Routing in Histograms*

Man-Kwun Chiu[1], Jonas Cleve[1], Katharina Klost[1], Matias Korman[3], Wolfgang Mulzer[1], André van Renssen[4], Marcel Roeloffzen[5], and Max Willert[1]

1 **Institut für Informatik, Freie Universität Berlin, 14195 Berlin, Germany**
`{chiumk,jonascleve,kathklost,mulzer,willerma}@inf.fu-berlin.de`

3 **Department of Computer Science, Tufts University, Medford, MA, USA**
`matias.korman@tufts.edu`

4 **School of Computer Science, University of Sydney, Sydney, Australia**
`andre.vanrenssen@sydney.edu.au`

5 **TU Eindhoven, Eindhoven, The Netherlands**
`m.j.m.roeloffzen@tue.nl`

Abstract

Let P be a *histogram* with n vertices, i.e., an x-monotone orthogonal polygon whose upper boundary is a single edge. Two points $p, q \in P$ are *co-visible* if and only if the (axis-parallel) bounding rectangle of p and q is in P. In the r-visibility graph of P, we connect two vertices of P with an *unweighted* edge if and only if they are co-visible. We consider *routing with preprocessing* in P. We may preprocess P to obtain a *label* and a *routing table* for each vertex of P. Then, we must be able to route a packet between any two vertices s and t of P, where each step may use only the label of the target node t, the routing table, and the neighborhood of the current node.

We present a routing scheme for histograms that sends any data packet along a shortest path. Each label needs $O(\log n)$ bits, while the routing table of each node consists of a single bit.

1 Introduction

The *routing* problem is a classic question in distributed graph algorithms [15, 22]. We have a graph G and would like to preprocess it for the following task: route a data packet located at some *source* vertex s to a *target* vertex t, given by its *label* – a bit string that identifies the node in the network. The routing should have the following properties: (A) *locality*: to determine the next step of the packet, it should use only information available locally at the current vertex. The most important local information consists of a *routing table* for each vertex; (B) *efficiency*: the packet should travel along a path whose length is not much larger than the length of a shortest path between s and t. The ratio between the length of this routing path and a shortest path is called the *stretch factor*; and (C) *compactness*: the space requirements for labels and routing tables should be small. Storing the complete shortest path tree of v in every node v of G leads to a perfect efficiency but lacks compactness.

There are many compact routing schemes for general graphs [1, 2, 11–13, 23, 24]. For example, the scheme by Roditty and Tov [24] needs to store a poly-logarithmic number of bits in the packet header and it routes a packet from s to t on a path of length $O\big(k\Delta + m^{1/k}\big)$, where Δ is the shortest path distance between s and t, $k > 2$ is any fixed integer, n is the number of nodes, and m is the number of edges. The local routing tables use $mn^{O(1/\sqrt{\log n})}$ space. In the late 1980's, Peleg and Upfal [22] proved that in general graphs, any routing

* Partially supported by ERC STG 757609, DFG grant MU 3501/1-2, MEXT KAKENHI No. 17K12635, NSF award CCF-1422311, and JST ERATO Grant Number JPMJER1201, Japan. A full version is available on the arXiv under `arXiv:1902.06599`.

35th European Workshop on Computational Geometry, Utrecht, The Netherlands, March 18–20, 2019.
This is an extended abstract of a presentation given at EuroCG'19. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

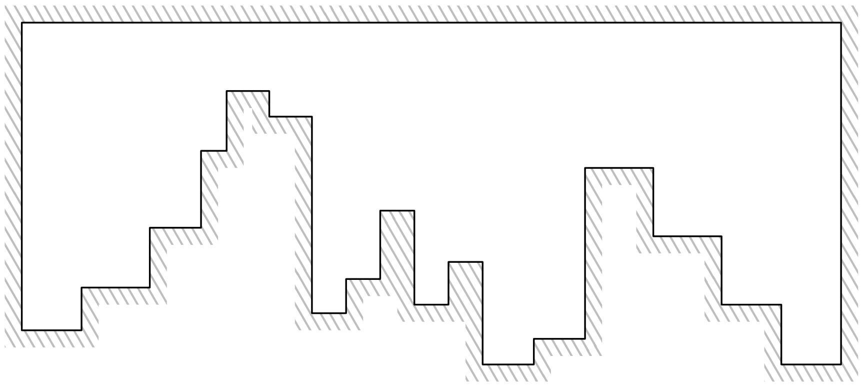

Figure 1 The boundary of a histogram has an x-monotone chain and a single horizontal edge.

scheme with constant stretch factor must store $\Omega(n^c)$ bits per vertex, for some constant $c > 0$. Thus, it is natural to focus on special graph classes to obtain better routing schemes. For instance, there are compact and efficient routing schemes for trees, planar graphs, unit disk graphs, and metric spaces with bounded doubling dimension [14, 18, 19, 25–27, 29].

Another approach is *geometric routing*: the graph resides in a geometric space, and the routing algorithm has to determine the next vertex for the packet purely based on the local geometric information (and possibly the packet header), see for instance [9, 10] and the references therein. There are no routing tables. In a recent result, Bose *et al.* [10] show that when vertices do not store any routing tables, no geometric routing scheme can achieve stretch factor $o(\sqrt{n})$. This lower bound applies irrespective of the header size.

We consider routing in visibility graphs of polygons. Banyassady *et al.* [3] presented an efficient and compact routing scheme for polygonal domains assuming Euclidean weights. They ask whether there is an efficient routing scheme for visibility graphs with unit weights, arguably a more applied setting. We address this open problem by combining the two approaches of geometric and compact routing: we use routing tables at the vertices to represent information about the structure of the graph, but we also assume that the labels of all adjacent vertices are stored in a link table and are therefore available for each node. This is reasonable from a practical point of view. The link table is not part of the routing table and the size of this list is not relevant for the compactness, since it depends purely on the graph and cannot be influenced during preprocessing. We focus on r-visibility (*rectilinear*-visibility) graphs of histograms: a histogram P is an orthogonal polygon bounded by an x-monotone polygonal chain and a single horizontal line segment, see Figure 1. Two vertices v, w in P are connected in the r-visibility graph $G(P)$ by an unweighted edge if and only if the axis-parallel rectangle spanned by v and w is contained in the (closed) region P. Even this seemingly simple case turns out to be quite challenging and reveals the whole richness of the compact routing problem in unweighted, geometrically defined graphs. Furthermore, histograms constitute a natural starting point, since they are crucial building blocks in many visibility problems; see, for instance, [4–8, 17]. In addition, r-visibility is a popular concept that enjoys many useful structural properties, see, e.g., [16, 17, 20, 21, 28].

We present a routing scheme for $G(P)$ with label size $2 \cdot \lceil \log n \rceil$, routing table size 1 and stretch 1, i.e., we route on a shortest path.

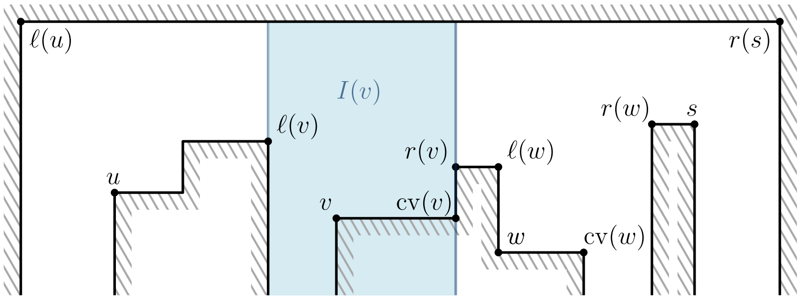

Figure 2 The interval $I(v)$ as well as left, right, and corresponding vertices.

2 Preliminaries

Routing schemes. Let $G = (V, E)$ be an *undirected, unweighted, simple, connected graph*. The (closed) *neighborhood* of a vertex $v \in V$, $N(v)$, is the set containing v and its adjacent nodes. The length of a path π in G is denoted by $|\pi|$. Moreover, for $v, w \in V$, we let $d(v, w)$ denote the length of a shortest path in G with endpoints v and w.

We define a *routing scheme*. Every node v is assigned a *label* $\text{lab}(v) \in \{0,1\}^*$ to identify v in the network and a *routing table* $\rho(v) \in \{0,1\}^*$ storing relevant properties of G. Labels and routing tables are chosen during *preprocessing*. Moreover, every node has a *link table*—a list of the labels of $N(v)$. The algorithm to find the next step of a packet is modeled by a *routing function* $f\colon \big(\{0,1\}^*\big)^3 \to V$. The function uses the link and routing table at a current node s as well as the label $\text{lab}(t)$ of the target node t to determine a next node v adjacent to s where the packet is forwarded to. The routing scheme is *correct* if the following holds: for any $s, t \in V$, let $p_0 = s$ and $p_{i+1} = f\Big(\text{lab}\left(N(p_i)\right), \rho(p_i), \text{lab}(t)\Big)$, for $i \geq 0$. Then, there is a $k = k(s,t) \geq 0$ with $p_k = t$ and $p_i \neq t$, for $i < k$. The routing scheme *reaches* t in k steps. We call $\pi : \langle p_0, \dots, p_k \rangle$ the *routing path* from s to t. The *routing distance* is $k(s,t)$.

The various pieces of information used for the routing should be small. This is measured by the *label* and *routing table size*. The routing path should be as small as possible. This is measured by the *stretch*—the ratio of the lengths of the routing and the shortest path.

Polygons. Let P be an *x-monotone orthogonal polygon* in general position with n vertices $V(P)$. No three vertices in $V(P)$ lie on a horizontal line. We call P a *histogram* if the upper boundary is a single horizontal *base edge*. Its endpoints are the *base vertices*. The vertices of P are indexed counterclockwise from 0 to $n-1$ starting at the left base vertex. For $v \in V(P)$, we write v_x for the x-coordinate, v_y for the y-coordinate, and v_{id} for the index.

We consider the *r-visibility graph* $G(P) = \big(V(P), E(P)\big)$ of P: there is an edge between two vertices $v, w \in V(P)$ if they are *co-visible*, i.e., the axis-aligned rectangle spanned by v and w is in (the closed set) P. We call $d(v, w)$ the *hop distance* between two vertices in v, w.

Next, we classify the vertices of P. A vertex v in P is incident to exactly one horizontal edge h. We call v a *left* vertex if it is the left endpoint of h; otherwise, v is a *right* vertex. Furthermore, v is *convex* if the interior angle at v is $\pi/2$; otherwise, v is *reflex*. Accordingly, every vertex of P is either *ℓ-convex*, *r-convex*, *ℓ-reflex*, or *r-reflex*.

Visibility Landmarks. Let P be a histogram. We associate with each $v \in V(P)$ three landmark vertices in P; see Figure 2. The *corresponding vertex* of v, $\text{cv}(v)$, shares the same horizontal edge with v. The *left bounding vertex* of v, $\ell(v)$, is the leftmost visible vertex

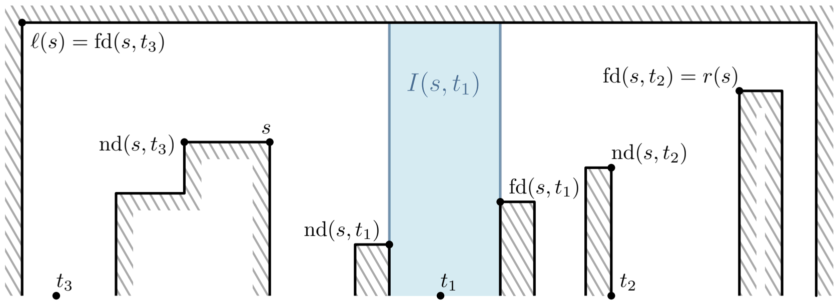

Figure 3 The near and the far dominators. Observe that $\text{fd}(s, t_3)$ is not a vertex.

from v closest to the base edge, i.e., $\ell(v) = \text{argmin}\{w_{\text{id}} \mid w \in N(v)\}$. The *right bounding vertex* of v, $r(v)$, is defined analogously, i.e., $r(v) = \text{argmax}\{w_{\text{id}} \mid w \in N(v)\}$.

Let $v, w \in V(P)$. The *interval* $[v, w]$ is the set of vertices in P whose x-coordinates lie between those of v and w, i.e., $[v, w] = \{u \in V(P) \mid v_x \leq u_x \leq w_x\}$. By general position, this corresponds to index intervals. More precisely, if v is either an r-reflex vertex or the left base vertex and w is either ℓ-reflex or the right base vertex, then $[v, w] = \{u \in V(P) \mid v_{\text{id}} \leq u_{\text{id}} \leq w_{\text{id}}\}$. The set $I(v) = [\ell(v), r(v)]$ is called the *interval of* v. We have $N(v) \subseteq I(v)$.

Let s and t be two vertices with $t \in I(s) \setminus N(s)$. We define two more landmarks for s and t. Assume $s_x < t_x$, the other case is symmetric. The *near dominator* $\text{nd}(s, t)$ of t with respect to s is the rightmost vertex in $N(s)$ that is not to the right of t. If there is more than one such vertex, $\text{nd}(s, t)$ is the vertex closest to the base line. The *far dominator* $\text{fd}(s, t)$ of t with respect to s is the leftmost vertex in $N(s)$ that is no to the left of t. If there is more than one such vertex, $\text{fd}(s, t)$ is the vertex closest to the base line. The interval $I(s, t) = [\text{nd}(s, t), \text{fd}(s, t)]$ has all vertices between the near and far dominator; see Figure 3.

3 Visibility and Paths

Let P be a histogram. We present some observations on the visibility in P. Then, we analyze the structure of (shortest) paths in a histograms. We omit the proofs for space reasons.

▶ **Observation 3.1.** *Let $v \in V(P)$ be r-reflex or the left base vertex, and let $u \in [v, r(v)]$ be a vertex distinct from v and $r(v)$. Then, $I(u) \subseteq [v, r(v)]$.*

▶ **Observation 3.2.** *Let $v \in V(P)$ be a left (right) vertex distinct from the base vertex. Then, v can see exactly two vertices to its right (left): $\text{cv}(v)$ and $r(v)$ ($\ell(v)$).*

The following lemma identifies some vertices that must appear on any path; see Figure 4.

▶ **Lemma 3.3.** *Let $v, w \in V(P)$ be co-visible vertices such that v is either r-reflex or the left base vertex and w is either ℓ-reflex or the right base vertex. Let s and t be two vertices with $s \in [v, w]$ and $t \notin [v, w]$. Then, any path between s and t includes v or w.*

The next lemma shows that if $t \notin I(s)$, there is a shortest path from s to t that uses the higher vertex of $\ell(s)$ and $r(s)$; see Figure 4.

▶ **Lemma 3.4.** *Let s and t be two vertices with $t \notin I(s)$. If $\ell(s)_y > r(s)_y$ ($\ell(s)_y < r(s)_y$), then there is a shortest path from s to t using $\ell(s)$ ($r(s)$).*

The next lemma considers the case where t is in $I(s)$. Then, the near and far dominator are the potential vertices that lie on a shortest path from s to t.

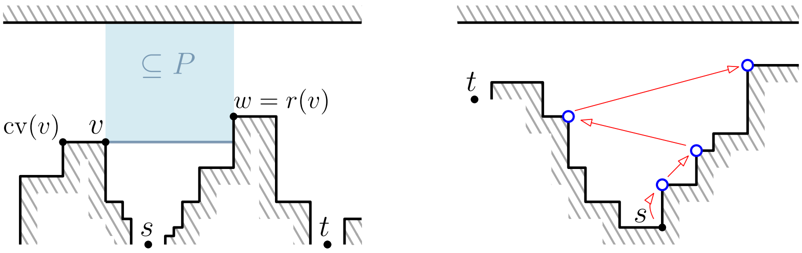

Figure 4 Left: Any s-t-path includes v or w. Right: A shortest s-t path using the higher vertex.

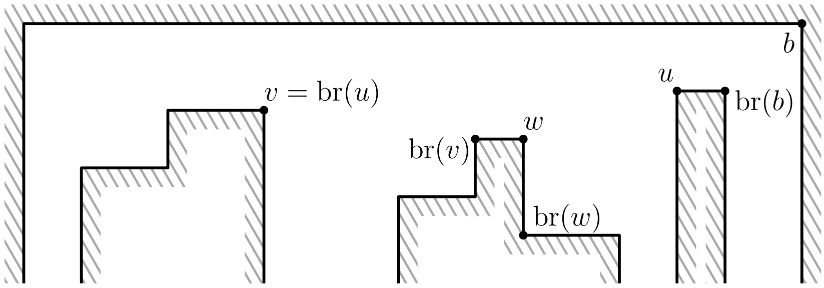

Figure 5 The breakpoints of some vertices.

▶ **Lemma 3.5.** *Let s and t be two vertices with $t \in I(s) \setminus N(s)$. Then,* $\mathrm{nd}(s,t)$ *is reflex and either* $\mathrm{fd}(s,t) = \ell(\mathrm{nd}(s,t))$ *or* $\mathrm{fd}(s,t) = r(\mathrm{nd}(s,t))$.

4 The Routing Scheme

Let P be a histogram, $|V(P)| = n$. Our approach is as follows: as long as a target vertex t is not contained in the interval $I(s)$ of a current vertex s, i.e., as long as there is a higher vertex blocking visibility between s and t, we have to leave the current pocket as quickly as possible. Once we have reached a high enough spike, we find the pocket containing t.

Labels and routing tables. Let v be a vertex. If v is convex and not a base vertex we let $\mathrm{lab}(v) = v_{\mathrm{id}}$. Otherwise, suppose that v is an r-reflex vertex or the left base vertex. The *breakpoint of* v, $\mathrm{br}(v)$, is the left endpoint of the horizontal edge with the highest y-coordinate to the right of and below v visible from v; analogous definitions apply to ℓ-reflex vertices and the right base vertex; see Figure 5. We set $\mathrm{lab}(v) = (v_{\mathrm{id}}, \mathrm{br}(v)_{\mathrm{id}})$. The routing table $\rho(v)$ stores one bit, indicating whether $\ell(v)_y > r(v)_y$, or not.

The routing function. We are given the current vertex s and the label $\mathrm{lab}(t)$ of the target vertex t. If t is visible from s, i.e., if $\mathrm{lab}(t) \in \mathrm{lab}(N(s))$, we directly go from s to t. Thus, assume $t \notin N(s)$. First, we check $t \in I(s)$ as follows: we determine the smallest and largest id in the link table of s, i.e., we determine $\ell(s)_{\mathrm{id}}$ and $r(s)_{\mathrm{id}}$ and check $t_{\mathrm{id}} \in [\ell(s)_{\mathrm{id}}, r(s)_{\mathrm{id}}]$, which is the case if and only if $t \in I(s)$. There are two cases, illustrated in Figure 6.

First, assume $t \notin I(s)$. If $\rho(s)$ indicates $\ell(s)_y > r(s)_y$, we take the hop to $\ell(s)$; otherwise, we take the hop to $r(s)$. By Lemma 3.4, this hop is on a shortest path from s to t.

Second, suppose that $t \in I(s) \setminus N(s)$. This case is a bit more involved. We use the link table of s and the label of t to determine $\mathrm{fd}(s,t)$ and $\mathrm{nd}(s,t)$. Again, we can do this by

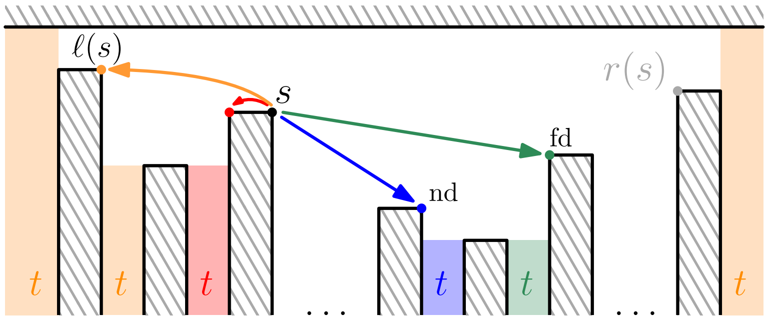

Figure 6 The cases where the vertex t lies and the vertices where the data packet is sent to.

comparing the ids. Lemma 3.5 states that either $\mathrm{fd}(s,t) = \ell(\mathrm{nd}(s,t))$ or $\mathrm{fd}(s,t) = r(\mathrm{nd}(s,t))$. We discuss the case that $\mathrm{fd}(s,t) = r(\mathrm{nd}(s,t))$, the other case is symmetric. By Lemma 3.3, any shortest path between s and t includes $\mathrm{fd}(s,t)$ or $\mathrm{nd}(s,t)$. Moreover, due to Lemma 3.5, $\mathrm{nd}(s,t)$ is reflex, and we can use its label to access $b_{\mathrm{id}} = \mathrm{br}(\mathrm{nd}(s,t))_{\mathrm{id}}$. The vertex b splits $I(s,t) = [\mathrm{nd}(s,t), \mathrm{fd}(s,t)]$ into two disjoint subintervals $[\mathrm{nd}(s,t), b]$ and $[\mathrm{cv}(b), \mathrm{fd}(s,t)]$. Also, b and $\mathrm{cv}(b)$ are not visible from s, as they are located strictly between the far and the near dominator. Based on b_{id}, we can now decide on the next hop.

If $t \in [\mathrm{nd}(s,t), b]$, we take the hop to $\mathrm{nd}(s,t)$. If $t = b$, our packet uses a shortest path. Assume that t lies between $\mathrm{nd}(s,t)$ and b. Then, b is ℓ-reflex and we can apply Lemma 3.3 to see that any shortest path from s to t includes $\mathrm{nd}(s,t)$ or b. But since $d(s,b) = 2$, our data packet routes along a shortest path. If $t \in [\mathrm{cv}(b), \mathrm{fd}(s,t)]$, we take the hop to $\mathrm{fd}(s,t)$. The argument is similar. The following theorem summarizes our discussion.

▶ **Theorem 4.1.** *Let P be a histogram with n vertices. There is a routing scheme for $G(P)$ with label size $2 \cdot \lceil \log n \rceil$, routing table size 1, and stretch 1.*

5 Conclusion

We gave the first routing scheme for the hop-distance in simple polygons. In particular, we have a routing scheme for histograms with label size $2 \cdot \lceil \log n \rceil$, routing table size 1, and stretch 1. As we show in the full version, our method also extends to more general *double histograms*. The following open problems arise naturally. First, it would be interesting to see how the routing scheme extends to monotone polygons as well as arbitrary orthogonal polygons, assuming r-visibility. After that, it will be interesting to take a closer look at (orthogonal) polygons assuming the more common notion of l-visibility (*line*-visibility). Here, the structure of visibility—even in simple histograms—is much more complicated and we can no longer assume integer coordinates.

References

1 Ittai Abraham and Cyril Gavoille. On approximate distance labels and routing schemes with affine stretch. In *Proc. 25th Int. Symp. Dist. Comp. (DISC)*, pages 404–415, 2011.

2 Baruch Awerbuch, Amotz Bar-Noy, Nathan Linial, and David Peleg. Improved routing strategies with succinct tables. *J. Algorithms*, 11(3):307–341, 1990.

3 Bahareh Banyassady, Man-Kwun Chiu, Matias Korman, Wolfgang Mulzer, André van Renssen, Marcel Roeloffzen, Paul Seiferth, Yannik Stein, Birgit Vogtenhuber, and Max

Willert. Routing in polygonal domains. In *Proc. 28th Annu. Internat. Sympos. Algorithms Comput. (ISAAC)*, pages 10:1–10:13, 2017.

4 Andreas Bärtschi. Coloring variations of the art gallery problem. *Master's thesis, Department of Mathematics, ETH Zürich*, 2011.

5 Andreas Bärtschi, Subir Kumar Ghosh, Matúš Mihalák, Thomas Tschager, and Peter Widmayer. Improved bounds for the conflict-free chromatic art gallery problem. In *Proc. 30th Annu. Sympos. Comput. Geom. (SoCG)*, page 144, 2014.

6 Andreas Bärtschi and Subhash Suri. Conflict-free chromatic art gallery coverage. *Algorithmica*, 68(1):265–283, 2014.

7 Pritam Bhattacharya, Subir Kumar Ghosh, and Sudebkumar Pal. Constant approximation algorithms for guarding simple polygons using vertex guards. `arXiv:1712.05492`, 2017.

8 Pritam Bhattacharya, Subir Kumar Ghosh, and Bodhayan Roy. Approximability of guarding weak visibility polygons. *Discrete Applied Mathematics*, 228:109–129, 2017.

9 Prosenjit Bose, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. Optimal local routing on Delaunay triangulations defined by empty equilateral triangles. *SIAM J. Comput.*, 44(6):1626 – 1649, 2015.

10 Prosenjit Bose, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. Competitive local routing with constraints. *J. of Computational Geometry*, 8(1):125–152, 2017.

11 Shiri Chechik. Compact routing schemes with improved stretch. In *Proc. ACM Symp. Princ. Dist. Comp. (PODC)*, pages 33–41, 2013.

12 Lenore J Cowen. Compact routing with minimum stretch. *J. Algorithms*, 38(1):170–183, 2001.

13 Tamar Eilam, Cyril Gavoille, and David Peleg. Compact routing schemes with low stretch factor. *J. Algorithms*, 46(2):97–114, 2003.

14 Pierre Fraigniaud and Cyril Gavoille. Routing in trees. In *Proc. 28th Internat. Colloq. Automata Lang. Program. (ICALP)*, pages 757–772, 2001.

15 Silvia Giordano and Ivan Stojmenovic. Position based routing algorithms for ad hoc networks: A taxonomy. In *Ad hoc wireless networking*, pages 103–136. Springer-Verlag, 2004.

16 Frank Hoffmann. On the rectilinear art gallery problem. In *Proc. 17th Internat. Colloq. Automata Lang. Program. (ICALP)*, pages 717–728, 1990.

17 Frank Hoffmann, Klaus Kriegel, Subhash Suri, Kevin Verbeek, and Max Willert. Tight bounds for conflict-free chromatic guarding of orthogonal art galleries. *Comput. Geom. Theory Appl.*, 2018.

18 Haim Kaplan, Wolfgang Mulzer, Liam Roditty, and Paul Seiferth. Routing in unit disk graphs. *Algorithmica*, 80(3):830–848, 2018.

19 Goran Konjevod, Andréa W. Richa, and Donglin Xia. Scale-free compact routing schemes in networks of low doubling dimension. *ACM Trans. Algorithms*, 12(3):27:1–27:29, 2016.

20 Rajeev Motwani, Arvind Raghunathan, and Huzur Saran. Covering orthogonal polygons with star polygons: The perfect graph approach. *J. Comput. System Sci.*, 40(1):19–48, 1990.

21 Joseph O'Rourke. *Art gallery theorems and algorithms.* Oxford University Press, 1987.

22 David Peleg and Eli Upfal. A trade-off between space and efficiency for routing tables. *J. ACM*, 36(3):510–530, 1989.

23 Liam Roditty and Roei Tov. New routing techniques and their applications. In *Proc. ACM Symp. Princ. Dist. Comp. (PODC)*, pages 23–32, 2015.

24 Liam Roditty and Roei Tov. Close to linear space routing schemes. *Distributed Computing*, 29(1):65–74, 2016.

25 Nicola Santoro and Ramez Khatib. Labelling and implicit routing in networks. *The Computer Journal*, 28(1):5–8, 1985.

26 Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *J. ACM*, 51(6):993–1024, 2004.

27 Mikkel Thorup and Uri Zwick. Compact routing schemes. In *Proc. 13th ACM Symp. Par. Algo. Arch. (SPAA)*, pages 1–10, 2001.

28 Chris Worman and J Mark Keil. Polygon decomposition and the orthogonal art gallery problem. *Internat. J. Comput. Geom. Appl.*, 17(02):105–138, 2007.

29 Chenyu Yan, Yang Xiang, and Feodor F Dragan. Compact and low delay routing labeling scheme for unit disk graphs. *Comput. Geom. Theory Appl.*, 45(7):305–325, 2012.