Algorithmica (2013) 66:479–511
DOI 10.1007/s00453-012-9647-1

How to Use Spanning Trees to Navigate in Graphs

Feodor F. Dragan · Yang Xiang

Received: 14 June 2009 / Accepted: 31 March 2012 / Published online: 18 April 2012

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 Breadth-First-Search-tree guarantees short enough routing paths. In many cases, the obtained results are optimal up to a constant factor.

Keywords Graph algorithms · Navigating in graphs · Spanning trees · Routing in graphs · Distances · Ancestry · k-Chordal graphs · Tree-length λ graphs · δ-Hyperbolic graphs · Lower bounds

Part of these results is presented at MFCS 2009 Conference, August 24–28, 2009, Novy Smokovec, High Tatras, Slovak Republic.
Y. Xiang was supported in part by the NSF under Grant #1019343 to the Computing Research Association for the CIFellows Project.

F.F. Dragan (✉)
Algorithmic Research Laboratory, Department of Computer Science, Kent State University, Kent, OH 44242, USA
e-mail: dragan@cs.kent.edu

Y. Xiang
Department of Biomedical Informatics, The Ohio State University, Columbus, OH 43210, USA
e-mail: yxiang@bmi.osu.edu

1 Introduction

As part of the recent surge of interest in different kinds 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 (a network) from a given vertex towards a target vertex along a path that is close to shortest. Current strategies include (but not limited to): routing using 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 [24, 25, 29, 36, 42, 48] and papers cited therein); routing through common membership in groups, popularity, and geographic proximity in social networks and e-mail networks (see [2, 3, 20, 36, 39] and the literature 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 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 [29] and paper [39]). 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 [5, 35, 38]. 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 reasons (see [19, 37, 43]). In addition, implementations of recovery schemes are either based on non-rigorous heuristics or on complicated planarization procedures. To overcome these shortcomings, recent papers [19, 37, 43] propose routing algorithms which assign virtual coordinates to vertices in a metric space X and forward messages using geographic routing in X. In [43], 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 [19], 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 [19] is a modification of the ℓ_1 norm. Both [19] and [43] 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 [19] and [43], the embedding of [37] guarantees that the geographic routing will always be successful in finding a route to the destination, if such a route exists. The algorithm of [37] 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, [37] 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 guarantees delivery relies 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 [37] 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.

1.1 Previous Work

Motivated by the work of Robert Kleinberg [37], in paper [17], 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 [27, 40, 41] 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 $g_{G,T}(x, y) \le d_G(x, y) + r$ for each ordered pair $x, y \in V$ (in a similar way one can also define a *multiplicative t-carcass* of G, where $g_{G,T}(x, y)/d_G(x, y) \le t$). Note that this notion differs from the notion of "remote-spanners" introduced recently in [34].

In [17], 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 3-sun-free chordal graph admits an additive 2-carcass, 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 [17]) 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 [17].

1.2 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 the literature on how to construct (and maintain) different spanning trees in a number of settings; including in a distributed way, in a self stabilizing way, in a localized way, etc. (see [18, 21–23, 32, 33] and the literature cited therein).

Additionally to the 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 descendant of v in T. This can be done in the following way (see [46] and Fig. 1). By a depth-first search tour of T, starting at the 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 the last descendant of u visited by the 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 $G\}$.

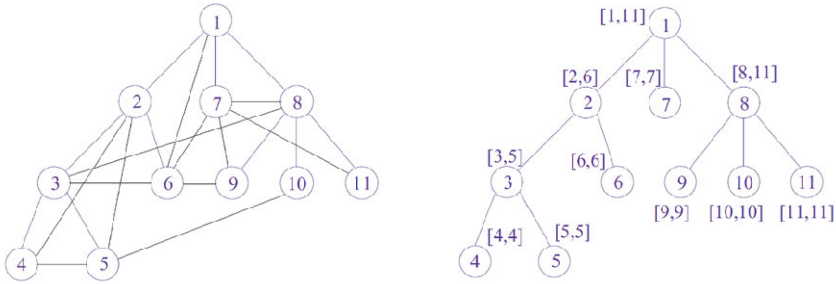


Fig. 1 A graph and its rooted spanning tree with precomputed ancestry intervals. For (ordered) pair of vertices 10 and 4, both IGR and IGRF produce path 10, 8, 3, 4 (TDGR produces 10, 5, 4). For pair 5 and 8, IGR produces path 5, 2, 1, 8, while IGRF produces path 5, 3, 8 (TDGR produces 5, 10, 8). For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

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 the 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, the interval of which contains x and is smallest by inclusion).

Note that both, the 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 stored in intervals gives a "sense of direction" in navigation in G (the 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 the strategy under consideration (namely, either the IGR strategy or the 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., both the IGR and IGRF strategies guarantee delivery). Moreover, this path is unique for each ordered pair x, y of vertices (note that, depending on the 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 the IGR strategy (resp., by the IGRF strategy) is at most $d_G(x, y) + r$ for each ordered pair $x, y \in V$. In a similar way one can also define a *multiplicative t-frame* (resp., a *multiplicative t-fframe*) of G, where $g_{G,T}(x, y)/d_G(x, y) \leq t$.

In Sects. 2 and 3, we show that each distance-hereditary graph admits an additive 0-frame (as well as an additive 0-fframe) and each dually chordal graph (and, hence, each interval graph, each strongly chordal graph) admits an additive 0-frame. In Sect. 4, we show that each k-chordal graph admits an additive $(k-1)$-frame (as well as an additive $(k-1)$-fframe), each chordal graph (and, hence, each k-tree) admits an additive 1-frame (as well as an additive 1-fframe), each AT-free graph admits an additive 2-frame (as well as an additive 2-fframe), each chordal bipartite graph admits an additive 0-frame (as well as an additive 0-fframe). Definitions of the graph families will be given in appropriate sections (see also [8] for many equivalent definitions of these families of graphs).

To better understand full potentials and limitations of the proposed routing strategies, in Sect. 5, we also investigate the following generalizations of them. Let $G = (V, E)$ 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, do:
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 also define 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.

Table 1 Upper bounds

Graph class	l-localized additive r-frame		l-localized additive r-fframe	
	l	r	l	r
Distance-hereditary	1	0	1	0
Dually chordal	1	0	–	–
Chordal bipartite	1	0	1	0
Chordal	1	1	1	1
AT-free	1	2	1	2
k-Chordal ($k \geq 3$)	1	$k-1$	1	$k-1$
Tree-length λ	λ	5λ	λ	5λ
δ-Hyperbolic	4δ	8δ	4δ	8δ

Table 2 Lower bounds for n-vertex planar tree-length λ graphs

	No l-localized additive r-fframe for every $\lambda \geq 3$	No l-localized additive r-frame for every $\lambda \geq 4$	No l-localized additive r-carcass for every $\lambda \geq 6$
l	$\lambda - 2$	$\lfloor 2(\lambda-2)/3 \rfloor$	$\lfloor (\lambda-2)/4 \rfloor$
r	$\frac{1}{2}\sqrt{\log \frac{n-1}{\lambda}}$	$\frac{2}{3}\sqrt{\log \frac{3(n-1)}{4\lambda}}$	$\frac{3}{4}\sqrt{\log \frac{n-1}{\lambda}}$

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 the k-localized IGR strategy (resp., by the 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 Sect. 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 the 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.

Our results are summarized in Tables 1 and 2.

In Appendix A, we empirically compare the performance of TDGR, IGR and IGRF, and their corresponding k-localized versions, on Unit Disk Graphs, which often model wireless ad hoc networks.

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 produced by a DFS. The following facts are immediate from the definitions of the 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 descendant 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 an *IGR routing path* (resp., *IGRF routing path, TDGR routing path*).

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

Corollary 2 *Both the IGR strategy and the IGRF strategy 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, the IGR and IGRF strategies produce the same routing path $R_{G,T}(x, y)$ from x to y. In particular, if a BFS-tree T is an additive r-fframe of G then it is also an additive r-frame of G.*

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 [17], it was shown that routing paths produced by the TDGR strategy are also induced paths.

A graph G is called *distance-hereditary* if any induced path of G is a shortest path (see [8] 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 of where it is rooted. Any BFS-tree of a distance-hereditary graph G is an additive* 0-*fframe of G.*

It has been shown in [17] that a column-wise Hamiltonian path *HP* of any Rectilinear Grid G (see Fig. 2) is an additive 0-carcass of G. Since we can number the vertices

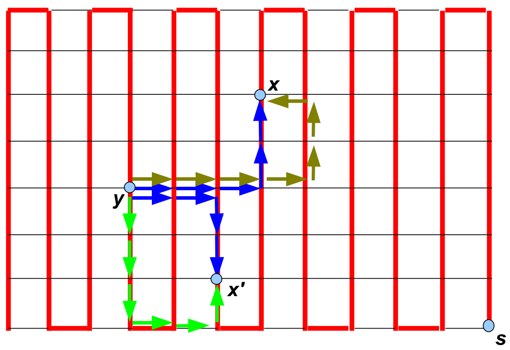


Fig. 2 Rectilinear grid and its column-wise Hamiltonian path (Color figure online)

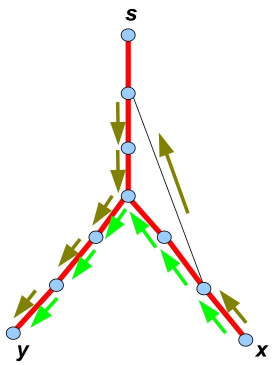


Fig. 3 A simple graph demonstrating that the IGRF strategy may produce a shorter routing path than the IGR strategy (Color figure online)

of G from 1 to n simply following the path HP and the distance in HP between a vertex with number i and a vertex with number j can be computed by formula $|i - j|$, the strategies IGR and IGRF, for Rectilinear Grids, cannot give improvements over the TDGR strategy neither in memory size per vertex nor in route stretch. Consequently, for Rectilinear Grids, the IGR and IGRF strategies are not interesting. Furthermore, one can easily see (consult Fig. 2) that HP is only an additive 2-frame for the grid G (while routing paths produced in G by IGRF, using HP, can be arbitrarily longer than shortest paths). In Fig. 2, we consider routing from y to x and from y to x'. The two blue paths are the routing paths generated by the TDGR strategy. The brown path is the routing path from y to x generated by the IGR strategy. The green path is the routing path from y to x' generated by the IGRF strategy.

In Fig. 2, one can see that IGR provides a shorter routing path than IGRF from y to x. However, in some cases IGRF can outperform IGR, too. For example, in Fig. 3, IGRF produces a shorter routing path (green) from x to y than that (brown) produced by IGR. Later, in Sect. 4.1, we will also see that there are chordal graphs admitting additive 0-fframes but not having any additive 0-carcasses or additive 0-frames.

3 Frames for Dually Chordal Graphs

In this section, we will show that each dually chordal graph admits an additive 0-frame.

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 [17].

Lemma 4 [17] *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, the IGR and TDGR strategies produce the same routing path $R_{G,T}(x, y)$ from x to y (regardless of 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 the IGR strategy (resp., by the 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 the 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 the 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 the IGR and TDGR strategies produce the same routing path $R_{G,T}(x, y)$ from x to y. □

From Lemmas 4 and 5, we immediately obtain the following corollary.

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

It has been shown in [7] that the graphs admitting locally connected spanning trees are precisely the dually chordal graphs. Furthermore, [7] 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 [6, 7], 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

In this section, we will employ two types of vertex orderings: *breadth-first-search orderings* and *lexicographic-breadth-first-search orderings* of Rose et al [45]. Let $\sigma = [v_1, v_2, \dots, v_n]$ be any ordering of the vertex set of a graph G. We will write $a < b$ whenever in a given ordering σ vertex a has a smaller number than vertex b. Let s be a vertex of G. We define the layers of G with respect to vertex s as follows: $L_i(s) = \{v : d(s, v) = i\}$ for $i = 0, 1, 2, \dots$.

In a *breadth-first search (BFS)*, started at a vertex s, the vertices of a graph G with n vertices are numbered from n to 1 in decreasing order. The vertex s is numbered by n and put on an initially empty queue of vertices. Then a vertex v at the head of the queue is repeatedly removed, and neighbors of v that are still unnumbered are consequently numbered and placed onto the queue. Clearly, BFS operates by placing vertices in layers: the vertices closest to the start vertex are numbered first, and the most distant vertices are numbered last. BFS may be seen to generate a rooted tree T (called the *BFS-tree*) with vertex s as the root. A vertex v is the *father* in T of exactly those neighbors in G which are inserted into the queue when v is removed.

An ordering σ of the vertex set of a graph G generated by a BFS will be called a *BFS-ordering* of G. Denote by $f(v)$ the father of a vertex v with respect to σ. The following properties of a BFS-ordering will be used in what follows. Since all layers of V considered here are with respect to s, we will frequently use notation L_i instead of $L_i(s)$.

(P1) If $x \in L_i$, $y \in L_j$ and $i < j$, then $x > y$ in σ.
(P2) If $x, y, z \in L_j$, $x > y > z$ and $f(x)z \in E$, then $f(x) = f(y) = f(z)$ (in particular, $f(x)y \in E$).

We will also need the following fact.

Lemma 6 [16] *Let G be an arbitrary graph and T be a BFS-tree of G with the root s. Also let v be a vertex of G and w ($w \neq v$) be an ancestor of v in T from layer $L_i(s)$. Then, for any vertex $x \in L_i(s) \setminus \{w\}$ with $d_G(v, w) = d_G(v, x)$, inequality $x < w$ holds.*

Lexicographic breadth-first search (LexBFS), started at a vertex s, orders the vertices of a graph by assigning numbers from n to 1 in the following way. The vertex s gets the number n. Then each next available number k is assigned to a vertex v (as yet unnumbered) which has lexically largest vector $(a_n, a_{n-1}, \dots, a_{k+1})$, where $a_i = 1$ if v is adjacent to the vertex numbered i, and $a_i = 0$ otherwise.

An ordering of the vertex set of a graph generated by LexBFS we will call a *LexBFS-ordering*. Clearly, any LexBFS-ordering is a BFS-ordering (but not conversely). Note also that for a given graph G, both a BFS-ordering and a LexBFS-ordering can be generated in linear time [30]. LexBFS may be seen to generate a special BFS-tree T (called *LexBFS-tree*) with vertex s as the root.

4.1 k-Chordal Graphs

In this subsection, we will show that each k-chordal graph admits an additive $(k-1)$-fframe (which is also an additive $(k-1)$-frame) and each chordal graph admits an additive 1-fframe (which is also an additive 1-frame). 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.

For chordal graphs we will need the following lemmata from [12] and [45].

Lemma 7 [12] *If vertices a and b of a disk $D_k(s)$ of a chordal graph G are connected by a path $P(a,b)$ outside of $D_k(s)$ [i.e., $P(a,b)\cap D_k(s)=\{a,b\}$], then a and b must be adjacent in G.*

Lemma 8 [45] *Let G be a chordal graph and σ be a LexBFS-ordering of G. For any vertices a,b,c of G with $a<b<c$ in σ and $b,c\in N_G(a)$, b and c must be adjacent in G.*

Now, we prove the main result of this subsection.

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, by Corollary 3, an additive $(k-1)$-frame) of G. If G is a chordal graph (i.e., $k=3$), then any LexBFS-tree T of G is an additive 1-fframe (and, hence, by Corollary 3, an additive 1-frame) of G.*

Proof First of all notice that since T is a BFS-tree of G rooted at s, for any vertex $v\in V$, $d_G(v,s)=d_T(v,s)$, and, for any edge $uv\in E$, $|d_T(u,s)-d_T(v,s)|\le 1$.

Assume that we want to route from a source vertex x to a target vertex y $(y\ne x)$ in G. If $x\in sTy$ or $y\in sTx$ then, according to the IGRF strategy, $R_{G,T}(x,y)=xTy$ (by Lemma 1) and, therefore, the length of $R_{G,T}(x,y)$ is equal to $d_G(x,y)$.

Let $P_G(x,y)$ be an arbitrary shortest path between x and y in G. We may assume that $d_G(x,y)\ge 2$, since otherwise both the IGRF routing path $R_{G,T}(x,y)$ and the shortest path $P_G(x,y)$ have length 1. So, we only need to consider the case when $x\notin sTy$, $y\notin sTx$ and $d_G(x,y)\ge 2$.

By Lemma 1, Corollary 3 and Lemma 3, the routing path $R_{G,T}(x,y)$ is induced and can be decomposed into three parts: subpath $x\dots a$ of path xTs, edge ab of G and subpath $b\dots y$ of path sTy. We have $|d_T(a,s)-d_T(b,s)|\le 1$.

Assume now that the length of path $R_{G,T}(x,y)$ is at least $d_G(x,y)+k$. Let z be a vertex of $P_G(x,y)$ closest to s in G. Consider also vertices $y',y''\in yTs$ and $x',x''\in xTs$ with $d_T(x',s)=d_T(z,s)=d_T(y',s)$ and $d_T(x'',s)=d_T(z,s)-1=d_T(y'',s)$ (see Fig. 4). As T is a BFS-tree, we have $d_G(x,x')\le d_G(x,z)$ and $d_G(y,y')\le d_G(y,z)$. We may assume that $d_G(a,s)\le d_G(x'',s)$ and $d_G(b,s)\le d_G(y'',s)$, since otherwise $d_G(x,y)+k\le length(R_{G,T}(x,y))\le d_G(x,x')+2+d_G(y,y')\le d_G(x,z)+d_G(y,z)+2=d_G(x,y)+2$, and a contradiction with $k\ge 3$ arises.

Let R' be the subpath of $R_{G,T}(x,y)$ between x'' and y''. Since:

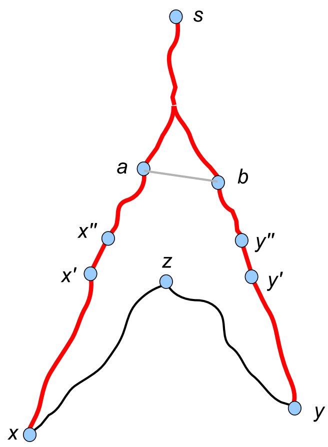


Fig. 4 Illustration to the proof of Theorem 3 (Color figure online)

- $d_G(x,y) + k \le length(R_{G,T}(x,y)) = length(xTx') + 1 + length(R') + 1 + length(y'Ty)$;
- $length(xTx') \le d_G(x,z)$;
- $length(y'Ty) \le d_G(y,z)$; and
- $d_G(x,y) = d_G(x,z) + d_G(y,z)$,

we get $length(R') \ge k-2$. Furthermore, if $length(R') = k-2$, then $d_G(x,x') = d_G(x,z)$ and $d_G(y,y') = d_G(y,z)$.

In the subgraph of G induced by vertices $x''Tx \cup P_G(x,y) \cup yTy''$, consider a shortest path P' connecting vertices x'' and y''. Clearly, no inner vertex of P' is adjacent to any inner vertex of R', i.e, paths R' and P' form an induced cycle in G. Since G cannot have induced cycles of length greater than k and $length(R') \ge k-2$, we conclude that either $length(P') = 1$ (i.e., $x''y'' \in E$, $x'' = a$, $y'' = b$, $length(R') = 1$ and $k = 3$), or $length(P') = 2$ (i.e., $length(R') = k-2 \ge 2$, $k \ge 4$ and $x''z, y''z \in E$). In both cases, $d_G(x,x') = d_G(x,z)$ and $d_G(y,y') = d_G(y,z)$ because $length(R') = k-2$. Also, in both cases, $x''z, y''z \in E$ must hold. Indeed, when $k = 3$, G is a chordal graph and in a cycle of G formed by vertices $x''Tx \cup P_G(x,y) \cup yTy''$, edge $x''y''$ must belong to a triangle. It is evident that the third vertex of that triangle must be z. Now, since $d_G(x,x') = d_G(x,z)$ and $d_G(y,y') = d_G(y,z)$, by Lemma 6, $z < x'$ and $z < y'$. On the other hand, since $x''z, y''z \in E$, by property (P2) of BFS-ordering, $x'' = y''$ must hold, which is impossible. The contradiction obtained proves that the length of path $R_{G,T}(x,y)$ is at most $d_G(x,y) + k - 1$.

To prove the second part of the theorem, let G be a chordal graph (i.e., $k = 3$) and let the length of path $R_{G,T}(x,y)$ be equal to $d_G(x,y) + 2$. Using the same notations as before and denoting by $R(x',y')$ the subpath of $R_{G,T}(x,y)$ between x' and y', we get $d_G(x,y) + 2 = length(R_{G,T}(x,y)) = length(xTx') + length(R(x',y')) + length(y'Ty) \le d_G(x,z) + length(R(x',y')) + d_G(y,z) = d_G(x,y) + length(R(x',y'))$, i.e., $length(R(x',y')) \ge 2$. Furthermore, if $length(R(x',y')) = 2$, then $d_G(x,x') = d_G(x,z)$ and $d_G(y,y') = d_G(y,z)$, and if $length(R(x',y')) = 3$, then $d_G(x,x') = d_G(x,z)$ or $d_G(y,y') = d_G(y,z)$.

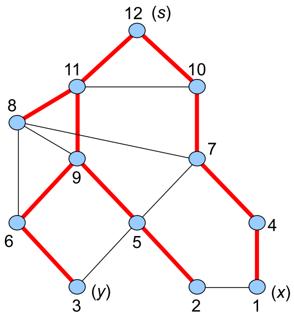


Fig. 5 A 5-chordal graph with a LexBFS-ordering: $length(R_{G,T}(x,y)) = 7 = 4 + d_G(x,y)$ (Color figure online)

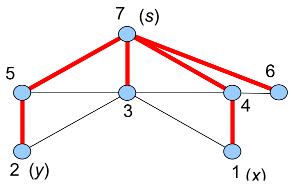


Fig. 6 A chordal graph with a BFS-ordering: $d_G(x,y) = 2 = 4 - 2 = length(R_{G,T}(x,y)) - 2$ (Color figure online)

Assume $d_G(x,x') = d_G(x,z)$ and $d_G(y,y') = d_G(y,z)$. Then, by Lemma 6, $z < x'$ and $z < y'$, and, by Lemma 7, $x'z, y'z \in E$. Since $x'y' \notin E$, a contradiction with Lemma 8 occurs.

Hence, we must have $length(R(x',y')) > 2$, i.e., $x'y', x'y'', y'x'' \notin E$ and $x'' \neq y''$. Furthermore, by Lemma 7, vertices x'' and y'' are adjacent, i.e., $length(R(x',y')) = 3$. Assume, without loss of generality, that $d_G(x,x') = d_G(x,z)$. Then, necessarily, $d_G(y,y') = d_G(y,z) - 1$ must hold. By Lemmas 6 and 7, we have $z < x'$ and $x'z \in E$. According to property (P1) of BFS-ordering, $y'' > z$. Therefore, by Lemma 8, vertices z and y'' cannot be adjacent (otherwise, $x'y'' \in E$, which is impossible). Consider a cycle C of G formed by vertices $z, x', x'', y''Ty$ and a subpath $P(z,y)$ of $P_G(x,y)$ between z and y. Since G is chordal, edge $x''y''$ must form a triangle with some other vertex of C. As $zy'' \notin E$, the neighbor w of z in $P(z,y)$ must be in $L_{d_G(s,z)}(s)$ to form that triangle. Note that $d_G(y,y') = d_G(y,w) = d_G(y,z) - 1$. Again, by Lemmas 6 and 7, we have $w < y'$ and $y'w \in E$. According to property (P1) of BFS-ordering, $x'' > w$. As $x''w \in E$ and $x''y' \notin E$, a contradiction with Lemma 8 arises. This contradiction proves that the length of path $R_{G,T}(x,y)$ is at most $d_G(x,y) + 1$. □

Figure 5 shows that the result of Theorem 3 is tight for 5-chordal graphs, and cannot be improved if we consider a LexBFS-tree instead of a BFS-tree. Figure 6 shows a chordal graph for which the result of Theorem 3 is tight, and the result is not true anymore if LexBFS-tree is replaced by BFS-tree. It is easy to see (by a simple case analysis) that even the chordal graph obtained from the graph in Fig. 6 by removing vertex y has neither an additive 0-carcass nor a 0-frame, but has an additive 0-fframe (see Fig. 8). Figure 9 presents a chordal graph with an additive 0-frame (which is also an additive 0-fframe) and this graph does not have any additive 0-carcass. It is also interesting to ask whether some LexBFS-tree T of a chordal graph G is an additive r-carcass for some constant r. The following lemma is true.

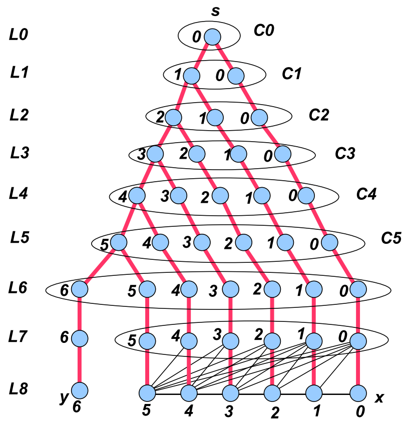


Fig. 7 A chordal graph with a LexBFS-tree. This tree is not an additive r-carcass for $r < 5$ (Color figure online)

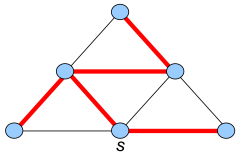


Fig. 8 A chordal graph with an additive 0-fframe. This *graph* has neither additive 0-carcass nor additive 0-frame (Color figure online)

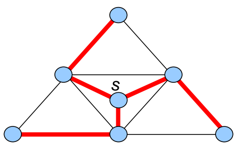


Fig. 9 A chordal graph with an additive 0-frame (which is also an additive 0-fframe). This *graph* does not have any additive 0-carcass (Color figure online)

Lemma 9 *For any constant integer $r \geq 1$, there is a chordal graph G with a LexBFS-tree T such that T is not an additive r-carcass of G.*

Proof We will construct a chordal graph G for $r = 4$. It will be clear how the method can be extended to an arbitrary r. Figure 7 shows such a graph. It has 9 levels, from L_0 to L_8 (vertices on each level are labeled/numbered as shown in the figure). Levels L_6, L_7 and L_8 have 7 vertices each. If $i \leq 6$, then level L_{i-1} has one vertex less than level L_i. Level L_0 has only one vertex s. Each level L_i ($i \leq 6$) forms a clique C_i in G. Level L_7 consists of a clique C_7, formed by vertices $\{0, 1, \dots, 5\}$, and of an isolated vertex 6. Level L_8 consists of an induced path formed by vertices $\{0, 1, \dots, 5\}$, and of an isolated vertex 6. Each clique $C_0, \dots, C_7$ is marked by a circle. This describes the vertex set and the inner-level edges of G. Additionally, all vertices labeled by 6 form an induced path in G, and, for each $k = 1, \dots, 8$, vertex i ($i \neq 6$ if $k = 7, 8$) in L_k is adjacent to any vertex j in L_{k-1} with $j \leq i$.

One can easily verify that the graph G constructed is chordal, and that the spanning tree depicted in bold (red) is a LexBFS-tree of G. Let us name a vertex by (i, j), where i is the index of level it belongs to and j is its label in that level. By the TDGR strategy, a message from $x = (8, 0)$ to $y = (8, 6)$ will use the following path which has 10 edges: $(8, 0) \rightarrow (8, 1) \rightarrow (8, 2) \rightarrow (8, 3) \rightarrow (8, 4) \rightarrow (8, 5) \rightarrow$

$(7, 5) \to (6, 5) \to (6, 6) \to (7, 6) \to (8, 6)$, while a shortest path between x and y has only 5 edges (for example, $(8, 0) \to (7, 0) \to (6, 0) \to (6, 6) \to (7, 6) \to (8, 6)$). Thus, T is not an additive 4-carcass of G. □

4.2 Chordal Bipartite Graphs and AT-Free Graphs

In this subsection, we will show that each chordal bipartite graph admits an additive 0-fframe (which is also an additive 0-frame) and each AT-free graph admits an additive 2-fframe (which is also an additive 2-frame).

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. We will show that every chordal bipartite graph admits a special LexBFS-tree which is an additive 0-frame as well as an additive 0-fframe. We will need the following result from [16].

Lemma 10 [16] *Let G be a chordal bipartite graph. Then, there is a LexBFS-ordering σ of G with the property: for any vertices a, b, c, d of G such that $ab, ac, bd \in E$, $a < d$, $b < c$ in σ, d and c must be adjacent in G. Such a special LexBFS-ordering of G can be found in $O(n^2)$ time.*

It is also easy to prove the following lemma (an analog of Lemma 7) for chordal bipartite graphs.

Lemma 11 *If vertices a and b of a disk $D_k(s)$ of a chordal bipartite graph G are connected by a path $P(a, b)$ outside of $D_k(s)$ [i.e., $P(a, b) \cap D_k(s) = \{a, b\}$], then a and b must have a common neighbor in $L_{k+1}(s) \cap P(a, b)$.*

Theorem 4 *Every chordal bipartite graph G admits a special LexBFS-tree which is an additive 0-fframe (and, hence, an additive 0-frame) of G.*

Proof We will use the same notation as in the proof of Theorem 3. Let T be the LexBFS-tree associated with that special LexBFS-ordering of chordal bipartite graph G (indicated in Lemma 10). Assume, the root of T is s. Notice that, since G is bipartite, there is no edge ab in G with $d_G(a, s) = d_G(b, s)$ (call such an edge *horizontal*). As in the proof of Theorem 3, we only need to consider the case when $x \notin sTy$, $y \notin sTx$ and $d_G(x, y) \geq 2$.

Let $P_G(x, y)$ be a shortest path between x and y in G with $d_G(s, P_G(x, y))$ is minimum. Let again $R_{G,T}(x, y)$ be the routing path from x to y produced by the IGRF strategy. Vertices $z \in P_G(x, y)$, $y', y'' \in yTs$ and $x', x'' \in xTs$ are defined as in the proof of Theorem 3. Note that, since G is bipartite (no horizontal edges), $x''y'', x'y', x'z, y'z \notin E$.

First assume that $x'y'', y'x'' \notin E$. Hence, $x'' \neq y''$. By Lemma 11, we may assume that $x''z, y''z \in E$. Let, without loss of generality, $x'' < y''$. Applying Lemma 10 to vertices x', x'', z, y'' with $x'y'' \notin E$, we conclude that $z > x'$ must hold. But then, by Lemma 6, $d_G(x, x') \neq d_G(x, z)$. Since G is bipartite, the latter implies $d_G(x, x') = d_G(x, z) - 2$, i.e., there is a shortest path between x and y in G involving vertices x', x'', z. Since path $P_G(x, y)$ was chosen with minimum $d_G(s, P_G(x, y))$, a contradiction occurs.

So, we may assume that $x''y' \in E$ or $x'y'' \in E$, i.e., $length(R_{G,T}(x, y)) \le d_G(x, x') + 2 + d_G(y, y')$. If $d_G(x, x') \ne d_G(x, z)$, then $d_G(x, x') = d_G(x, z) - 2$ and, therefore, $length(R_{G,T}(x, y)) \le d_G(x, x') + 2 + d_G(y, y') \le d_G(x, z) - 2 + 2 + d_G(y, z) = d_G(x, y)$, implying T is an additive 0-fframe of G. The statement of the theorem is true also if $x' = y'$.

We may assume, now, that $x' \ne y'$, $d_G(x, x') = d_G(x, z)$ and $d_G(y, y') = d_G(y, z)$. Note that, in this case, $L_{d_G(z,s)}(s) \cap P_G(x, y)$ contains only z. We claim that $d_G(x, x') = d_G(x, y')$ or $d_G(y, y') = d_G(y, x')$. If not, then $z \ne x'$ and $z \ne y'$. Consider a common neighbor $a \in L_{d_G(z,s)+1}(s)$ of z and x' with $d_G(a, x) = d_G(x', x) - 1$ and a common neighbor $b \in L_{d_G(z,s)+1}(s)$ of z and y' with $d_G(b, y) = d_G(y', y) - 1$ (they exist by Lemma 11). Necessarily, $ay', bx' \notin E$. By Lemma 6, we have $z < x'$ and $z < y'$. Without loss of generality, let $a < b$. Applying Lemma 10 to vertices x', a, z, b, we get $bx' \in E$, which is a contradiction.

Thus, $d_G(x, x') = d_G(x, y')$ or $d_G(y, y') = d_G(y, x')$ must hold. Let, without loss of generality, $d_G(y, y') = d_G(y, x')$. Consider the neighbor v of y' in yTy'. By Lemma 6, $x' < y'$, and, by Lemma 11, there is a common neighbor $u \in L_{d_G(x',s)+1}(s)$ of x' and y' with $d_G(u, y) = d_G(y', y) - 1$. If $u = v$ then $length(R_{G,T}(x, y)) \le d_G(x, x') + d_G(y, v) + 1 = d_G(x, x') + d_G(y, y') \le d_G(x, z) + d_G(y, z) = d_G(x, y)$, implying T is an additive 0-fframe of G. If $u \ne v$ (i.e., $x'v \notin E$) then, by Lemma 6, $u < v$, and, by Lemma 11, there is a common neighbor $w \in L_{d_G(x',s)+2}(s)$ of u and v with $d_G(w, y) = d_G(y', y) - 2$. By property (P1) of BFS-ordering, $w < x'$. Applying Lemma 10 to vertices x', u, w, v, we get a contradiction with $x'v \notin E$. This final contradiction completes the proof. □

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 5 *Let G be an AT-free graph. Any BFS-tree T of G is an additive 2-fframe (and, hence, an additive 2-frame) of G.*

Proof We will again use the same notations as in the proof of Theorem 3. Let T be a BFS-tree of G, rooted at s. As before, we only need to consider the case when $x \notin sTy$, $y \notin sTx$ and $d_G(x, y) \ge 2$.

Let $P_G(x, y)$ be any shortest path between x and y in G. Let again $R_{G,T}(x, y)$ be the routing path from x to y produced by the IGRF strategy. Vertices $z \in P_G(x, y)$, $y', y'' \in yTs$ and $x', x'' \in xTs$ are defined as in the proof of Theorem 3. If $x' = y'$ or $x'' = y''$ or $x'y'' \in E$ or $x''y' \in E$ or $x'y' \in E$, then $length(R_{G,T}(x, y)) \le d_G(x, x') + 2 + d_G(y, y') \le d_G(x, z) + d_G(y, z) + 2 = d_G(x, y) + 2$, implying T is an additive 2-fframe of G.

We may assume then that $x' \ne y'$, $x'' \ne y''$, $x'y'', x''y', x'y' \notin E$. Consider vertices s, x' and y' of G. They form an asteroidal triple, since path $x'Tx \cup P_G(x, y) \cup yTy'$ avoids the closed neighborhood of s, path $x'Ts$ avoids the closed neighborhood of y', and path $y'Ts$ avoids the closed neighborhood of x'. As G cannot have asteroidal triples, this situation is not possible, proving the theorem. □

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). 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.

5.1 Tree-Length λ Graphs

Tree-decomposition is a rich concept introduced by Robertson and Seymour [44] and is widely used to solve various graph problems. In particular efficient algorithms exist for graphs having a tree-decomposition into subgraphs (or *bags*) of bounded size, i.e., for bounded *tree-width* 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 [15]. 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. [8, 14]). AT-free graphs and distance-hereditary graphs are of tree-length 2. More generally, [26] showed that 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.

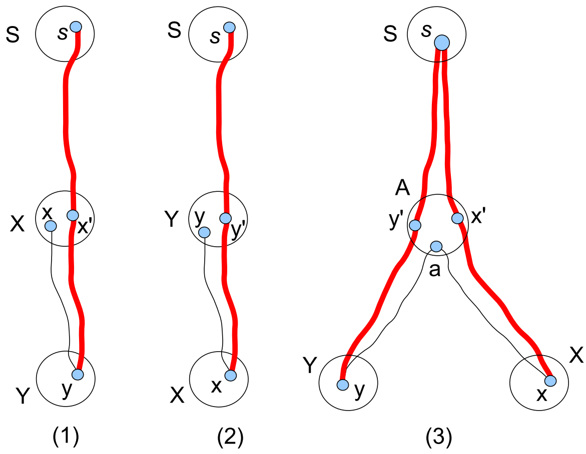
We now recall the definition of *tree-decomposition* introduced by Robertson and Seymour in their work on graph minors [44]. A tree-decomposition of a graph G is a tree T whose vertices, called *bags*, are subsets of $V(G)$ such that:

1. $\bigcup_{X\in V(T)} X = V(G)$;
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.

A well-known invariant related to tree-decompositions of a graph G is the *tree-width*, defined as minimum of $\max_{X\in V(T)} |X| - 1$ over all tree-decompositions T of G. We stress that the tree-width of a graph is not related to its tree-length. For

Fig. 10 (1) $NCA_{\mathcal{T}}(X,Y)=X$; (2) $NCA_{\mathcal{T}}(X,Y)=Y$; (3) $NCA_{\mathcal{T}}(X,Y):=A$ is neither X nor Y (Color figure online)



instance, cliques have unbounded tree-width and tree-length 1, whereas cycles have tree-width 2 and unbounded tree-length.

We will need the following property of tree-decomposition.

Proposition 1 [14] *Let X be a bag of a tree-decomposition T of G, and T_1, T_2 be arbitrary two different subtrees of $T\setminus\{X\}$ (obtained after removing bag X from T). Then, X separates in G vertices belonging to bags of T_1 but not to X from vertices belonging to bags of T_2 but not to X.*

First, we prove the main, positive result of this subsection.

Theorem 6 *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.*

Proof Assume $\mathcal{T}$ is a tree-decomposition of G of length λ, and T is a BFS-tree of G rooted at an arbitrary vertex s. Let $R_{G,T}(x,y)$ be the routing path from a vertex x to a vertex y produced by the λ-localized IGRF scheme using tree T. Let assume also that $\mathcal{T}$ is rooted at a bag S containing vertex s, and let X (resp., Y) be the closest to S bag in $\mathcal{T}$ containing vertex x (resp., vertex y). If $X=Y$, then $y\in D_\lambda(x,G)$, and the length of $R_{G,T}(x,y)$ is $d_G(x,y)$. Consider the nearest common ancestor $A=NCA_{\mathcal{T}}(X,Y)$ of X and Y in $\mathcal{T}$. We have three possible cases: $A=X$, $A=Y$ or A is different from both X and Y. Figure 10 shows all three cases (and bold paths there are paths of the BFS-tree T).

If $A=X$ then, according to Proposition 1, there must exist a vertex $x'\in X$ such that $x'\in yTs$ and $x'\in D_\lambda(x,G)$. It is easy to see then that the length of $R_{G,T}(x,y)$ is at most $d_G(x',y)+\lambda\le d_G(x,y)+2\lambda$.

Now let $A=Y$. There must exist a vertex $y'\in Y$ such that $y'\in xTs$. Since $y\in D_\lambda(y',G)$, we conclude that there must exist a vertex $x'\in xTy'$ (including the case $x'=y'$) and a vertex $y''\in yTs$ (including the case $y''=y$) such that $y''\in D_\lambda(x',G)$. If there is more than one such x', we assume x' is chosen to be closest to x in T. If there is more than one such y'', we assume y'' is chosen to be closest to y in T.

We have $d_G(x', y'') \le \lambda$, $d_G(y, y') \le \lambda$, and, by the triangle inequality, $d_G(y'', y) \le d_G(y'', x') + d_G(x', y') + d_G(y', y) \le d_G(x', y') + 2\lambda$. Since T is a BFS-tree, according to the λ-localized IGRF scheme, we get $length(R_{G,T}(x, y)) = d_T(x, x') + d_G(x', y'') + d_T(y'', y) = d_G(x, x') + d_G(x', y'') + d_G(y'', y) \le d_G(x, x') + \lambda + d_G(x', y') + 2\lambda = d_G(x, y') + 3\lambda \le d_G(x, y) + 4\lambda$.

Finally let $A \neq X$ and $A \neq Y$. There exist $x' \in A$, $y' \in A$ and $a \in A$ such that $x' \in xTs$, $y' \in yTs$, and a is on a shortest path from x to y in G. Since $y' \in D_\lambda(x', G)$, there must exist a vertex $x'' \in xTx'$ (including the case $x'' = x'$) and a vertex $y'' \in yTs$ (including the case $y'' = y'$) such that $y'' \in D_\lambda(x'', G)$. If there is more than one such x'', we assume x'' is chosen to be closest to x in T. If there is more than one such y'', we assume y'' is chosen to be closest to y in T. We have $y'' \in y'Ty$ or $y'' \in y'Ts$.

According to the λ-localized IGRF scheme, $length(R_{G,T}(x, y)) = d_T(x, x'') + d_G(x'', y'') + d_T(y'', y) = d_G(x, x'') + d_G(x'', y'') + d_G(y'', y) \le d_G(x, x'') + \lambda + d_G(y, y'')$. We know also that $d_G(x, y) = d_G(y, a) + d_G(x, a) \ge d_G(x, x') - \lambda + d_G(y, y') - \lambda = d_G(x, x') + d_G(y, y') - 2\lambda$. Furthermore, $d_G(y', y'') \le d_G(y', x') + d_G(x', x'') + d_G(x'', y'') \le d_G(x', x'') + 2\lambda$. Since $y'' \in y'Ty$ or $y'' \in y'Ts$, we also get $d_G(y, y'') \le d_G(y, y') + d_G(y', y'')$.

Consequently, $length(R_{G,T}(x, y)) \le d_G(x, x'') + \lambda + d_G(y, y'') \le d_G(x, x'') + \lambda + d_G(y, y') + d_G(y', y'') \le d_G(x, x'') + d_G(x', x'') + 3\lambda + d_G(y, y') = d_G(x, x') + 3\lambda + d_G(y, y') \le d_G(x, y) + 5\lambda$, and T is a λ-localized additive 5λ-fframe (and a λ-localized additive 5λ-frame) of G. □

5.2 Lower Bound Results

Now, we prove some lower bound results.

Lemma 12 *For any $\lambda \ge 3$, there exists a planar tree-length λ graph without any $(\lambda - 2)$-localized additive (λa)-fframe for any constant $a \ge 1$.*

Proof To prove this lemma, we construct a gadget graph composed of rockets (it will be of tree-length λ). A rocket consists of 1 triangle and b rectangles where $b \ge \lambda - 1$ (see [13] for other use of similar gadgets). Each horizontal "edge" of the triangle and any rectangle is a path of length $\lambda - 1$. Rectangle 1 shares a horizontal "edge" with the triangle. Rectangle i shares a horizontal "edge" with rectangle $i - 1$ and rectangle $i + 1$, respectively, for $2 \le i \le b - 1$. Figure 11 shows an example of a rocket consisting of 1 triangle and 5 rectangles. The top vertex on the triangle is labeled as root vertex s. The bottom left vertex and bottom right vertex are marked as p and q. We mark, on each horizontal path, the left-most vertex by l_i and the right-most vertex by r_i, where i is the level of the path (see Fig. 11). These vertices (which include p and q) are regarded as terminal vertices. The gadget is formed (in a tree-like manner) by taking one rocket as the "root rocket" and then developing the gadget by identifying the root vertex of a child rocket with a terminal vertex of its parent. In addition, we say that the gadget has "depth" 1 if it contains only 1 rocket, and has "depth" $k (k > 1)$ if all terminal vertices of the root gadget are the roots of a gadget of "depth" $k - 1$.

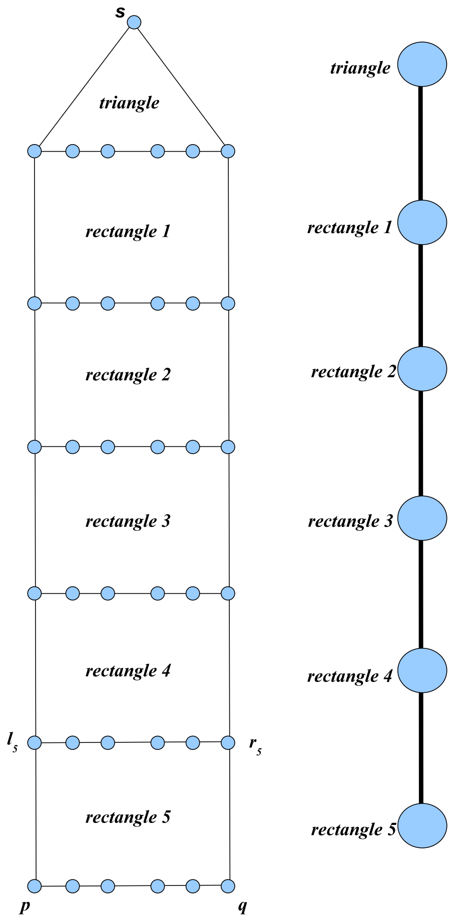


Fig. 11 A tree-length 6 graph and its corresponding tree-decomposition (Color figure online)

Let T be any (rooted) spanning tree of the gadget of "depth" k. We may assume that the spanning tree T is rooted at the root vertex of the "root rocket", which is our special vertex and we name it s_0. If T is not rooted at s_0, we can take two identical copies of the gadget of "depth" k, glue them at s_0, and then any rooted spanning tree of the resulting graph will produce in at least one of the two copies a spanning tree with root s_0. (This will only double the number of vertices in the graph and we will still work only in the appropriate copy). Furthermore, the subtree of T spanning any given rocket of that copy can be considered as rooted at the root of the rocket.

So, we assume that T is rooted at the root vertex s_0 of the "root rocket" of the gadget G with "depth" k. The following two claims are true.

Claim 1 *If T is a $(\lambda - 2)$-localized additive (λa)-fframe of G, where $a < \frac{2b+3}{\lambda} - 1$, then in each rocket, there must exist a horizontal path which is a path in T.*

Proof We can prove the claim by contradiction. Suppose that there is a rocket where no horizontal path is a path in T. Then, it is easy to conclude that the routing path

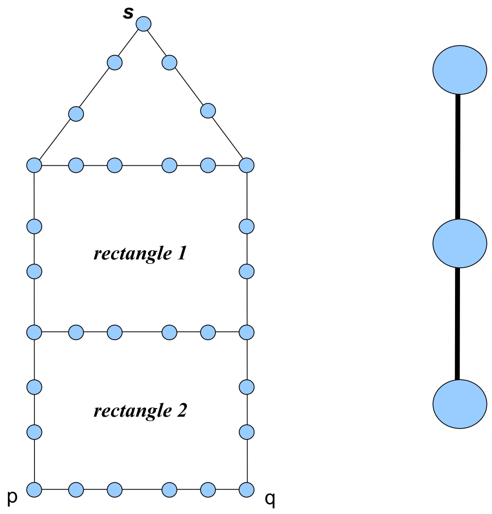


Fig. 12 A tree-length 8 graph and its corresponding tree-decomposition (Color figure online)

produced by the $(\lambda-2)$-localized IGRF strategy from p to q in the rocket is of length $2b+2$ (no matter how T spans the rocket), while the shortest path between p and q is $\lambda-1$. This contradicts the assumption that T is a $(\lambda-2)$-localized additive (λa)-fframe, where $a < \frac{2b+3}{\lambda} - 1$. □

Claim 2 *If the gadget G has "depth" $k > 2a+\lambda-3$, then T is not a $(\lambda-2)$-localized additive (λa)-fframe of G where $a < \frac{2b+3}{\lambda} - 1$.*

Proof We can prove the claim by contradiction. Suppose T is a $(\lambda-2)$-localized additive (λa)-fframe of G where $a < \frac{2b+3}{\lambda} - 1$. By Claim 1, in each rocket there exists a horizontal path which is a path in T.

Let consider an arbitrary rocket Λ at "depth" greater than $\lambda-2$. According to the gadget construction, $s_0 \notin D_{\lambda-2}(v, G)$ for any vertex v of the rocket Λ. Recall that s_0 is the root vertex of the "root rocket". Therefore, the routing path from v to s_0 produced by the $(\lambda-2)$-localized IGRF strategy will follow the spanning tree T in rocket Λ. By Claim 1, in rocket Λ there exists a horizontal path, name it *LR*, which is a path in T. Let the left most vertex of the path *LR* be L, and the right most vertex be R, and the root vertex of Λ be s_Λ. Since *LR* is a T path, it is easy to see that either the tree path from L to s_Λ passes through R, or the tree path from R to s_Λ passes through L. Therefore, either from R to s_0 or from L to s_0, the route produced by the $(\lambda-2)$-localized IGRF strategy will divert, in rocket Λ, from the corresponding shortest G path (in particular, we will have $d_T(L, s_\Lambda) \geq d_G(L, s_\Lambda) + \lambda - 1$ or $d_T(R, s_\Lambda) \geq d_G(R, s_\Lambda) + \lambda - 1$).

In what follows, we will locate a vertex t in a rocket at "depth" greater than $2a + \lambda - 3$, such that the route from t to s_0 produced by the $(\lambda-2)$-localized IGRF strategy will have length at least $d_G(t, s_0) + \lambda a + 1$. Starting from an arbitrary rocket Λ at "depth" $\lambda-1$, we identify a terminal vertex v with $d_T(v, s_\Lambda) \geq d_G(v, s_\Lambda) + \lambda - 1$ (such a vertex always exists according to the above analysis). Now, starting from the rocket rooted at v, we will repeat the same procedure, until we identify such a vertex

t in a rocket at "depth" greater than $2a + \lambda - 3$. Since the route $R_{G,T}(t, s_0)$ from t to s_0 produced by the $(\lambda - 2)$-localized IGRF strategy will follow the spanning tree T in any rocket of "depth" greater than $\lambda - 2$ and, in each such rocket, $R_{G,T}(t, s_0)$ has surplus $\lambda - 1$ with respect to shortest path, we conclude that $length(R_{G,T}(t, s_0)) - d_G(t, s_0) \geq 2a(\lambda - 1) \geq \lambda a + a \geq \lambda a + 1$. The latter means that T is not a $(\lambda - 2)$-localized additive (λa)-fframe of G. □

Now we can finish the proof of the lemma. Given $\lambda(\lambda > 2)$ and $a \geq 1$, we create our gadget G by letting $b > \frac{\lambda(a+1)-3}{2}$ and $k > 2a + \lambda - 3$. Then, by Claims 1 and 2, G does not have any $(\lambda - 2)$-localized additive (λa)-fframe. □

Corollary 5 *For any $\lambda \geq 3$, there exists a planar 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.*

Proof It is easy to see that the number of vertices in each rocket is $(b + 1)\lambda + 1$, and the number of terminals in each rocket is $2(b+1)$, where b is the number of rectangles in a rocket. According to the construction of a gadget, the number of vertices in a gadget is $n = \frac{(2(b+1))^k-1}{2(b+1)-1}(b + 1)\lambda + 1$. Therefore, we have $\log(n - 1) = \log((2(b + 1))^k - 1) - \log(2(b + 1) - 1) + \log(b + 1) + \log\lambda$.

Since $\log((2(b + 1))^k - 1) < \log(2(b + 1))^k$ and $\log(2(b + 1) - 1) > \log(2b)$, we have $\log(n - 1) < k\log(2(b + 1)) + \log((b + 1)/2b) + \log\lambda \leq k\log(2(b + 1)) + \log\lambda$, i.e., $k > \frac{\log(\frac{n-1}{\lambda})}{1+\log(b+1)}$.

According to the proof of Lemma 12, $k > 2a + \lambda - 3$ and $b > \frac{\lambda a+\lambda-3}{2}$. Therefore, for convenience, we can choose $k = 3a\lambda$ and $b = \lambda a - 1$. It is easy to verify that $3a\lambda > 2a + \lambda - 3$ and $\lambda a - 1 > \frac{\lambda a+\lambda-3}{2}$ given the fact that $\lambda \geq 3$ and $a \geq 1$. Denote $c = \lambda a$. Then, we have $k = 3c$, $b = c - 1$ and $3c > \frac{\log(\frac{n-1}{\lambda})}{1+\log c} > \frac{\log(\frac{n-1}{\lambda})}{c/3+c}$. Finally, $c > \frac{1}{2}\sqrt{\log \frac{n-1}{\lambda}}$. □

With similar proof technique we can prove also the following results.

Lemma 13 *For any $\lambda \geq 4$, there exists a planar tree-length λ graph without any $\lfloor 2(\lambda - 2)/3 \rfloor$-localized additive (λa)-frame for any constant $a \geq 1$.*

Proof To prove this lemma, we construct a gadget G in a similar way as we did in the proof of Lemma 12. We have only few small differences (see Fig. 12).

- Each horizontal "edge" of the triangle or a rectangle is a path of length $\lceil 2(\lambda - 2)/3 \rceil + 1$, and each vertical "edge" of a rectangle (or non-horizontal "edge" of the triangle) is a path of length $\lfloor (\lambda - 2)/3 \rfloor + 1$.
- Two "central" vertices of each horizontal path of a rocket are identified as terminal vertices. All other vertices are non-terminal vertices. If a horizontal path has an even number of vertices, the two "central" vertices are exactly the two middle vertices of the path. If a horizontal path has an odd number of vertices, the two "central" vertices are the two vertices adjacent to the middle vertex.
- We assume $b \geq 2$, i.e. a rocket should have two or more rectangles.

Similarly, as in the proof of Lemma 12, we assume that spanning tree T is rooted at the root vertex of the "root rocket", which is a special vertex and we name it s_0. With the same proof technique, we can show that the following claim holds.

Claim 1 *If T is a $\lfloor 2(\lambda-2)/3\rfloor$-localized additive (λa)-frame of G, where $a < \frac{2b\lambda-4b-6}{3\lambda}$, then in each rocket, there must exist a horizontal path which is a path in T.*

Proof We can prove the claim by contradiction. Suppose that there is a rocket where no horizontal path is a path in T. Then, it is easy to conclude that the routing path produced by the $\lfloor 2(\lambda-2)/3\rfloor$-localized IGR strategy from p to q (from the bottom left vertex to the bottom right vertex) in the rocket is of length $(2b+2)(\lfloor(\lambda-2)/3\rfloor+1)$ (no matter how T spans the rocket), while the shortest path between p and q is $\lceil 2(\lambda-2)/3\rceil+1$. We have

$$\begin{aligned}&(2b+2)(\lfloor(\lambda-2)/3\rfloor+1)-(\lceil 2(\lambda-2)/3\rceil+1)\\&\quad\geq(2b+2)(\lambda-2)/3-(2(\lambda-2)/3+2)=\frac{2b(\lambda-2)-6}{3}>\lambda a.\end{aligned}$$

This contradicts the assumption that T is a $(\lfloor 2(\lambda-2)/3\rfloor)$-localized additive (λa)-frame, where $a < \frac{2b\lambda-4b-6}{3\lambda}$. □

Now we prove the following claim.

Claim 2 *If the gadget has "depth" $k > \lambda a$, then T is not a $\lfloor 2(\lambda-2)/3\rfloor$-localized additive (λa)-frame, where $a < \frac{2b\lambda-4b-6}{3\lambda}$.*

Proof We can prove the claim by contradiction. Suppose T is a $\lfloor 2(\lambda-2)/3\rfloor$-localized additive (λa)-frame where $a < \frac{2b\lambda-4b-6}{3\lambda}$. By Claim 1, in each rocket there exists a horizontal path which is a path in T.

Consider an arbitrary rocket Λ, and let a horizontal path $P(L,R)$ of Λ be a path in T. Let the left most vertex of the path $P(L,R)$ be L, and the right most vertex be R, and two "central" vertices be C_L and C_R, and the root vertex of Λ be s_Λ (see Fig. 13(a) for an illustration). Since $P(L,R)$ is a T path, it is easy to see that either the tree path from L to s_Λ passes through R, or the tree path from R to s_Λ passes through L. Without loss of generality, assume that the tree path from L to s_Λ passes through R.

Vertex s_Λ and vertices on horizontal paths other than the path $P(L,R)$ do not belong to $D_{\lfloor 2(\lambda-2)/3\rfloor}(C_L,G)$. Then, in $D_{\lfloor 2(\lambda-2)/3\rfloor}(C_L,G)$, the vertex closest to s_0 in $C_L T s_0$ is on the path of T from C_L to s_Λ, which passes through R. Hence, the route produced by the $\lfloor 2(\lambda-2)/3\rfloor$-localized IGR strategy from C_L to s_0 will divert, in rocket Λ, from the corresponding shortest G path (in particular, we will have $d_T(C_L,s_\Lambda)\geq d_G(C_L,s_\Lambda)+1$).

Again, as in the proof of Lemma 12, we can locate a vertex t in a rocket at "depth" greater than λa, such that the route from t to s_0 produced by the $\lfloor 2(\lambda-2)/3\rfloor$-localized IGR strategy will have length at least $d_G(t,s_0)+\lambda a+1$. The latter means that T is not a $\lfloor 2(\lambda-2)/3\rfloor$-localized additive (λa)-frame of G. □

Now we can finish the proof of the lemma. Given $\lambda(\lambda > 3)$ and $a \geq 1$, we create our gadget G by letting $b > \frac{3\lambda a+6}{2\lambda-4}$ and $k > \lambda a$. Then, by Claims 1 and 2, G does not have any $\lfloor 2(\lambda-2)/3 \rfloor$-localized additive (λa)-frame. □

Corollary 6 *For any $\lambda \geq 4$, there exists a planar 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.*

Proof The proof is similar to the proof of Corollary 5. We need just to mention that the number of vertices in a rocket now is $n_r = (b+1)\lambda + (b+1)\lfloor (\lambda-2)/3 \rfloor + 1 < (b+1)\frac{4\lambda}{3} + 1$, and we choose $k = \frac{3}{2}\lambda a$ and $b = \frac{2}{3}\lambda a - 1$. □

Lemma 14 *For any $\lambda \geq 6$, there exists a planar tree-length λ graph without any $\lfloor (\lambda-2)/4 \rfloor$-localized additive (λa)-carcass for any constant $a \geq 1$.*

Proof To prove this lemma, we construct a gadget G in a similar way as we did in the proof of Lemma 12 (see Fig. 11). The only difference is that we let two "central" vertices (which are defined in the proof of Lemma 13) of each horizontal path of a rocket be terminal vertices and all other vertices are non-terminal vertices. Let T be any spanning tree on the gadget G. It is easy to observe that, for any horizontal path of a rocket, there is at most one edge that is not a T edge.

First we prove the following claims.

Claim 1 *If T is a $\lfloor (\lambda-2)/4 \rfloor$-localized additive (λa)-carcass of G, where $a < \frac{2b+3}{\lambda} - 1$, then in each rocket,*

(1) *there exists a horizontal path which is a path in T, or*
(2) *there exists a horizontal path which contains a non-tree edge, say $ab \in E(G) \setminus E(T)$, such that $\min(d_G(L,a), d_G(R,b)) \leq \lfloor (\lambda-2)/4 \rfloor - 1$, where L is the leftmost vertex and R is the rightmost vertex on the path, and a is on the left side of b.*

Proof We can prove the claim by contradiction. Suppose for a rocket neither (1) nor (2) holds, i.e., each horizontal path is not a path in T and, in each horizontal path, for non-tree edge $ab \in E(G) \setminus E(T)$, $\min(d_G(L,a), d_G(R,b)) > \lfloor (\lambda-2)/4 \rfloor - 1$. Then, we have $d_G(L,b) > \lfloor (\lambda-2)/4 \rfloor$ and $d_G(R,a) > \lfloor (\lambda-2)/4 \rfloor$. Let $R_{G,T}(p,q)$ be a routing path from vertex p to vertex q produced by the $\lfloor (\lambda-2)/4 \rfloor$-localized TDGR scheme using tree T. It is easy to see that $length(R_{G,T}(p,q)) = 2b+2$, while the shortest path between p and q in G is $\lambda - 1$. The latter contradicts the assumption that T is a $\lfloor (\lambda-2)/4 \rfloor$-localized additive (λa)-carcass, where $a < \frac{2b+3}{\lambda} - 1$. □

Claim 2 *If the gadget has "depth" $k > \lambda a$, then T is not a $\lfloor (\lambda-2)/4 \rfloor$-localized additive (λa)-carcass of G where $a < \frac{2b+3}{\lambda} - 1$.*

Proof We can prove the claim by contradiction. Suppose T is a $\lfloor (\lambda-2)/4 \rfloor$-localized additive (λa)-carcass, where $a < \frac{2b+3}{\lambda} - 1$. By Claim 1, in each rocket (1) or (2) holds.

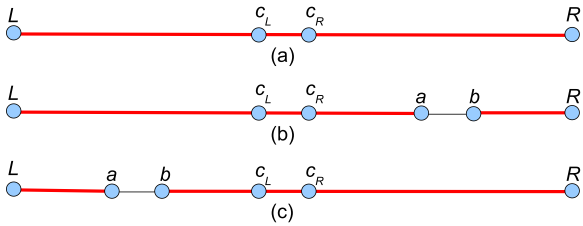


Fig. 13 (**a**) A path is a tree path. (**b**) A path is not a tree path. The edge ab is on the right side of C_R. (**c**) A path is not a tree path. The edge ab is on the left side of C_L (Color figure online)

Consider an arbitrary rocket Λ. Assume case (1) holds, i.e., there exists a horizontal path $P(L,R)$ which is a path in T (see Fig. 13(a)). Let the root vertex of Λ be s_Λ. Since $P(L,R)$ is a T path, either the tree path from L to s_Λ passes through R, or the tree path from R to s_Λ passes through L. Without loss of generality, assume that the tree path from L to s_Λ passes through R. Vertices of Λ other than vertices on the path $P(L,R)$ do not belong to $D_{\lfloor(\lambda-2)/4\rfloor}(C_L,G)$. Therefore, a route produced by the $\lfloor(\lambda-2)/4\rfloor$-localized TDGR strategy from C_L to s_0 will divert, in rocket Λ, from the corresponding shortest G path (in particular, we will have $d_T(C_L,s_\Lambda)\geq d_G(C_L,s_\Lambda)+1$).

Assume case (2) holds, i.e., there exists a horizontal path which contains a non-tree edge $ab\in E(G)\setminus E(T)$, such that $\min(d_G(L,a),d_G(R,b))\leq\lfloor(\lambda-2)/4\rfloor-1$ (see Fig. 13(b) and (c)). Without loss of generality, assume that ab is on the right side of C_R. Since $d_G(R,b)\leq\lfloor(\lambda-2)/4\rfloor-1\leq(\lambda-2)/4-1$ and $d_G(C_R,R)=\lfloor(\lambda-2)/2\rfloor>(\lambda-2)/2-1$, we conclude $d_G(C_R,b)=d_G(C_R,R)-d_G(R,b)>(\lambda-2)/4\geq\lfloor(\lambda-2)/4\rfloor$. Therefore, $b\notin D_{\lfloor(\lambda-2)/4\rfloor}(C_R,G)$. Consequently, a route produced by the $\lfloor(\lambda-2)/4\rfloor$-localized TDGR strategy from C_R to s_0 (which is the root vertex of the "root rocket") will divert, in rocket Λ, from the corresponding shortest G path (in particular, we will have $d_T(C_R,s_\Lambda)\geq d_G(C_R,s_\Lambda)+1$).

Again, as in the proof of Lemma 12, we can locate a vertex t in a rocket at "depth" greater than λa, such that the route from t to s_0 produced by the $\lfloor(\lambda-2)/4\rfloor$-localized TDGR strategy will have length at least $d_G(t,s_0)+\lambda a+1$. The latter means that T is not a $\lfloor(\lambda-2)/4\rfloor$-localized additive (λa)-carcass of G. □

Now we can finish the proof of the lemma. Given $\lambda(\lambda>5)$ and $a\geq 1$, we create our gadget G by letting $b>\frac{\lambda(a+1)-3}{2}$ and $k>\lambda a$. Then, by Claim 1 and Claim 2, G does not have any $\lfloor(\lambda-2)/4\rfloor$-localized additive (λa)-carcass. □

Corollary 7 *For any $\lambda\geq 6$, there exists a planar 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.*

Proof The proof is similar to the proof of Corollary 5. We need just to choose $k=\frac{4}{3}\lambda a$ and $b=\lambda a-1$. □

5.3 δ-Hyperbolic Graphs

δ-Hyperbolic metric spaces have been defined by M. Gromov [31] 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 [47] (see also [1]) 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 [37]). 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 [10]) that all graphs with tree-length λ are λ-hyperbolic, and each δ-hyperbolic graph has tree-length $O(\delta \log n)$.

We will need the following lemma which is an easy consequence of results in [4, 9, 11, 28, 31].

Lemma 15 *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 $d_G(a, b) \le 4\delta$ holds.*

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

Theorem 7 *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 an arbitrary vertex s. Let $R_{G,T}(x, y)$ be the routing path from a vertex x to a vertex y produced by the 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 15, 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 $d_G(a, b) \le 4\delta$ 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 $d_G(x, y) \le d_G(a, x) + d_G(b, y) \le d_G(x, y) + 1$.

Let x' be a vertex of xTs with $d_G(x', sTy) \le 4\delta$ closest to x. Clearly, x' belongs to subpath aTx of path sTx. Let y' be a vertex of path yTs with $d_G(x', y') \le 4\delta$ (i.e., $y' \in D_{4\delta}(x', G)$) closest to y. Then, according to the 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 $length(R_{G,T}(x, y)) = d_G(x, x') + d_G(x', y') + d_G(y', y) \le d_G(x, a) + 4\delta + d_G(b, y) \le d_G(x, y) + 4\delta + 1$.

Assume now that $y' \in bTs$ and $y' \ne b$. Then, we have also $x' \ne a$. 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, $d_G(y', b) \le d_G(y', x') \le 4\delta$ and, therefore, $length(R_{G,T}(x, y)) = d_G(x, x') + d_G(x', y') + d_G(y', y) \le d_G(x, a) - 1 + 4\delta + d_G(b, y') + d_G(b, y) \le d_G(x, y) + 1 - 1 + 8\delta = d_G(x, y) + 8\delta$.

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

6 Conclusion and Future Work

In this paper, we investigated 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 investigated 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 Breadth-First-Search-tree guarantees short enough routing paths. In many cases, the obtained results are optimal up to a constant factor.

Many questions and problems remain open. Here, we list only few of them.

- What other interesting graph families admit (k-localized) c-frames (c-carcasses, c-fframes) for small constants k and c?
- Can our lower bound techniques be modified to obtain lower bound results for other families of graphs?
- Given a graph G, numbers k and c, how hard is it to decide whether G admits a k-localized c-frame (c-carcass, c-fframe)? If it exists, how hard is it to construct one?
- What other (decentralized) (small piece of) information from a spanning tree of G would be useful for navigating in G?
- What other (decentralized) (small piece of) global information can be useful for navigating in graphs?

Acknowledgements We would like to thank the reviewer for many useful comments.

Appendix A: Experimental Results

In this appendix, we empirically compare the performance of TDGR, IGR and IGRF, and their corresponding k-localized versions, on Unit Disk Graphs (UDGs), which often model the wireless ad hoc networks. We use three kinds of spanning trees, breadth-first-search tree (BFST, for short), minimum-spanning tree (MST, for short), and depth-first-search tree (DFST, for short) for TDGR, IGR, and IGRF. Since the IGR scheme is the same as the IGRF scheme if the spanning tree is BFST, we only need to report on one of them. Therefore, the routing scheme pairs

Table 3 Average densities for different radiuses

Radius	150	170	190	210	230	250	270	290
Density ($\|E\|/\|V\|$)	3.184	3.857	4.853	5.784	6.822	8.134	9.576	10.311

("strategy type"–"tree type"), we report on, are BFST-IGR, BFST-TDGR, MST-IGR, MST-IGRF, MST-TDGR, DFST-IGR, DFST-IGRF, and DFST-TDGR. All methods are implemented in C++.

To generate a Unit Disk Graph, we first fix an area S, a radius R, and the number of vertices N. Then, in S we randomly generate N vertices/points. Two vertices/points are connected by an edge if and only if their Euclidean distance is at most R.

In the experiments, we care about the maximum multiplicative-stretch factor and average multiplicative-stretch factor of each routing scheme using different spanning trees. The multiplicative-stretch factor of two vertices u and v is defined as $\frac{g_{G,T}(u,v)}{d_G(u,v)}$, which is a good indication of how close the routing path is to the shortest path. Here, $g_{G,T}(u,v)$ is the length of the route produced by an appropriate strategy from u to v on G using tree T, and $d_G(u,v)$ is the distance in G between u and v. The maximum stretch factor of a graph $G=(V,E)$ is defined as $\max_{u,v\in V}\{\frac{g_{G,T}(u,v)}{d_G(u,v)}\}$, and the average stretch factor is defined as $\frac{1}{n^2}\sum_{u,v\in V}\frac{g_{G,T}(u,v)}{d_G(u,v)}$.

A.1 Performance Under Various Densities

In this set of experiments, we report on the performance of these routing schemes on randomly generated UDGs with different densities, i.e., $|E|/|V|$. However, it is difficult to "randomly" generate a UDG with a fixed density. Instead, we vary densities by choosing the radius R to be 150, 170, 190, 210, 230, 250, 270 and 290, with $|V|$ fixed to be 100. For each radius R, we randomly generate 10 UDGs. The average density of 10 UDGs corresponding to each R is listed in Table 3. In the following figures, each value is an average result on the 10 randomly generated UDGs.

The maximum multiplicative-stretch factors achieved by routing strategies under different radiuses are shown in Fig. 14. We see that DFST-IGRF and MST-IGRF have the worst maximum multiplicative-stretch factors and their performances are not stable when the radius changes. Other routing schemes have quite low maximum multiplicative-stretch factors which decrease gradually when the radius increases. Among them, BFST-IGR, BFST-TDGR, MST-IGR, and MST-TDGR have the lowest maximum multiplicative-stretch factors.

Figure 15 shows average multiplicative-stretch factors achieved by routing strategies under different radiuses. Again, DFST-IGRF and MST-IGRF have the worst performances and BFST-IGR, BFST-TDGR, MST-IGR, and MST-TDGR have the best performances.

A.2 Performance Under Various Localities

In this appendix, we show how the k-localized version of these routing strategies performs. The experimental settings are similar to those from Appendix A.1 except that $|V|$ is fixed at 120 and the radius is fixed at 130. We randomly generate 10

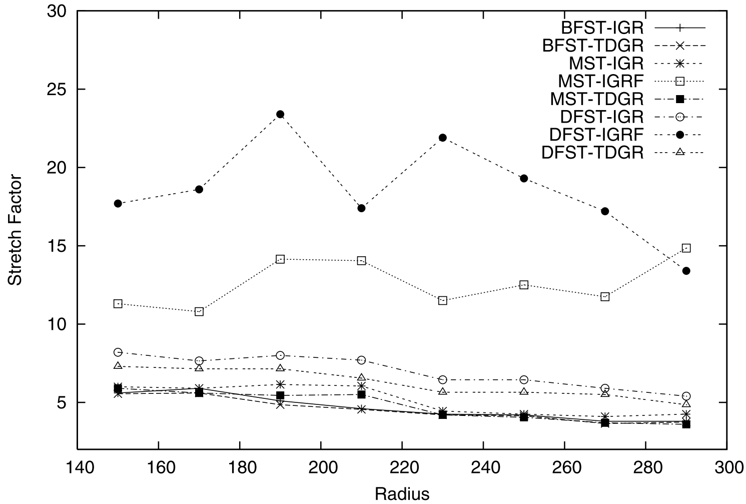


Fig. 14 Maximum multiplicative-stretch factors by varying densities (Color figure online)

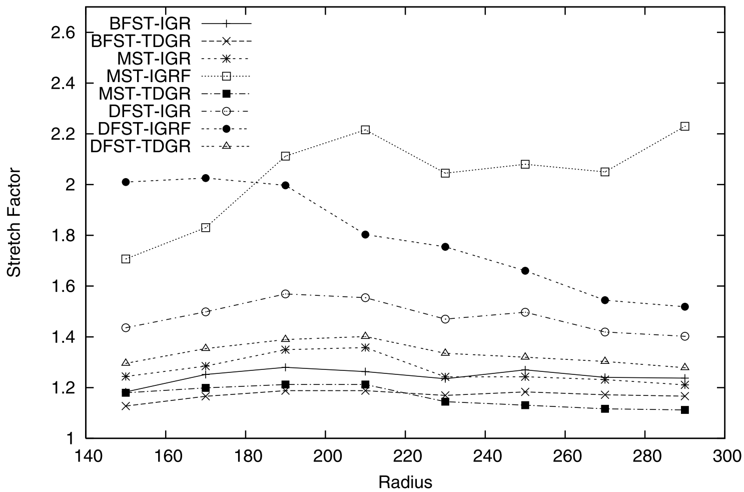


Fig. 15 Average multiplicative-stretch factors by varying densities (Color figure online)

UDGs. The average diameter of these UDGs is 18 (ranging from 16 to 20). For each UDG, we range the locality k from 1 to 8 to see how each routing strategy performs. Each value in the following figures is an average result on the 10 randomly generated UDGs.

Figure 16 shows the maximum multiplicative-stretch factors achieved by each routing scheme with different localities, and Fig. 17 shows the average multiplicative-stretch factors achieved by each routing scheme with different localities. In both figures, we observe that the multiplicative-stretch factor of each k-localized routing scheme converges to 1 when locality increases from 1 to 8. Increase in locality al-

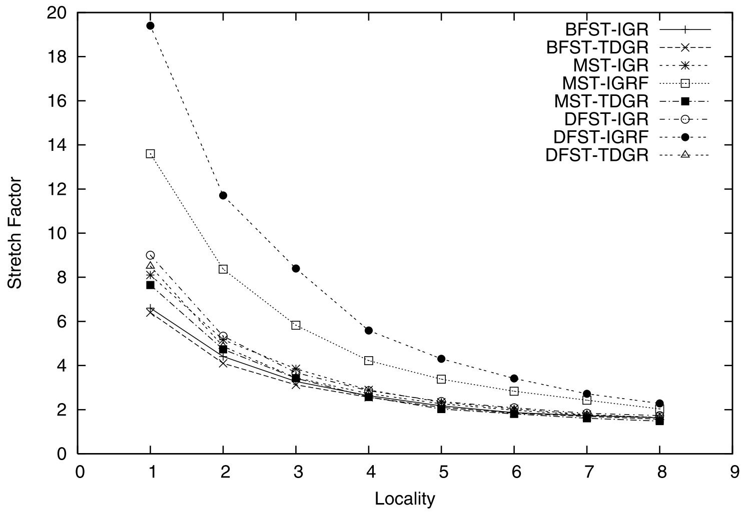


Fig. 16 Maximum multiplicative-stretch factors by varying localities (Color figure online)

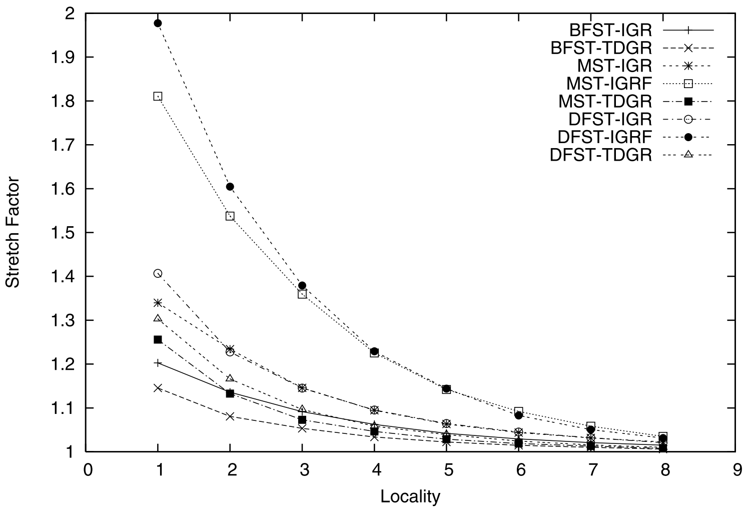


Fig. 17 Average multiplicative-stretch factors by varying localities (Color figure online)

lows to obtain better routing paths, however, it also increases the computational and communication costs. A good tradeoff between stretch factor and locality is needed.

Finally, when locality is more than 4, except for DFST-IGRF and MST-IGRF, all the other routing schemes have quite small maximum and average multiplicative-stretch factors. This is consistent with the observations from the previous subsection.

References

1. Abraham, I., Balakrishnan, M., Kuhn, F., Malkhi, D., Ramasubramanian, V., Talwar, K.: Reconstructing approximate tree metrics. In: Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC 2007), Portland, Oregon, USA August 12–15, 2007 pp. 43–52. ACM, New York (2007)
2. Adamic, L.A., Lukose, R.M., Huberman, B.A.: Local Search in Unstructured Networks. Willey, New York (2002)
3. Adamic, L.A., Lucose, R.M., Puniyani, A.R., Huberman, B.A.: Search in power-law networks. Phys. Rev. E **64**(046135), 1–8 (2001)
4. Alonso, J.M., Brady, T., Cooper, D., Ferlini, V., Lustig, M., Mihalik, M., Shapiro, M., Short, H.: Notes on word hyperbolic groups. In: Ghys, E., Haefliger, A., Verjovsky, A. (eds.) Group Theory from a Geometrical Viewpoint, ICTP, Trieste 1990, pp. 3–63. World Scientific, Singapore (1991)
5. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. In: Proceedings of the 3rd International Workshop on Discrete algorithms and Methods for Mobile Computing and Communications, pp. 48–55. ACM Press, New York (1999)
6. Brandstädt, A., Chepoi, V.D., Dragan, F.F.: The algorithmic use of hypertree structure and maximum neighbourhood orderings. Discrete Appl. Math. **82**, 43–77 (1998)
7. Brandstädt, A., Dragan, F.F., Chepoi, V.D., Voloshin, V.I.: Dually chordal graphs. SIAM J. Discrete Math. **11**, 437–455 (1998)
8. Brandstädt, A., Le Bang, V., Spinrad, J.P.: In: Graph Classes: A Survey. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia (1999)
9. Bridson, M., Haefliger, A.: Metric Spaces of Non-Positive Curvature. Springer, Berlin (1999)
10. Chepoi, V., Dragan, F.F., Estellon, B., Habib, M., Vaxès, Y.: Diameters, centers, and approximating trees of δ-hyperbolic geodesic spaces and graphs. In: Proceedings of the 24th Annual ACM Symposium on Computational Geometry, June 9–11 (2008)
11. Chepoi, V., Dragan, F.F., Estellon, B., Habib, M., Vaxès, Y., Xiang, Y.: Additive spanners and distance and routing labeling schemes for δ-Hyperbolic graphs. Algorithmica **62**, 713–732 (2012)
12. Corneil, D.G., Dragan, F.F., Köhler, E.: On the power of BFS to determine a graph's diameter. Networks **42**, 209–222 (2003)
13. Corneil, D.G., Dragan, F.F., Köhler, E., Xiang, Y.: Lower bounds for collective additive tree spanners. In preparation
14. Diestel, R.: Graph Theory, 2nd edn. Graduate Texts in Mathematics, vol. 173. Springer, Berlin (2000)
15. Dourisboure, Y., Gavoille, C.: Tree-decompositions with bags of small diameter. Discrete Math. **307**, 2008–2029 (2007)
16. Dragan, F.F.: Estimating all pairs shortest paths in restricted graph families: A unified approach. J. Algorithms **57**, 1–21 (2005)
17. Dragan, F.F., Matamala, M.: Navigating in a graph by aid of its spanning tree metric. SIAM J. Discrete Math. **25**, 306–332 (2011)
18. Elkin, M.: A faster distributed protocol for constructing a minimum spanning tree. J. Comput. Syst. Sci. **72**, 1282–1308 (2006)
19. 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: Proceedings of the Second USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI 2005) (2005)
20. Fraigniaud, P.: Small worlds as navigable augmented networks: model, analysis, and validation. In: Proceedings of the 15th Annual European Symposium, ESA 2007, Eilat, Israel, October 8–10, 2007. Lecture Notes in Computer Science, vol. 4698, pp. 2–11. Springer, Berlin (2007). 2007
21. Fraigniaud, P., Korman, A., Lebhar, E.: Local MST computation with short advice. Theory Comput. Syst. **47**, 920–933 (2010)
22. Garg, V.K., Agarwal, A.: Distributed maintenance of a spanning tree using labeled tree encoding. In: Processing of Euro-Par 2005. Lecture Notes in Computer Science, vol. 3648, pp. 606–616 (2005)
23. Gartner, F.C.: A survey of self-stabilizing spanning-tree construction algorithms. Technical report ic/2003/38, Swiss Federal Institute of Technology (EPFL) (2003)
24. Gavoille, C.: A survey on interval routing schemes. Theor. Comput. Sci. **245**, 217–253 (1999)
25. Gavoille, C.: Routing in distributed networks: overview and open problems. ACM SIGACT News-Distribute Comput. Column **32** (2001)
26. Gavoille, C., Katz, M., Katz, N.A., Paul, C., Peleg, D.: In: Approximate distance labeling schemes, 9th Annual European Symposium on Algorithms, ESA. Lecture Notes in Computer Science, vol. 2161, pp. 476–488. Springer, Berlin (2001)

27. Gavoille, C., Peleg, D., Pérennès, S., Raz, R.: Distance labeling in graphs. J. Algorithms **53**, 85–112 (2004)
28. Ghys, E., de la Harpe, P. (eds.): Les Groupes Hyperboliques d'après M. Gromov. Progress in Mathematics, vol. 83. Birkhäuser, Basel (1990)
29. Giordano, S., Stojmenovic, I.: Position based routing algorithms for ad hoc networks: a taxonomy. In: Cheng, X., Huang, X., Du, D. (eds.) Ad Hoc Wireless Networking, pp. 103–136. Kluwer, Amsterdam (2004)
30. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press, New York (1980)
31. Gromov, M.: Hyperbolic groups. In: Gersten, S.M. (ed.) Essays in group theory. MSRI Series, vol. 8, pp. 75–263 (1987)
32. Henzinger, M.R., King, V.: Maintaining minimum spanning trees in dynamic graphs. In: Proceedings of 24th International Colloquium on Automata, Languages and Programming (ICALP'97), Bologna, Italy, 7–11, July 1997. Lecture Notes in Computer Science, vol. 1256, pp. 594–604. Springer, Berlin (1997)
33. 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**, 723–760 (2001)
34. Jacquet, P., Viennot, L.: Remote spanners: what to know beyond neighbors. In: 23rd IEEE International Parallel and Distributed Processing Symposium (IPDPS 2009), pp. 1–15 (2009)
35. Karp, B., Kung, H.T.: GPSR: greedy perimeter stateless routing for wireless networks. In: Proceedings of the 6th ACM/IEEE MobiCom, pp. 243–254. ACM, New York (2000)
36. Kleinberg, J.M.: The small-world phenomenon: an algorithm perspective. In: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing (STOC 2000), May 21–23 (2000)
37. Kleinberg, R.: Geographic routing using hyperbolic space. In: Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007), Anchorage, AK pp. 1902–1909. IEEE, New York (2007)
38. Kuhn, F., Wattenhofer, R., Zhang, Y., Zollinger, A.: Geometric ad-hoc routing: of theory and practice. In: Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing, pp. 63–72. ACM, New York (2003)
39. Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., Tomkins, A.: Geographic routing in social networks. Proc. Natl. Acad. Sci. USA **102**, 11623–11628 (2005)
40. Linial, N., London, E., Rabinovich, Y.: The geometry of graphs and some of its algorithmic applications. Combinatorica **15**, 215–245 (1995)
41. Peleg, D.: Proximity-Preserving labeling schemes and their applications. J. Graph Theory **33**, 167–176 (2000)
42. Peleg, D.: Distributed Computing: A Locality-Sensitive Approach. SIAM Monographs on Discrete Math. Appl. SIAM, Philadelphia (2000)
43. Rao, A., Papadimitriou, C., Shenker, S., Stoica, I.: Geographical routing without location information. In: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom 2003), pp. 96–108 (2003)
44. Robertson, N., Seymour, P.D.: Graph minors. II. Algorithmic aspects of tree-width. J. Algorithms **7**, 309–322 (1986)
45. Rose, D., Tarjan, R.E., Lueker, G.: Algorithmic aspects on vertex elimination on graphs. SIAM J. Comput. **5**, 266–283 (1976)
46. Santoro, N., Khatib, R.: Labelling and implicit routing in networks. Comput. J. **28**, 5–8 (1985)
47. Shavitt, Y., Tankel, T.: On internet embedding in hyperbolic spaces for overlay construction and distance estimation. In: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), Hong Kong, China, March 7–11, 2004, pp. 7–11. IEEE, New York (2004)
48. Thorup, M., Zwick, U.: Compact routing schemes. In: Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA 2001), Heraklion, Crete Island, Greece, July 4–6, 2001, pp. 1–10. ACM, New York (2001)