G with n vertices for which no $\lfloor 2(\lambda - 2)/3 \rfloor$-localized additive $\frac{2}{3}\sqrt{\log \frac{3(n-1)}{4\lambda}}$-frame exists.*

Lemma 8. *For any $\lambda \geq 6$, there exists a tree-length λ graph without any $\lfloor (\lambda - 2)/4 \rfloor$-localized additive (λa)-carcass for any constant $a \geq 1$.*

Corollary 7. *For any $\lambda \geq 6$, there exists a tree-length λ graph G with n vertices for which no $\lfloor (\lambda - 2)/4 \rfloor$-localized additive $\frac{3}{4}\sqrt{\log \frac{n-1}{\lambda}}$-carcass exists.*

δ-hyperbolic graphs. δ-Hyperbolic metric spaces were defined by M. Gromov [16] in 1987 via a simple 4-point condition: for any four points u, v, w, x, the two larger of the distance sums $d(u, v) + d(w, x), d(u, w) + d(v, x), d(u, x) + d(v, w)$ differ by at most 2δ. They play an important role in geometric group theory, geometry of negatively curved spaces, and have recently become of interest in several domains of computer science, including algorithms and networking. For example, (a) it has been shown empirically in [30] that the Internet topology embeds with better accuracy into a hyperbolic space than into an Euclidean space of comparable dimension, (b) every connected finite graph has an embedding in the hyperbolic plane so that the greedy routing based on the virtual coordinates obtained from this embedding is guaranteed to work (see [21]). A connected graph $G = (V, E)$ equipped with standard graph metric d_G is *δ-hyperbolic* if the metric space (V, d_G) is δ-hyperbolic. It is known (see [4]) that all graphs with tree-length λ are λ-hyperbolic, and each δ-hyperbolic graph has the tree–length $O(\delta \log n)$.

Lemma 9. *Let G be a δ-hyperbolic graph. Let s, x, y be arbitrary vertices of G and $P(s, x)$, $P(s, y)$, $P(y, x)$ be arbitrary shortest paths connecting those vertices in G. Then, for vertices $a \in P(s, x)$, $b \in P(s, y)$ with $d_G(s, a) = d_G(s, b) = \lfloor \frac{d_G(s,x)+d_G(s,y)-d_G(x,y)}{2} \rfloor$, the inequality $d_G(a, b) \leq 4\delta$ holds.*

It is clear that δ takes values from $\{0, \frac{1}{2}, 1, \frac{3}{2}, 2, \frac{5}{2}, 3, \ldots\}$, and if $\delta = 0$ then G is a tree. Hence, in what follows. we will assume that $\delta \geq \frac{1}{2}$.

Theorem 6. *If G is a δ-hyperbolic graph, then any BFS-tree T of G is a 4δ-localized additive 8δ-fframe (and, hence, a 4δ-localized additive 8δ-frame) of G.*

Proof. Let T be an arbitrary BFS-tree of G rooted at a vertex s. Let $R_{G,T}(x,y)$ be the routing path from a vertex x to a vertex y produced by 4δ-localized IGRF scheme using tree T. If x is on the T path from y to s, or y is on the T path from x to s, it is easy to see that $R_{G,T}(x,y)$ is a shortest path of G.

Let sTx (resp., sTy) be the path of T from s to x (resp., to y) and $P(y,x)$ be an arbitrary shortest path connecting vertices x and y in G. By Lemma 9, for vertices $a \in sTx$, $b \in sTy$ with $d_G(s,a) = d_G(s,b) = \lfloor \frac{d_G(s,x)+d_G(s,y)-d_G(x,y)}{2} \rfloor$, the inequality $d_G(a,b) \leq 4\delta$ holds. Furthermore, since $d_G(a,x) + d_G(a,s) = d_G(s,x)$ and $d_G(b,y) + d_G(b,s) = d_G(s,y)$, from the choice of a and b, we have $d_G(x,y) \leq d_G(a,x) + d_G(b,y) \leq d_G(x,y) + 1$.

Let x' be a vertex of xTs with $d_G(x', sTy) \leq 4\delta$ closest to x. Clearly, x' belongs to subpath aTx of path sTx. Let y' be a vertex of path yTs with $d_G(x',y') \leq 4\delta$ (i.e., $y' \in D_{4\delta}(x',G)$) closest to y. Then, according to 4δ-localized IGRF scheme, the routing path $R_{G,T}(x,y)$ coincides with $(xTx')\cup$(a shortest path of G from x' to $y')\cup(y'Ty)$. We have $length(R_{G,T}(x,y)) = d_G(x,x') + d_G(x',y') + d_G(y',y)$.

If $y' \in bTy$, then $length(R_{G,T}(x,y)) = d_G(x,x') + d_G(x',y') + d_G(y',y) \leq d_G(x,a)+4\delta+d_G(b,y) \leq d_G(x,y)+4\delta+1$. Assume now that $y' \in bTs$ and $y' \neq b$. Then, we have also $x' \neq a$. Since T is a BFS-tree of G, $d_G(y',b)$ must be at most $d_G(y',x')$ (otherwise, x' is closer than b to s in G, which is impossible). Thus, $d_G(y',b) \leq d_G(y',x') \leq 4\delta$ and, therefore, $length(R_{G,T}(x,y)) = d_G(x,x') + d_G(x',y') + d_G(y',y) \leq d_G(x,a) - 1 + 4\delta + d_G(b,y') + d_G(b,y) \leq d_G(x,y) + 1 - 1 + 8\delta = d_G(x,y) + 8\delta$.

Combining all cases, we conclude that T is a 4δ-localized additive 8δ-fframe (and a 4δ-localized additive 8δ-frame) of G. □

References

1. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. In: 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communic., pp. 48–55. ACM Press, New York (1999)
2. Brandstädt, A., Dragan, F.F., Chepoi, V.D., Voloshin, V.I.: Dually chordal graphs. SIAM J. Discrete Math. 11, 437–455 (1998)
3. Brandstädt, A., Bang Le, V., Spinrad, J.P.: Graph Classes: A Survey, Philadelphia. SIAM Monographs on Discrete Mathematics and Applications (1999)
4. Chepoi, V., Dragan, F.F., Estellon, B., Habib, M., Vaxès, Y.: Diameters, centers, and approximating trees of δ-hyperbolic geodesic spaces and graphs. In: SoCG 2008, pp. 59–68 (2008)
5. Dourisboure, Y., Gavoille, C.: Tree-decompositions with bags of small diameter. Discrete Mathematics 307, 2008–2029 (2007)
6. Dragan, F.F., Matamala, M.: Navigating in a graph by aid of its spanning tree. In: Hong, S.-H., Nagamochi, H., Fukunaga, T. (eds.) ISAAC 2008. LNCS, vol. 5369, pp. 788–799. Springer, Heidelberg (2008)

7. Dragan, F.F., Xiang, Y.: How to use spanning trees to navigate in graphs, full version, http://www.cs.kent.edu/~dragan/MFCS2009-journal.pdf
8. Fonseca, R., Ratnasamy, S., Zhao, J., Ee, C.T., Culler, D., Shenker, S., Stoica, I.: Beacon vector routing: Scalable point-to-point routing in wireless sensornets. In: 2nd USENIX/ACM Symp. on Networked Systems Design and Implementation (2005)
9. Fraigniaud, P.: Small Worlds as Navigable Augmented Networks: Model, Analysis, and Validation. In: Arge, L., Hoffmann, M., Welzl, E. (eds.) ESA 2007. LNCS, vol. 4698, pp. 2–11. Springer, Heidelberg (2007)
10. Fraigniaud, P., Korman, A., Lebhar, E.: Local MST computation with short advice. In: SPAA 2007, 154–160 (2007)
11. Garg, V.K., Agarwal, A.: Distributed maintenance of a spanning tree using labeled tree encoding. In: Cunha, J.C., Medeiros, P.D. (eds.) Euro-Par 2005. LNCS, vol. 3648, pp. 606–616. Springer, Heidelberg (2005)
12. Gartner, F.C.: A Survey of Self-Stabilizing Spanning-Tree Construction Algorithms, Technical Report IC/2003/38, Swiss Federal Institute of Technology (EPFL) (2003)
13. Gavoille, C.: Routing in distributed networks: Overview and open problems. ACM SIGACT News - Distributed Computing Column 32 (2001)
14. Gavoille, C., Peleg, D., Pérennès, S., Raz, R.: Distance labeling in graphs. J. Algorithms 53, 85–112 (2004)
15. Giordano, S., Stojmenovic, I.: Position based routing algorithms for ad hoc networks: A taxonomy. In: Ad Hoc Wireless Networking, pp. 103–136. Kluwer, Dordrecht (2004)
16. Gromov, M.: Hyperbolic Groups. In: Gersten, S.M. (ed.) Essays in group theory. MSRI Series, vol. 8, pp. 75–263 (1987)
17. Holm, J., de Lichtenberg, K., Thorup, M.: Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. J. ACM 48(4), 723–760 (2001)
18. Jacquet, P., Viennot, L.: Remote spanners: what to know beyond neighbors. In: IPDPS 2009, pp. 1–15 (2009)
19. Karp, B., Kung, H.T.: GPSR: greedy perimeter stateless routing for wireless networks. In: 6th ACM/IEEE MobiCom., pp. 243–254. ACM Press, New York (2000)
20. Kleinberg, J.M.: The small-world phenomenon: an algorithm perspective. In: STOC 2000, pp. 163–170. ACM, New York (2000)
21. Kleinberg, R.: Geographic routing using hyperbolic space. In: INFOCOM 2007, pp. 1902–1909 (2007)
22. Kuhn, F., Wattenhofer, R., Zhang, Y., Zollinger, A.: Geometric ad-hoc routing: of theory and practice. In: PODC, pp. 63–72. ACM, New York (2003)
23. Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., Tomkins, A.: Geographic routing in social networks. PNAS 102, 11623–11628 (2005)
24. Linial, N., London, E., Rabinovich, Y.: The Geometry of Graphs and Some of its Algorithmic Applications. Combinatorica 15, 215–245 (1995)
25. Peleg, D.: Proximity-Preserving Labeling Schemes and Their Applications. J. of Graph Theory 33, 167–176 (2000)
26. Peleg, D.: Distributed Computing: A Locality-Sensitive Approach. SIAM Monographs on Discrete Math. Appl. SIAM, Philadelphia (2000)
27. Rao, A., Papadimitriou, C., Shenker, S., Stoica, I.: Geographical routing without location information. In: Proceedings of MobiCom 2003, pp. 96–108 (2003)
28. Robertson, N., Seymour, P.D.: Graph minors. II. Algorithmic aspects of tree-width. Journal of Algorithms 7, 309–322 (1986)

29. Santoro, N., Khatib, R.: Labelling and Implicit Routing in Networks. The Computer Journal 28(1), 5–8 (1985)
30. Shavitt, Y., Tankel, T.: On internet embedding in hyperbolic spaces for overlay construction and distance estimation. In: INFOCOM 2004 (2004)
31. Thorup, M., Zwick, U.: Compact routing schemes. In: 13th Ann. ACM Symp. on Par. Alg. and Arch., July 2001, pp. 1–10 (2001)