Navigating in a graph by aid of its spanning tree metric*

Feodor F. Dragan[1] and Martin Matamala[2]

[1] Algorithmic Research Laboratory, Department of Computer Science
Kent State University, Kent, OH 44242, USA * * *
dragan@cs.kent.edu
[2] Departmento de Ingeniería Matemática, Universidad de Chile
Centro de Modelamiento Matemático UMR 2071CNRS, Santiago, Chile
mmatamal@dim.uchile.cl

Abstract. Let $G = (V, E)$ be a graph and T be a spanning tree of G. We consider the following strategy in advancing in G from a vertex x towards a target vertex y: 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 (breaking ties arbitrarily). 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). For each source vertex x and target vertex y, this way, a path, called a greedy routing path, is produced. Denote by $g_{G,T}(x, y)$ the length 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 this paper, we investigate 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 show 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 are easy to construct.

Key words. Navigating in graphs, spanning trees, distances, routing in graphs, hypercubes, rectilinear grids, chordal graphs, chordal bipartite graphs, dually chordal graphs, k-trees, efficient algorithms

AMS subject classifications. 05C05, 05C10, 05C12, 05C78, 05C85, 94C15, 68R10, 68Q25, 68W25

1 Introduction

As part of the recent surge of interest in different kind of networks, there has been active research exploring strategies for navigating synthetic and real-world networks (modeled usually as graphs). These strategies specify some rules to be used to advance in a graph (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 [25, 26, 34, 40, 31, 48] and papers cited therein); routing through common membership in groups, popularity, and geographic proximity in social networks and e-mail networks (see [3, 4, 22, 34, 39] and literature cited therein).

In this paper we use terminology used mostly for communication networks. Thus, navigation 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

* This work was partially supported by CONICYT through grants Anillo en Redes ACT08, FONDAP and BASAL-CMM projects. Results of this paper were partially presented at the ISAAC'08 conference [17].
* * * These results were obtained while the first author was visiting the Universidad de Chile, Santiago.

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

1.1 Some Known Strategies

In *routing using full-tables*, each vertex v of G knows for each destination u the first edge along some shortest path from v to u (so-called *complete routing table*). When v needs to send a message to u, it just sends the message along the edge stored for destination u. While this approach guarantees routing along a shortest path, it is too expensive for large systems since it requires to store locally $O(n \log \delta)$ bits of global information for an n-vertex graph with maximum degree δ.

Unfortunately, if one insists on a routing via shortest paths, $\Omega(n \log \delta)$ bits is the lower bound on the memory requirements per vertex [30] (this much each vertex needs to know at least). To obtain routing schemes for general graphs that use $o(n)$ of memory at each vertex, one has to abandon the requirement that packets are always delivered via shortest paths, and settle instead for the requirement that packets are routed on paths that are relatively close to shortest. The efficiency of a routing scheme is measured in terms of its additive stretch, called *deviation* (or multiplicative stretch, called *delay*), namely, the maximum surplus (or ratio) between the length of a route, produced by the scheme for a pair of vertices, and the shortest route. There is a tradeoff between the memory requirements of a routing scheme (how much of global information is available locally at a vertex) and the worst case stretch factor it guarantees. Any multiplicative t-stretched routing scheme must use $\Omega(n)$ bits for some vertices in some graphs for $t < 3$ [27] (see also [23]), and $\Omega(n \log n)$ bits for $t < 1.4$ [30]. These lower bounds show that it is not possible to lower memory requirements of a routing scheme for an arbitrary network if it is desirable to route messages along paths close to optimal. Therefore, it is interesting, both from a theoretical and a practical view point, to look for specific routing strategies on graph families with certain topological properties.

One specific way of routing, called *interval routing*, has been introduced in [44] and later generalized in [38]. In this method, the complete routing tables are compressed by grouping the destination addresses which correspond to the same output edge. Then each group is encoded as an interval, so that it is easy to check whether a destination address belongs to the group. This approach requires $O(\delta \log n)$ bits of memory per vertex, where δ is the maximum degree of a vertex of the graph. A graph must satisfy some topological properties in order to support interval routing, especially if one insists on paths close to optimal. Routing schemes for many graph classes were obtained by using interval routing techniques. The classical and most recent results in this field are presented in [25, 26].

Recently, so-called *routing labeling schemes* [40] become very popular. A number of interesting results for general graphs and particular classes of graphs were obtained. These are schemes that label the vertices of a graph with short labels (describing some global topology information) in such a way that given the label of a source vertex and the label of a destination, it is possible to compute efficiently the edge from the source that heads in the direction of the destination. In [24, 48], a shortest path routing scheme for trees with $O(\log^2 n/ \log\log n)$-bit labels is described. For general graphs, the most general result to date is a multiplicative $(4k-5)$-stretched routing labeling scheme that uses labels of size $\tilde{O}(kn^{1/k})$ bits[1] is obtained in [48] for every $k \geq 2$. For planar graphs, a shortest path routing labeling scheme which uses $8n + o(n)$ bits per vertex is developed in [28], and a multiplicative $(1+\epsilon)$-stretched routing labeling scheme for every $\epsilon > 0$ which uses $O(\epsilon^{-1} \log^3 n)$ bits per vertex is developed in [47]. This has been generalized in [1] to graphs excluding a fixed minor with the same stretch and space bounds. Routing in graphs with

[1] Here, $\tilde{O}(f)$ means $O(f$ polylog $n)$.

doubling dimension α has been considered in [2, 12, 45, 46]. It was shown that any graph with doubling dimension α admits a multiplicative $(1+\epsilon)$-stretched routing labeling scheme with labels of size $\epsilon^{-O(\alpha)} \log^2 n$ bits. Recently, the routing result for trees of [24, 48] was used in designing additive $O(1)$-stretched routing labeling schemes with $O(\log^{O(1)} n)$ bit labels for several families of graphs, including chordal graphs, chordal bipartite graphs, circular-arc graphs, AT-free graphs and their generalizations, the graphs with bounded longest induced cycle, the graphs of bounded tree–length, the bounded clique-width graphs, etc. (see [16, 18–20] and papers cited therein).

In wireless networks, the most popular strategy is the *geographic routing* (sometimes called also the *greedy geographic routing*), were each vertex forwards the packet to the neighbor geographically closest to the destination (see survey [31]). 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 [6, 33, 37]. These techniques typically exploit planar subgraphs (e.g., Gabriel graph, Relative Neighborhood graph), and packets traverse faces on such graphs using the well-known right-hand rule.

All earlier papers assumed that vertices are aware of their physical location, an assumption which is often violated in practice for various of reasons (see [21, 35, 42]). In addition, implementations of recovery schemes are either based on non-rigorous heuristics or on complicated planarization procedures. To overcome these shortcomings, recent papers [21, 35, 42] propose routing algorithms which assign virtual coordinates to vertices in a metric space X and forward messages using geographic routing in X. In [42], 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 [21], 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 [21] is a modification of the ℓ_1 norm. Both [21] and [42] 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 [21] and [42], the embedding of [35] guarantees that the geographic routing will always be successful in finding a route to the destination, if such a route exists. Algorithm of [35] 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, [35] gets virtual coordinates for vertices of a graph G by embedding in the hyperbolic plane a spanning tree of G. The proof that this method guaranties delivery is relied only on the fact that the hyperbolic greedy route is no longer than the spanning tree route between two vertices; even more, it could be much shorter as greedy routes take enough short cuts (edges which are not in the spanning tree) to achieve significant saving in stretch. However, although the experimental results of [35] 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.

1.2 Our Approach

Motivated by the work of Robert Kleinberg [35], in this paper, we initiate exploration of the following strategy in advancing in a graph from a source vertex towards a target vertex. Let $G = (V, E)$ be a 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:

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
(break ties arbitrarily).

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 [29, 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 of a longest greedy routing path that can be produced for x and y using this strategy and T. We say that a spanning tree T of a graph G is an *additive r-carcass* for G if $g_{G,T}(x, y) \leq d_G(x, y) + r$ for each ordered pair $x, y \in V$ (in a similar way one can define also a *multiplicative t-carcass* of G). Note that this notion differs from the notion of "remote-spanners" introduced recently in [32].

In this paper, we start investigating 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, and give our preliminary results. We show 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 are easy to construct.

2 Preliminaries

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 the 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., $D_k(v) = \{u \in V : d_G(u, v) \leq k\}$. A set $S \subseteq V$ is a *clique* (an *independent set*) of G if all vertices of S are pairwise adjacent (respectively, nonadjacent) in G. A clique of G is *maximal* if it is not contained in any other clique of G.

Next we recall the definitions of special graph classes mentioned in this paper (see survey [10]). A graph is *chordal* if it does not have any induced cycle of length greater than 3. A *p-sun* $(p \geq 3)$ is a chordal graph on $2p$ vertices whose vertex set can be partitioned into two sets, $U = \{u_0, \ldots, u_{p-1}\}$ and $W = \{w_0, \ldots, w_{p-1}\}$, such that W is an independent set, U is a clique, and every w_i is adjacent only to u_i and u_{i+1} (mod p). A chordal graph having no induced subgraphs isomorphic to p-suns (for any $p \geq 3$) is called a *strongly chordal graph*. A chordal graph having no induced subgraphs isomorphic to 3-sun is called a *3-sun-free chordal graph*. A graph is *chordal bipartite* if it is bipartite and has no induced cycles of length greater than 4. *A dually chordal graph* is the intersection graph of the maximal cliques of a chordal graph (see [10, 9] for many equivalent definitions of dually chordal graphs and strongly chordal graphs). A graph is *interval* if it is the intersection graph of intervals of a line. It is known that interval graphs are strongly chordal and strongly chordal graphs are dually chordal (see [10, 9]). A graph G is *distance-hereditary* if every induced path of G is shortest (see [10] for many equivalent definitions of distance-hereditary graphs). The *k-trees* are defined recursively: a clique of size k (denoted by K_k) is a k-tree; if G is a k-tree, then a graph obtained from G by adding a new vertex v adjacent to all vertices of some clique K_k of G is a k-tree. It is known (see [10]) that all k-trees are chordal graphs and that maximal cliques of a k-tree have size at most $k + 1$.

Let now $G = (V, E)$ be a graph and T be a spanning tree of G. In what follows, we will use the following notations. For vertices v and u from V, denote by vTu the (unique) path of T connecting vertices v and u. For a source vertex x and a target vertex y in G, denote by $R_{G,T}(x, y)$ a greedy routing path obtained for x and y by using tree T and the strategy described in Subsection 1.2. Clearly, for the same pair of vertices x and y, breaking ties differently, different greedy routing paths $R_{G,T}(x, y)$ can be produced. Denote, as before, by $g_{G,T}(x, y)$, the length of a longest greedy routing path that can be produced for x and y. If no confusion can arise, we will omit indexes G and T, i.e., use $R(x, y)$ and $g(x, y)$ instead of $R_{G,T}(x, y)$ and $g_{G,T}(x, y)$.

For $r \geq 0$ and $t \geq 1$, a spanning tree T of a graph G is called an *additive r-carcass* (a *multiplicative t-carcass*) for G if $g_{G,T}(x, y) \leq d_G(x, y) + r$ (respectively, $g_{G,T}(x, y) \leq t\ d_G(x, y)$) for each ordered pair $x, y \in V$.

Let x^* be the neighbor of x in $R_{G,T}(x, y)$ and x' be the neighbor of x in xTy. Since both x^* and x' are in $N_G(x)$ and $d_T(x', y) = d_T(x, y) - 1$, according to our strategy $d_T(x^*, y) \leq d_T(x', y) = d_T(x, y) - 1$ must hold. Furthermore, any subpath of a greedy routing path $R_{G,T}(x, y)$ containing y is a greedy routing path to y as well. Hence, one can conclude, by induction, that the length of any greedy routing path $R_{G,T}(x, y)$ never exceeds $d_T(x, y)$. It is clear also, that a greedy routing path $R_{G,T}(x, y) := (x := x_0, x_1, x_2, \ldots, y := x_\ell)$ cannot have a chord $x_i x_j \in E$ with $j > i + 1$ (since $d_T(x_{i+1}, y) > d_T(x_j, y)$), i.e., any greedy routing path is an induced path. Thus, we have the following.

Observation 1. *Let G be an arbitrary graph and T be its arbitrary spanning tree. Then, for any vertices x, y of G,*

(a) $g_{G,T}(x, y) \leq d_T(x, y)$,
(b) any greedy routing path $R_{G,T}(x, y)$ is an induced path of G,
(c) a tale of any greedy routing path is a greedy routing path.

Since in distance-hereditary graphs each induced path is a shortest path, by Observation 1(b), we conclude.

Corollary 1. *Any spanning tree of a distance-hereditary graph G is a 0-carcass of G.*

Papers [14, 5, 43] introduce and investigate a family of graphs, denoted by $DH(k, +)$, which generalizes in a parameterized way the class of distance-hereditary graphs. Let $k \geq 0$ be an integer. A graph G belongs to the family $DH(k, +)$ if and only if the length of any induced (x, y)-path of G is at most $d_G(x, y) + k$ for any x, y in G. Hence, we have.

Corollary 2. *Any spanning tree of a graph G from $DH(k, +)$ is an additive k-carcass of G.*

There are well-known notions of additive tree r-spanners and multiplicative tree t-spanners. For $r \geq 0$ and $t \geq 1$, a spanning tree T of a graph G is called an *additive tree r-spanner* (a *multiplicative tree t-spanner*) of G if $d_T(x, y) \leq d_G(x, y) + r$ (respectively, $d_T(x, y) \leq t\ d_G(x, y)$) for each pair $x, y \in V$ [11]. By Observation 1(a), we obtain.

Corollary 3. *Any additive tree r-spanner (multiplicative tree t-spanner) of a graph G is an additive r-carcass (multiplicative t-carcass) of G.*

Note that the converse of Corollary 3 is not generally true. As we will see in next sections, there are many families of graphs which do not admit any tree r-spanners (additive as well as multiplicative) for any constant r, yet they admit very good carcasses. For example, there is no constant r such that any 2-tree or any chordal bipartite graph has a tree r-spanner (additive or multiplicative), but both these families of graphs admit additive 4-carcasses (see Section 5 and Section 6 for details).

In what follows, in a rooted tree T, by $f(v)$ we will denote the father of a vertex v.

3 Rectilinear Grids and Hypercubes

In this section we show that the rectilinear grids and the hypercubes admit additive 0-carcasses.

Let G be a $m \times n$ rectilinear grid with m rows and n columns. The rows are numbered from 1 to m and the columns are numbered from 1 to n. A cell (i, j) denotes the cell in row i and column j in the grid, for $1 \leq i \leq m$ and $1 \leq j \leq n$. Assume that it is naturally embedded into the plane such that all inner faces of G are squares (see Fig. 1) and $(1, 1)$ is the left upper corner.

First we notice that G does not admit any good tree spanner. For this, consider an arbitrary spanning tree T of G, and assume that m and n are odd integers and $m \leq n$. Since T is a planar graph with only the outer face, we can connect by a Jordan curve $\mathcal{C}$ a point of the plane inside the central square of G with a point in the outer face of G without intersecting the tree T. Let R be the first square of G crossed by $\mathcal{C}$ and x and y be two opposite vertices of R (see Fig. 1(a) for an illustration). Clearly, for x and y, $d_T(x, y) \geq m + 1$ holds, while $d_G(x, y) = 2$. Here, we considered nonadjacent vertices of G since adjacent vertices are of no interest in our greedy routing. Thus, there are no good tree spanners for rectilinear grids. On the other hand, G admits an additive 0-carcass. Consider a Hamiltonian path of G depicted on Fig. 1(b), called *column-wise Hamiltonian path*. This path is an additive 0-carcass of G. We leave verification of this fact to the reader. We prove instead the following result.

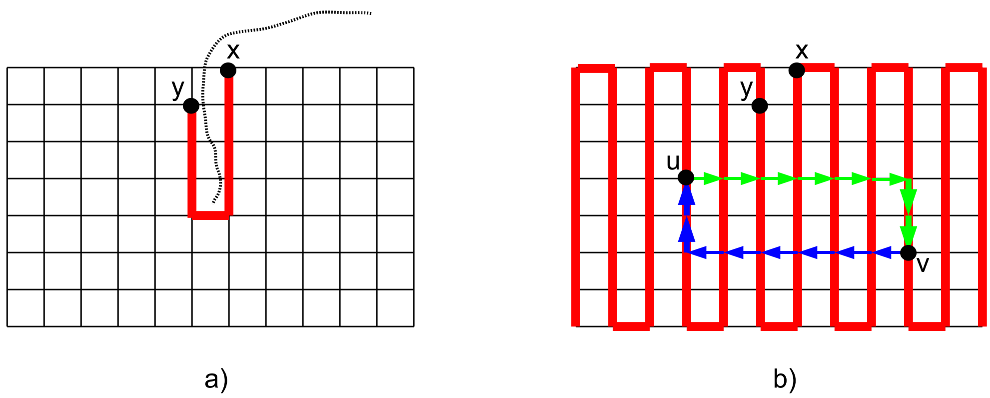

Fig. 1. Rectilinear grids do not admit any (additive or multiplicative) tree r-spanners with a constant r, but have additive 0-carcasses.

Theorem 1. *Let G be the $m \times n$ grid and T be its additive 0-carcass. Then T is the column-wise Hamiltonian path or the row-wise Hamiltonian path, or it is two parallel paths connected by an edge if* $\min\{m, n\} = 2$.

Proof. Let us assume that T is a 0-carcass for the grid G with m rows and n columns. Remember that we label the vertices of G as (i, j) where $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$. We assume that the $(1, 1)$ vertex is in the upper left corner and (m, n) vertex is in the lower right corner.

Let v be a leaf of T. Let us assume that $v = (i, j)$ with $1 \leq i \leq m$ and that $a = (i, j + 1)$ is its father. Then, $j = 1$ as, otherwise, the greedy routing from a to $(i, j - 1)$ will not choose v as the next vertex since the path on T between v and $(i, j - 1)$ passes through a. Let k be the smallest integer such that the vertex (i, k) has a neighbor in T in the same column. Let us assume that this neighbor is $(i - 1, k)$. As $(i, 1)$ is a leaf and we are assuming that its father is $(i, 2)$, we know that $k \geq 2$. If $i < m$, then the greedy routing from $(i-1, k-1)$ to $(i+1, k-1)$ may not choose $(i, k-1)$ as next vertex since the distance in T from $(i, k - 1)$ to $(i + 1, k - 1)$ is at least the distance in T from $(i - 1, k)$ to $(i + 1, k - 1)$. Therefore, $i = m$. By an analogous argument, if the neighbor in T of (i, k) is the vertex $(i + 1, k)$, then we can prove that $i = 1$. Therefore, we have the following property.

(1) Any leaf of T lays in a corner of the grid.

Furthermore, the following property holds.

(2) If $(i, j), \ldots, (i, j + p)$ is a subpath of T and $(i, j + p)(i - 1, j + p)$ is an edge of T, then $(i - 1, j + 1), \ldots, (i - 1, j + p)$ is a subpath of T. (Analogously, if $(i, j), \ldots, (i, j + p)$ is a subpath of T and $(i, j)(i - 1, j)$ is an edge of T, then $(i - 1, j), \ldots, (i - 1, j + p - 1)$ is a subpath of T.)

If this is not the case, then the distance in T between $(i - 1, j + 1)$ and $(i - 1, j + p)$ is at least their distance in G plus two. As this latter quantity equals the distance in T between (i, j) and $(i - 1, j + p)$, the greedy routing strategy could choose (i, j) as the next vertex when routing from $(i - 1, j)$ to $(i - 1, j + p)$. Hence, a routing path from $(i - 1, j)$ to $(i - 1, j + p)$ will be longer than a shortest path.

As a corollary we obtain one more property.

(3) If $(i, j), \ldots, (i, j + p)$ $(p \geq 2)$ is a subpath of T, then at most one edge $(i, j + t)(i - 1, j + t)$ is an edge of T for $0 \leq t \leq p$.

Let us assume that T is not a Hamiltonian path. Let $v = (i, j)$ be a vertex having at least three neighbors in T and let us assume that these neighbors are $(i + 1, j), (i - 1, j)$ and $(i, j + 1)$. If $(i, j + 1)$ is adjacent in T neither to $(i - 1, j + 1)$ nor to $(i + 1, j + 1)$, then either there is an edge $(i, l)(i + 1, l)$ or $(i - 1, l)(i, l)$ in T with $l > j + 1$. Note that vertex $(i, j + 1)$ cannot be a leaf of T by (1). Moreover, any leaf of T is in a corner of the grid. But, the existence of either edge also is a contradiction with property (3).

Hence, we can assume that $(i, j+1)(i - 1, j + 1)$ is an edge of T. As the edge $(i-1, j+1)(i-1, j)$ does not belong to T, by property (2), we deduce that edge $(i, j + 1)(i, j + 2)$ does not belong to T either. As the distance in T between $(i+1, j)$ and $(i-1, j+1)$ is three, the edge $(i, j+1)(i+1, j+1)$ must belong to T as, otherwise, one can choose $(i - 1, j)$ as the next vertex when greedily routing from $(i - 1, j + 1)$ to $(i + 1, j + 1)$. By property (3), we know that the subpath of T starting with $(i, j)(i+1, j)$ cannot go right when it leaves column j for the first time. Hence, either it never leaves column j, in which case $j = 1$ or it does and then $j > 1$. As a similar argument can be applied to the subpath of T starting with $(i, j)(i-1, j)$ we conclude, by property (3), that $j = 1$. Similarly, we conclude that $n = 2$ by arguing as before with the subpaths of T starting at $(i, j + 1)$ and moving along column $j + 1$. Therefore, if T is not Hamiltonian, then $\min\{n, m\} = 2$ and T consists of two parallel paths joined by an edge.

To finish the proof, let us assume that $\min\{n, m\} > 2$ and let v be one of the two ends of the Hamiltonian path T. Then we already know that v is a corner of G. Let us say that $v = (1, 1)$ and that the father of v is $(2, 1)$. Let $(k, 1)$ be the first time the path T enters column 2. By property (2), we know that T moves up at least until it reaches vertex $(2, 2)$.

Then $(k-1,2)$ is contained in the subpath of T from $(k,1)$ to any vertex (i,j) with $j>1$ or $i>k$. This implies that $(k-1,2)$ can be the next vertex when greedily routing from $(k-1,1)$ to any vertex (i,j) with $j>1$ or $i>k$. As the existence of vertex $(k+1,1)$ would lead to a contradiction, we conclude that $k=m$.

As T is a path, $(2,2)$ is not a leaf of T and $(1,1)$ is a leaf of T, then the edges $(2,2)(1,2)$ and $(1,2)(1,3)$ belong to T. We can apply again property (2) to conclude that T continues from $(1,3)$ in column 3 at least until it reaches vertex $(m-1,3)$. It is clear that unless T had reached column n we can apply the same argument as before to show that T moves in G column by column. $\square$

Now we turn to the hypercubes. Let $H_q=(V,E)$ be the q-dimensional hypercube whose vertices are binary words of length q and two vertices are adjacent if they differ in exactly one letter. Let $a\in\{0,1\}$ and $i\in\{1,\ldots,q\}$. Let $H_q^{a,i}$ be a subgraph of H_q induced by vertices having letter a in the position i. Then, $H':=H_q^{a,i}$ is isomorphic to the $(q-1)$-dimensional hypercube and $d_{H_q}(x,y)=d_{H'}(x,y)$, whenever the letter in the position i of x and y is a.

Let T be the *Gray-Hamiltonian* path of H_q defined recursively as follows. If $x_1,\ldots,x_{2^{q-1}}$ is the Gray-Hamiltonian path for H_{q-1}, then T is given by T_0T_1, where $T_0=x_10,\ldots,x_{2^{q-1}}0$ and $T_1=x_{2^{q-1}}1,\ldots,x_11$. By applying two steps of previous recursion, it is clear that T can be decomposed into four consecutive subpaths $T=T_{00}T_{10}T_{11}T_{01}$, where the subpath T_w contains all the vertices of H_q ending with w. Notice that T_0 is the Gray-Hamiltonian path for $H_q^{0,q}$, T_1 is the reverse of the Gray-Hamiltonian path for $H_q^{1,q}$ and $T_{10}T_{11}$ is the Gray-Hamiltonian path for the hypercube $H_q^{1,q-1}$.

By using induction on q, we prove that $g(x,y):=g_{G,T}(x,y)=d_G(x,y)$, where $G=H_q$ and T is the Gray-Hamiltonian path of H_q. When x and y belong to T_0 (resp. T_1), conclusion is obtained by applying induction hypothesis to vertices x and y in the hypercube $H_q^{0,q}$ (resp. $H_q^{1,q}$) with the Gray-Hamiltonian path T_0 (resp. T_1). Similarly, when x belongs to T_{10} and y belongs to T_{11}, we can apply induction in the hypercube $H_q^{1,q-1}$ with the Gray-Hamiltonian path $T_{10}T_{11}$.

For the remaining cases, let x^* denote the vertex next to x in a greedy routing path $R_{G,T}(x,y)$. If x^* and y belong to T_1, and x belongs to T_0, then $d_G(x^*,y)=d_G(x,y)-1$, since x^* and y agree in their last letters. By applying induction to x^* and y using T_1, we get that $g(x^*,y)=d_{H_q^{1,q}}(x^*,y)=d_G(x^*,y)$. Hence, $g(x,y)=1+g(x^*,y)=1+d_G(x^*,y)=d_G(x,y)$. Let us now consider the case when x and x^* belong to T_0, and y belongs to T_1. As x has a neighbor in T_1 and we are assuming that x^* belongs to T_0, vertex y must belong to T_{11}. We have already considered the case when x belongs to T_{10}. Hence, let us assume that x belongs to T_{00}. As x has a neighbor in T_{10}, vertex x^* must belong to T_{10}. Since in this case $d_G(x^*,y)=d_G(x,y)-1$, we can conclude as before, by applying induction to x^* and y in $H_q^{1,q-1}$ with the Gray-Hamiltonian path $T_{10}T_{11}$.

So, we can state the following theorem.

Theorem 2. *Every rectilinear grid and every hypercube admits an additive 0-carcass (which is a Hamiltonian path) constructible in linear time.*

4 Locally Connected Spanning Trees are Additive 0-Carcasses: Dually Chordal Graphs

In this section, we show that every dually chordal graph $G=(V,E)$ admits an additive 0-carcass constructible in linear time. Recall that every dually chordal graph has an additive tree 3-spanner and there are dually chordal graphs without any additive tree 2-spanners (see [8]). Clearly, those additive tree 3-spanners are additive 3-carcasses, but it is not hard to see that they are not necessarily additive 0-carcasses (see, e.g., Fig. 2).

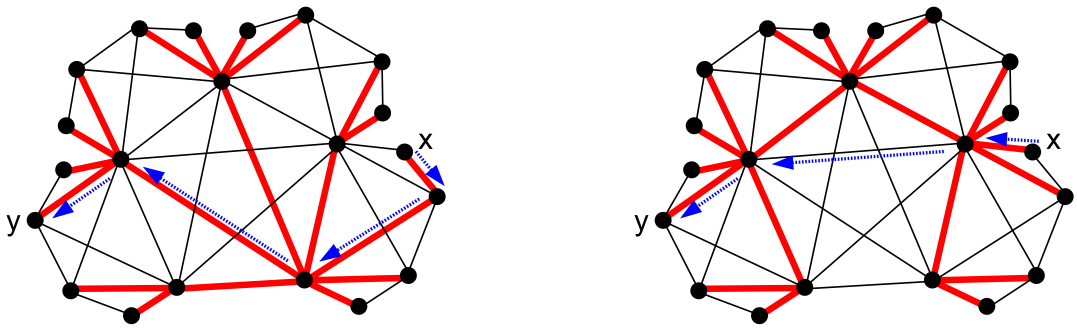

Fig. 2. A dually chordal graph with an additive tree 3-spanner (on the left) and an additive 0-carcass (on the right). This dually chordal graph does not have any additive tree 2-spanner. A greedy routing path from x to y with respect to the corresponding tree is shown on both pictures.

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). See the right picture on Fig. 2 for an example of a locally connected spanning tree.

Theorem 3. *If T is a locally connected spanning tree of a graph G, then T is an additive 0-carcass of G.*

Proof. Assume that we want to route from a source vertex x to a target vertex y in G. Let v be an arbitrary vertex of G and v^* be a vertex from $N_G[v]$ closest to y in T. Since $T \cap N_G[v]$ is a connected subgraph of T, for each vertex $v \in V$ such a neighbor v^* is unique (any subtree of a tree has only one vertex closest in T to a given vertex y). Moreover, $v^* \neq v$, unless $v = y$. In what follows, we will assume that the tree T is rooted at vertex y.

Claim 1. *For any vertex $v \in V$, the vertex v^* belongs to a shortest path of G connecting v and y.*

Proof. We prove by induction on $d_G(v, y)$. If $d_G(v, y) \leq 1$, then $v^* = y$ and therefore v^* belongs to any shortest path between v and y. So, assume that $d_G(v, y) \geq 2$. Consider a shortest path $P(v, y) := (v, a, b, \ldots, y)$ in G connecting v and y, where a and b are the first two (after v) vertices of this path. They exist since $d_G(v, y) \geq 2$.

To obtain the conclusion, we prove that v^* and b are adjacent. For sake of contradiction, let us assume that they are not adjacent. Since $a, b^* \in N_G(b)$ and $T \cap N_G[b]$ is a connected subgraph of T, we get that v^* is not on path aTb^*. We also know that $a \in N_G[v] \cap N_G[b]$ and therefore both v^* and b^* are ancestors in T of a (recall that we have rooted T at y). Hence, b^* is on the path aTv^*. As $a, v^* \in N_G(v)$ and $T \cap N_G[v]$ is a connected subgraph of T, we get that v and b^* are adjacent. By induction, b^* belongs to a shortest path of G between b and y, which leads to the following contradiction: $d_G(v, y) \leq 1 + d_G(b^*, y) = d_G(b, y) < d_G(v, y)$. □(of Claim)

Now we prove by induction on $d_T(x, y)$ that $g(x, y) = d_G(x, y)$. Indeed, $g(x, y) = 1 + g(x^*, y)$ and, by induction, $g(x^*, y) = d_G(x^*, y)$ as $d_T(x, y) > d_T(x^*, y)$. By Claim 1, we conclude $g(x, y) = 1 + d_G(x^*, y) = d_G(x, y)$. □

It has been shown in [9] that the graphs admitting locally connected spanning trees are precisely the dually chordal graphs. Furthermore, [9] 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 corollary.

Corollary 4. *Every dually chordal graph admits an additive 0-carcass constructible in linear time. In particular, any strongly chordal graph (any interval graph) admits an additive 0-carcass constructible in linear time.*

Note that, in [7, 9], 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.

5 Additive 4-Carcasses for Chordal Bipartite Graphs

In this section, we show that every chordal bipartite graph $G = (V, E)$ admits an additive 4-carcass constructible in linear time. Recall that chordal bipartite graphs do not have any tree r-spanners (additive or multiplicative) with a constant r (see, e.g., [13]).

Let $G = (V, E)$ be a chordal bipartite graph. Consider an arbitrary vertex $s \in V$ and let $L_i := \{v \in V : d_G(v, s) = i\}$, $i = 0, 1, 2, \ldots$ be the layering of G with respect to s. For a vertex $u \in L_j$, denote by $N{\downarrow}_G(u) := N_G(u) \cap L_{j-1}$ and $deg{\downarrow}(u) := |N{\downarrow}_G(u)|$ the down-neighborhood and the down-degree of u, respectively. We assume that $deg{\downarrow}(s) := 0$.

First we state an auxiliary lemma which will be useful in our construction of a carcass and its analysis.

Lemma 1. *If a vertex $v \in L_i$ has two neighbors x and y in L_{i-1}, then the down-neighborhoods of x and y are comparable, i.e., $N{\downarrow}_G(x) \subseteq N{\downarrow}_G(y)$ or $N{\downarrow}_G(y) \subseteq N{\downarrow}_G(x)$.*

Proof. Assume that the down-neighborhoods of x and y are not comparable and let $x' \in N{\downarrow}_G(x) \setminus N{\downarrow}_G(y)$ and $y' \in N{\downarrow}_G(y) \setminus N{\downarrow}_G(x)$. Since G is bipartite, $xy, x'y' \notin E$. Consider an induced path $P(x', y')$ connecting vertices x' and y' in G and using only vertices from layers $L_j, j < i - 2$ as the inner vertices. Clearly, such a path exists. Note now that a cycle of G formed by $P(x', y')$ and edges $x'x, xv, vy, yy' \in E$ is induced and has length at least 6. The latter is in contradiction with G being a chordal bipartite graph. □

A carcass T for G we construct by connecting every vertex $v \in L_i$, $i > 0$, by an edge to its neighbor $f(v) \in L_{i-1}$ that has the minimum down-degree $deg{\downarrow}(f(v))$ among all neighbors of v in L_{i-1}. Vertex $f(v)$ will be called *the father of* v in T. Clearly, given a graph G, such a tree T can be constructed in linear time. It turns out that T is an additive 4-carcass of G.

Theorem 4. *Every chordal bipartite graph $G = (V, E)$ admits an additive 4-carcass T constructible in linear time.*

Proof. We prove that the tree T constructed above is an additive 4-carcass of G. Assume that we want to route from a vertex $x \in L_i$ to a vertex $y \in L_j$. Note that, if x is an ancestor of y or y is an ancestor of x in T, then $g(x, y) \le d_T(x, y) = d_G(x, y)$ by Observation 1(a) and since T is a shortest path tree (SP-tree) of G. Hence, in what follows, we will assume that neither $x \in sTy$ nor $y \in sTx$. We distinguish between two cases: x is adjacent in G to a vertex x' of sTy (we say x sees sTy) or it is not adjacent in G to any vertex of sTy (we say x misses sTy).

Case 1: *x sees sTy.*

Let x' be a vertex of $N_G(x)$ and sTy, and x^* be a vertex from $N_G(x)$ closest to y in T. We have $g(x, y) = 1 + g(x^*, y) \le 1 + d_T(x^*, y) \le 1 + d_T(x', y) = 1 + d_G(x', y)$ by choice of x^*, Observation 1(a) and since x' is an ancestor of y in SP-tree T of G. Since G is bipartite, x' belongs to L_{i+1} or to L_{i-1}. If $x' \in L_{i+1}$, then $d_G(x', y) = j - i - 1$ and therefore $g(x, y) \le 1 + d_G(x', y) = j - i \le d_G(x, y)$

since $d_G(x,y) \geq j-i$. If $x' \in L_{i-1}$, then $d_G(x',y) = j-i+1$ and therefore $g(x,y) \leq 1 + d_G(x',y) = j-i+2 \leq d_G(x,y)+2$. Thus, in this case, we get $g(x,y) - d_G(x,y) \leq 2$.

Case 2: *x misses sTy.*

Let again x^* be a vertex from $N_G(x)$ closest to y in T. We claim that, in this case, $x^* \in L_{i-1}$. Assume, by way of contradiction, that $x^* \in L_{i+1}$. We have $d_T(f(x),y) \geq d_T(x^*,y)$ (by choice of x^*) and $d_T(f(x),y) < d_T(x,y)$, $d_T(f(x^*),y) < d_T(x^*,y)$ (since x, x^* are not ancestors of y in T). Therefore, $f(x^*) \neq x$ and, by the construction of T, $deg{\downarrow}(f(x^*)) \leq deg{\downarrow}(x)$. By Lemma 1, $N{\downarrow}_G(f(x^*)) \subseteq N{\downarrow}_G(x)$ and therefore vertex x must be adjacent in G to $f(f(x^*))$. Since x misses sTy, vertex $f(f(x^*))$ cannot belong to sTy. Consequently, the nearest common ancestor of y and x^* in T is an ancestor of $f(f(x^*))$ as well. But then, $d_T(f(f(x^*)),y) = d_T(x^*,y) - 2$. The latter combined with $f(f(x^*)) \in N_G(x)$ is in contradiction with the choice of x^*.

Thus, we proved that if $x \in L_i$ misses sTy then x^* must belong to L_{i-1}. Consider now a greedy routing path $R(x,y) := (x, x^*, \ldots, u, v, y', \ldots, y)$ and assume that $v \in R(x,y)$ is the first vertex of this path (when moving from x to y) which sees sTy. Clearly, v exists and $v \neq x$. Let $v \in L_k$. By the above proof, subpath $(x, x^*, \ldots, u, v)$ of the path $R(x,y)$ has length $i-k$. Then, $g(x,y) = d_G(x,v) + g(v,y) = i-k+g(v,y)$. Let v' be the vertex from $N_G(v)$ and sTy closest to y. By the proof of Case 1, we have also that $g(v,y)$ equals $j-k$, if $v' \in L_{k+1}$, and is at most $j-k+2$, if $v' \in L_{k-1}$. Thus,

$$g(x,y) \leq \begin{cases} i+j-2k, & \text{if } v' \in L_{k+1}; \\ i+j-2k+2, & \text{if } v' \in L_{k-1}. \end{cases} \tag{1}$$

Consider now in G a shortest (x,y)-path $P(x,y)$, and let z be a vertex of this path closest to s and assume $z \in L_q$ (see Fig. 3 for an illustration). If $q \leq k+1$ then $d_G(x,y) = d_G(x,z) + d_G(z,y) \geq i-q+j-q = i+j-2q \geq i+j-2k-2 \geq g(x,y)-4$, and we are done. Therefore, we may assume that $q \geq k+2$.

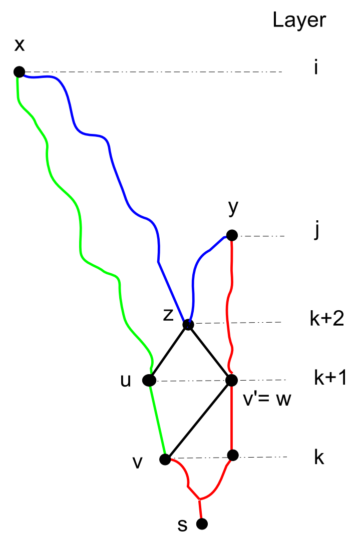

Fig. 3. Paths $P(x,y)$ (in blue), sTy, sTv (both in red) and $(x, x^*, \ldots, u, v)$ (in green).

We have from the proof above that vertex u of path $(x, x^*, \ldots, u, v)$ belongs to L_{k+1}. Let w be the vertex of sTy from layer L_{k+1}. In a subgraph of G induced by vertices of $(x, x^*, \ldots, u)$, $P(x,y)$ and those vertices of sTy that are at distance at least $k+1$ from s, choose a shortest path $Above(u,w)$ connecting u and w. Analogously, in a subgraph of G induced by vertices of paths $sTv \cup vu$ and

sTw, choose a shortest path $Below(u, w)$ connecting u and w. Since G cannot have any induced cycle of length greater than 4 and no edge exists in G between vertices of $Above(u, w) \setminus \{w, u\}$ and vertices of $Below(u, w) \setminus \{w, u\}$, both paths $Above(u, w)$ and $Below(u, w)$ must have length 2. Let $Above(u, w) = (u, a, w)$ and $Below(u, w) = (u, b, w)$.

Since v is the first vertex of path $R(x, y)$ (when moving from x to y) that sees sTy, vertex a cannot be in sTy or in $(x, x^*, \dots, u)$ and hence must belong to $P(x, y)$, i.e., $q = k + 2$. By similar argument, vertex b must coincide with v, i.e., $v' = w$. We have $v' \in L_{k+1}$ and, therefore, $g(x, y) \leq i + j - 2k$. On the other hand, $d_G(x, y) = d_G(x, a) + d_G(a, y) \geq i - k - 2 + j - k - 2 = i + j - 2k - 4$. Thus, $d_G(x, y) \geq g(x, y) - 4$. □

6 Additive Carcasses for Chordal Graphs and *k*-Trees

In this section, we show that every chordal graph admits an additive ($\omega(G)+1$)-carcass constructible in linear time. Here $\omega(G)$ is the size of a maximum clique of G. In particular, every k-tree admits an additive $(k+2)$-carcass and any 2-tree admits an additive 4-carcass. Recall that 2-trees do not have any tree r-spanners (additive or multiplicative) with a constant r (see, e.g., [36]). We also show that every 3-sun-free chordal graph admits an additive 2-carcass.

We will need the following lemmata from [15] and [49].

Lemma 2. [15] *If vertices a and b of a disk $D_k(s)$ of a chordal graph 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.*

Lemma 3. [49] *If vertices of a clique of a chordal graph G are equidistant from a vertex of G, then this clique is not maximal.*

Lemma 4. [49] *If two adjacent vertices a and b of a chordal graph G are equidistant from a vertex v, then there must exist a vertex c in G which is adjacent to both a and b and is at distance $d_G(a, v) - 1 = d_G(b, v) - 1$ from v.*

Theorem 5. *Any shortest path tree of a chordal graph G is an additive $(\omega(G) + 1)$-carcass of G. Here $\omega(G)$ is the size of a maximum clique of G.*

Proof. Let $G = (V, E)$ be a chordal graph and $s \in V$ be an arbitrary vertex of G. Construct a shortest path tree (SP-tree) T of G rooted at s. We prove that T is an additive $(\omega(G) + 1)$-carcass of G. Let, as before, $L_i := \{v \in V : d_G(v, s) = i\}$, $i = 0, 1, 2, \dots$ be the layering of G with respect to s.

Assume that we want to route from a vertex $x \in L_i$ to a vertex $y \in L_j$. By the proof of Theorem 4, we may assume that neither $x \in sTy$ nor $y \in sTx$. Moreover, we may also assume that x misses sTy, since, otherwise, $g(x, y) \leq d_G(x, y) + 2$ and we are done.

Consider a greedy routing path $R(x, y) := (x_0 := x, x_1, \dots, x_{q-1}, v = x_q, w, \dots, y)$ and assume that $v \in R(x, y)$ is the first vertex of this path (when moving from x to y) which sees sTy. Clearly, v exists, $v \neq x$ and v is not an ancestor of y. Let $v \in L_k$.

According to our routing strategy, for each index $h \in \{0, \dots, q-1\}$, $x_h x_{h+1} \in E$ and $d_T(x_h, y) > d_T(x_{h+1}, y)$. We claim that, for every $h \in \{0, \dots, q-1\}$, $d_G(x_h, s) \geq d_G(x_{h+1}, s)$. Assume not, and consider a vertex $x_{h'}$ with $d_G(x_{h'}, s) = d_G(x_{h'+1}, s) - 1$. Since vertex $x_{h'}$ and the father $f(x_{h'+1})$ of $x_{h'+1}$ in T are connected outside of $D_\ell(s)$ ($\ell := d_G(x_{h'}, s)$) by a path, by Lemma 2, $x_{h'}$ and $f(x_{h'+1})$ must be adjacent. But, that is in contradiction with the choice of $x_{h'+1}$ because $x_{h'+1}$ is a vertex of $N_G(x_{h'})$ closest to y in T, yet $f(x_{h'+1}) \in N_G(x_{h'})$ and $d_T(x_{h'+1}, y) > d_T(f(x_{h'+1}), y)$ (note that $x_{h'+1}$ is not an ancestor of y in T and hence $d_T(x_{h'+1}, y) > d_T(f(x_{h'+1}), y)$). So, the claim is true. In particular, we have $d_G(x_h, s) \geq d_G(x_q, s) = d_G(v, s) = k$ for every $h \in \{0, \dots, q - 1\}$.

Denote by y_h the nearest common ancestor in T of x_h and y. From $d_T(x_h, y) > d_T(x_{h+1}, y)$ we can deduce that y_h coincides with y_{h+1} only if $d_T(x_h, s) = d_T(x_{h+1}, s) + 1$, i.e., only if $d_G(x_h, s) = d_G(x_{h+1}, s) + 1$. Furthermore, if y_h and y_{h+1} are different, then $d_T(y_h, s) < d_T(y_{h+1}, s)$ must hold. Thus, all vertices $y_0, y_1, \dots, y_{q-1}, y_q$ are at distance at most $d_T(y_q, s) \le k-1$ from s (recall that $v \in L_k$ and, since v is not an ancestor of y in T, $d_T(y_q, s) \le k-1$).

We claim now that the set $Y := \{y_0, y_1, \dots, y_{q-1}, y_q\}$ has cardinality at most $\omega(G)-1$. Assume, by way of contradiction, that there are $\omega(G)$ distinct vertices in Y, say $y_{h_1}, y_{h_2}, \dots, y_{h_{\omega(G)}}$. Let $a_{h_\mu} \in L_{k-1} \cap sTx_{h_\mu}$, $\mu = 1, 2, \dots, \omega(G)$. Vertices $\{a_{h_\mu} : \mu = 1, 2, \dots, \omega(G)\}$ are pairwise different and must be pairwise adjacent as each pair can be connected outside of $D_{k-1}(s)$ by a path (e.g., a_{h_1} and a_{h_2} are connected by $a_{h_1}Tx_{h_1}$ concatenated with a subpath of $R(x, y)$ between x_{h_1} and x_{h_2} concatenated with $x_{h_2}Ta_{h_2}$). Since vertices of a clique $\{a_{h_1}, a_{h_2}, \dots, a_{h_{\omega(G)}}\}$ are equidistant from s in G, by Lemma 3, this clique is not maximal. As G cannot have any clique of size greater than $\omega(G)$, the claim is proved by contradiction.

We have proved above that $d_G(x_h, s) \ge d_G(x_{h+1}, s) \ge k$ holds for every $h \in \{0, \dots, q-1\}$. Moreover, y_h coincides with y_{h+1} only if $d_G(x_h, s) = d_G(x_{h+1}, s)+1$, and if y_h and y_{h+1} are different then $d_T(y_h, s) < d_T(y_{h+1}, s)$. Since the cardinality of Y is at most $\omega(G) - 1$, the subpath $(x_0 := x, x_1, \dots, x_{q-1}, v = x_q)$ of path $R(x, y)$ can have at most $\omega(G)-2$ horizontal edges (an edge $ab \in E$ is called *horizontal* if $d_G(a, s) = d_G(b, s)$). Hence, the length of path $(x_0 := x, x_1, \dots, x_{q-1}, v = x_q)$ is at most $i - k + \omega(G) - 2$, and therefore $g(x, y) \le i - k + \omega(G) - 2 + g(v, y)$. Let v' be the vertex from $N_G(v)$ and sTy closest to y. Since v' is an ancestor of y, we get $g(v, y) = 1 + g(w, y) \le 1 + d_T(w, y) \le 1 + d_T(v', y) \le 1 + j - k + 1 = j - k + 2$ (furthermore, $g(v, y) \le j - k + 1$ if $v' \in L_{k+1} \cup L_k$). Combining two inequalities, we get

$$g(x, y) \le \begin{cases} i + j - 2k + \omega(G), & \text{if } v' \in L_{k-1}; \\ i + j - 2k + \omega(G) - 1, & \text{otherwise.} \end{cases} \tag{2}$$

Consider now in G a shortest (x, y)-path $P(x, y)$, and let z be a vertex of this path closest to s. If $d_G(z, s) \le k$ then $d_G(x, y) = d_G(x, z) + d_G(z, y) \ge i + j - 2k \ge g(x, y) - \omega(G)$, and we are done. Therefore, assume that $d_G(z, s) \ge k + 1$. Let y' be the vertex of sTy from layer L_ℓ, where $\ell = d_G(z, s) - 1$. Consider also the vertex x_h of path $(x_0 := x, x_1, \dots, x_{q-1}, v = x_q)$ which is at distance ℓ from s and has the smallest index $h \in \{1, 2, \dots, q\}$. These vertices x_h and y' are connected in G by a path outside $D_\ell(s)$ (by $y'Ty$ concatenated with $P(x, y)$ concatenated with $(x_0 := x, x_1, \dots, x_{h-1}, x_h)$). Hence, by Lemma 2, x_h and y' must be adjacent in G. As v is the first vertex of path $R(x, y)$ (when moving from x to y) that sees sTy, necessarily, $v = x_h$, i.e., $d_G(z, s) = k + 1$. Furthermore, v' must belong to L_k or L_{k+1}. Consequently, $d_G(x, y) \ge i - k - 1 + j - k - 1 = i + j - 2k - 2$ and $g(x, y) \le i + j - 2k + \omega(G) - 1$. Thus, $g(x, y) \le d_G(x, y) + 2 + \omega(G) - 1 = d_G(x, y) + \omega(G) + 1$. □

Since k-trees are chordal graphs with the size of a maximum clique is at most $k+1$, we conclude.

Corollary 5. *Every k-tree admits an additive $(k + 2)$-carcass constructible in linear time. In particular, any 2-tree admits an additive 4-carcass constructible in linear time.*

From the proof of Theorem 5 we can deduce also the following result.

Corollary 6. *Every 3-sun-free chordal graph admits an additive 2-carcass.*

Proof. Let $G = (V, E)$ be a 3-sun-free chordal graph and $s \in V$ be an arbitrary vertex of G. Let again $L_i := \{v \in V : d_G(v, s) = i\}$, $i = 0, 1, 2, \dots$ be the layering of G with respect to s. We call an edge of G *horizontal* if its end-vertices are equidistant from s. According to the proof of Theorem

5, if we can guarantee, by specially constructed shortest path tree T rooted at s, that a greedy routing path $R(x, y)$ will never use a horizontal edge of G (except, possibly, one which is incident to path sTy), then a surplus $\omega(G) - 1$ in the length of $R(x, y)$ can be avoided.

Let construct for a 3-sun-free chordal graph G a special shortest path tree T by specifying for each vertex $v \in L_i$ $(i = 1, 2, \ldots)$ that neighbor $u \in L_{i-1}$, to be its father in T, which maximizes $|N_G(u) \cap N_G(v)|$.

Let now $R(x, y)$ be a greedy routing path from x to y obtained by using the distances in this T, and assume that the first edge xx^* of this path is horizontal and is not incident to sTy. Let $x, x^* \in L_i$. Consider the fathers $f(x)$, $f(x^*)$ of x and x^* in T. Since $d_T(x, y) > d_T(x^*, y)$ and x^* is not an ancestor of y in T, vertices $f(x)$ and $f(x^*)$ are different. By Lemma 2, they must be adjacent in G. We know also, from $d_T(x^*, y) > d_T(f(x^*), y)$ and by choice of x^*, that $f(x^*) \notin N_G[x]$. To avoid an induced cycle on four vertices in G, $f(x)$ must be adjacent to x^* in G. Now, since $x \in N_G(x^*)$ is adjacent in G to $f(x)$ and not to $f(x^*)$, by construction of T, there must be a vertex $w \in N_G(x^*)$ which is adjacent in G to $f(x^*)$ and not to $f(x)$. By Lemma 2, necessarily, $w \in L_i$. As adjacent vertices $f(x)$ and $f(x^*)$ from L_{i-1} are equidistant from s, by Lemma 4, there must exist a vertex $c \in L_{i-2}$ which is adjacent to both $f(x)$ and $f(x^*)$ in G. It is easy to see now, from the distance requirements and the chordality of G, that the vertices $x, x^*, f(x), f(x^*), w, c$ induce a 3-sun in G, which is impossible.

Thus, we cannot have such a horizontal edge in $R(x, y)$. According to the proof of Theorem 5, $g(x, y) \leq d_G(x, y) + 3$, and if $g(x, y) = d_G(x, y) + 3$ then for vertices x, y, z, v, v' we have: $v' \in L_k$, $z \in L_{k+1}$, $d_G(x, v) = i - k$, $d_G(y, v') = j - k$, $d_G(x, z) = i - k - 1$ and $d_G(y, z) = j - k - 1$. We will show that these conditions lead to an induced 3-sun in G. Pick a neighbor a of v' on $v'Ty$ and a neighbor x_{q-1} of v on $R(x, y)$. We know that $va, x_{q-1}v' \notin E$. Since z and a are connected outside of $D_{j-k-1}(y)$ and z and x_{q-1} are connected outside of $D_{i-k-1}(x)$, z must be adjacent in G to a and x_{q-1} (note that z can coincide neither with a nor with x_{q-1}). By Lemma 4, vertices v, v' must have a common neighbor c in L_{k-1}. Now, from the distance requirements and the chordality of G, the vertices x_{q-1}, z, a, v', c, v induce a 3-sun in G, which is impossible. □

7 Conclusion and future work

In this paper, we investigated a new strategy 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 this strategy, each vertex v has full knowledge of its neighborhood $N_G[v]$ in G and uses a small piece of global information from spanning tree T (a small piece of the distance information from T), available locally at v, to navigate in G. We defined a new combinatorial structure *additive r-carcass* and 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 are easy to construct.

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

- What other interesting graph families do admit additive (or multiplicative) c-carcasses for small values of c?
- Given a graph G and a number c, how hard is it to decide whether G admits a c-carcass (additive or multiplicative)? If a c-carcass 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?

References

1. I. Abraham, C. Gavoille, Object location using path separators, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing (PODC 2006)*, Denver, Colorado, USA, July 23-26, 2006, ACM, pp 188-197.
2. I. Abraham, C. Gavoille, A.V. Goldberg, D. Malkhi, Routing in Networks with Low Doubling Dimension, *ICDCS 2006*, p. 75 (see also http://dept-info.labri.fr/ gavoille/article/AGGM06).
3. L. A. Adamic, R.M. Lukose, and B.A. Huberman, *Local Search in Unstructured Networks,* Willey, New-York, 2002.
4. L. A. Adamic, R. M. Lucose, A. R. Puniyani, and B. A. Huberman, Search in power-law networks, *Physical Review E* 64 (2001), 046135, 1-8.
5. Méziane Aïder, Almost distance-hereditary graphs, *Discrete Mathematics* 242, 2002, 1–16.
6. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, Routing with guaranteed delivery in ad hoc wireless networks, *Proceedings of the 3rd International Workshop on Discrete algorithms and Methods for Mobile Computing and Communications*, 1999, ACM Press, pp.48–55.
7. A. Brandstädt, V.D. Chepoi and F.F. Dragan, The algorithmic use of hypertree structure and maximum neighbourhood orderings, *Discrete Appl. Math.* 82 (1998), 43–77.
8. A. Brandstädt, V.D. Chepoi and F.F. Dragan, Distance approximating trees for chordal and dually chordal graphs, *Journal of Algorithms* 30 (1999), 166–184.
9. A. Brandstädt, F.F. Dragan, V.D. Chepoi and V.I. Voloshin, Dually chordal graphs, *SIAM J. Discrete Math.* 11 (1998), 437–455.
10. A. Brandstädt, V. Bang Le, Jeremy P. Spinrad, *Graph Classes: A Survey*, SIAM Monographs on Discrete Mathematics and Applications. Philadelphia, 1999.
11. L. Cai, D.G. Corneil, Tree spanners, *SIAM J. Disc. Math.,* 8 (1995), 359–387.
12. H. T.-H. Chan, A. Gupta, B. M. Maggs, and S. Zhou, On hierarchical routing in doubling metrics, *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, Vancouver, British Columbia, Canada, January 23-25, 2005. SIAM, pp. 762–771.
13. V.D. Chepoi, F.F. Dragan and C. Yan, Additive Sparse Spanners for Graphs with Bounded Length of Largest Induced Cycle, *Theoretical Computer Science* 347 (2005), 54–75.
14. S. Cicerone, G. Di Stefano, $(k,+)$-distance-hereditary graphs, *Journal of Discrete Algorithms* 1, 2003, 281-302.
15. D.G. Corneil, F.F. Dragan and E. Köhler, On the power of BFS to determine a graph's diameter, *Networks* 42 (2003), 209–222.
16. Y. Dourisboure, Compact Routing Schemes for Bounded Tree-Length Graphs and for k-Chordal Graphs, *DISC 2004*, 365–378.
17. F.F. Dragan and M. Matamala, Navigating in a graph by aid of its spanning tree, *19th International Symposium on Algorithms and Computation (ISAAC 2008)*, Surfers Paradise, Gold Coast, Australia, December 15–17, 2008, *Springer, Lecture Notes in Computer Science 5369*, pp. 789-800, 2008.
18. F.F. Dragan and C. Yan, Collective Tree Spanners in Graphs with Bounded Genus, Chordality, Tree-width, or Clique-width, *ISAAC 2005*, LNCS 3827, pp. 583–592.
19. F.F. Dragan, C. Yan and D.G. Corneil, Collective Tree Spanners and Routing in AT-free Related Graphs, *Journal of Graph Algorithms and Applications,* 10 (2006), 97–122.
20. F.F. Dragan, C. Yan and I. Lomonosov, Collective tree spanners of graphs, *SIAM J. Discrete Math.* 20 (2006), 241–260.
21. R. Fonseca, S. Ratnasamy, J. Zhao, C. T. Ee, D. Culler, S. Shenker, and I. Stoica, Beacon vector routing: Scalable point-to-point routing in wireless sensornets, *Proceedings of the Second USENIX/ACM Syposium on Networked Systems Design and Implementation (NSDI 2005),* 2005.
22. P. Fraigniaud, 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 4698*, Springer 2007, pp. 2–11.
23. P. Fraigniaud and C. Gavoille, Memory requirements for univesal routing schemes, In *Proceedings of the 14th Annual ACM Symposium on Principles of Distributed Computing*, Ontario, Canada, 1995, pp. 223–230.
24. P. Fraigniaud and C. Gavoille, Routing in trees, *28th International Colloquim on Atomata, Languages and Programming (ICALP)*, volume 2076 of Lecture Notes in Computer Science, 757–772, 2001.

25. C. Gavoille, A survey on interval routing schemes, *Theoretical Computer Science*, 245 (1999), 217–253.
26. C. Gavoille, Routing in distributed networks: Overview and open problems, *ACM SIGACT News - Distributed Computing Column,* 32 (2001).
27. C. Gavoille and M. Gengler, Space-efficiency of routing schemes of stretch factor three, *Journal of Parallel and Distributed Computing*, 61 (2001), 679–687.
28. C. Gavoille, N. Hanusse, Compact Routing Tables for Graphs of Bounded Genus, *ICALP 1999*, LNCS 1644, pp. 351–360.
29. C. Gavoille, D. Peleg, S. Pérennès, and R. Raz, Distance labeling in graphs, *J. Algorithms,* 53 (2004), 85–112.
30. C. Gavoille and S. Pérennès, Memory requirements for routing in distributed networks, In *Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing,* Philadelphia, Pennsylvania, 1996, pp. 125–133.
31. S. Giordano and I. Stojmenovic, Position based routing algorithms for ad hoc networks: A taxonomy, In X. Cheng, X. Huang, and D. Du, editors, *Ad Hoc Wireless Networking,* pages 103–136. Kluwer, 2004.
32. P. Jacquet and L. Viennot, Remote spanners: what to know beyond neighbors, *23rd IEEE International Parallel and Distributed Processing Symposium (IPDPS 2009))*, pp. 1–15, 2009.
33. B. Karp and H.T. Kung, GPSR: greedy perimeter stateless routing for wireless networks, *Proceedings of the 6th ACM/IEEE MobiCom.*, 2000, ACM Press, pp. 243–254.
34. J.M. Kleinberg, The small-world phenomenon: an algorithm perspective, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing (STOC 2000)*, May 21-23, 2000, Portland, OR, USA. ACM, pp. 163–170.
35. R. Kleinberg, Geographic routing using hyperbolic space, In *INFOCOM 2007*, pp. 1902–1909.
36. D. Kratsch, H.-O. Le, H. Müller, E. Prisner and D. Wagner, Additive tree spanners, *SIAM J. Discrete Math.* 17 (2003), 332–340.
37. F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, Geometric ad-hoc routing: of theory and practice, *Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing*, 2003, ACM Press, pp. 63–72.
38. J. van Leeuwen and R.B. Tan, Interval routing, *The Computer Journal,* 30 (1987), 298–307.
39. D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins, Geographic routing in social networks, *PNAS*, 102 (2005), 11623–11628.
40. D. Peleg, Distributed Computing: A Locality-Sensitive Approach, *SIAM Monographs on Discrete Math. Appl.*, SIAM, Philadelphia, 2000.
41. D. Peleg, Proximity-Preserving Labeling Schemes and Their Applications, *J. of Graph Theory*, 33 (2000), 167–176.
42. A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica, Geographical routing without location information, *Proceedings of MobiCom 2003*, 2003, pp. 96–108.
43. D. Rautenbach, Graphs with small additive stretch number, *Discussiones Mathematicae Graph Theory* 24, 2004, 291-302.
44. N. Santoro and R. Khatib, Labeling and implicit routing in networks, *The Computer Journal,* 28 (1985), 5–8.
45. A. Slivkins, Distance estimation and object location via rings of neighbors, *PODC 2005*, pp. 41–50.
46. K. Talwar, Bypassing the embedding: Algorithms for low dimensional metrics, *STOC 2004*, pp. 281–290.
47. M. Thorup, Compact oracles for reachability and approximate distances in planar digraphs, *Journal of the ACM,* 51 (2004), 993–1024.
48. M. Thorup and U. Zwick, Compact routing schemes, In 13^{th} *Ann. ACM Symp. on Par. Alg. and Arch.*, July 2001, pp. 1–10.
49. V.I. Voloshin, Properties of triangulated graphs, *Issledovaniye operaziy i programmirovanie,* Kishinev, 1982, pp. 24–32 (in Russian).