Local Routing in Convex Subdivisions*

Prosenjit Bose[1], Stephane Durocher[2], Debajyoti Mondal[2], Maxime Peabody[1], Matthew Skala[1], and Mohammad Abdul Wahid[3]

[1] Carleton University, Ottawa, Canada, jit@scs.carleton.ca, maximepeabody@cmail.carleton.ca
[2] University of Manitoba, Winnipeg, Canada, {durocher,jyoti,mskala}@cs.umanitoba.ca
[3] Bits in Glass, Calgary, Canada, wahidrahman@gmail.com

Abstract. In various wireless networking settings, node locations determine a network's topology, allowing the network to be modelled by a geometric graph drawn in the plane. Without any additional information, local geometric routing algorithms can guarantee delivery to the target node only in restricted classes of geometric graphs, such as triangulations. In order to guarantee delivery on more general classes of geometric graphs (e.g., convex subdivisions or planar subdivisions), previous local geometric routing algorithms required $\Theta(\log n)$ state bits to be stored and passed with the message. We present the first local geometric routing algorithm using only one state bit to guarantee delivery on convex subdivisions and the first local geometric memoryless routing algorithm that guarantees delivery on edge-augmented monotone subdivisions (including all convex subdivisions) when the algorithm has knowledge of the incoming port (the preceding node on the route).

1 Introduction

1.1 Local Geometric Routing

A *local routing algorithm* determines a sequence of forwarding decisions that defines a path in a network from a source node to a given target node, where each internal node along the path selects one of its neighbours to extend the path as a function of its local network neighbourhood and limited information about the target node. Additional information available to each node on the path may include the identity of its neighbour that forwarded the message (the incoming port on which the message arrived) as well as a small number of state bits passed with the message (which may be modified locally before forwarding). In various wireless networking settings, the locations of nodes and physical proximity between nodes determine the pairs of nodes that can communicate; that is, the network is determined geometrically. The network's geometric properties can provide navigational cues, enabling a local routing algorithm to use

* This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). A preliminary version of the algorithm presented in Section 2 appeared in Mohammad Abdul Wahid's M.Sc. thesis [19].

this additional geometric information to guide a message towards its destination. Each node may know its location, allowing every node on the path to make a forwarding decision as a function of the relative locations of its neighbours, the target node, and itself. We refer to such algorithms as *local geometric routing algorithms.* This paper examines the problem of defining local geometric routing algorithms that guarantee delivery from any source node to any target node on specific classes of geometric graphs.

1.2 Model and Definitions

We represent a network by an undirected graph G drawn in the plane, where each vertex is represented by a point and each edge is represented by a (straight) line segment connecting the vertices at its endpoints. Let $V(G)$ denote the set of vertices (points) of G and let $E(G)$ denote its set of edges (line segments), where $n = |V(G)|$ and $m = |E(G)|$. To simplify the discussion, we assume that vertices are in general position. By that we mean that no three points are collinear and no two points have the same x-coordinate or the same y-coordinate.

We require G to be connected for a route to exist between any pair of nodes. The drawing need not be planar, although some of our discussion relates to planar subdivisions. A drawing of a graph G in the plane is a *planar subdivision* (also *planar drawing*, *plane graph*, or *planar straight-line graph*) if each edge in $E(G)$ is drawn as a line segment and any two edges intersect only at their common endpoint. A planar subdivision partitions the plane into *faces.* When each internal face is a convex polygon and the boundary of the outer face is the convex hull, the drawing is a *convex subdivision.* When each internal face is a triangle, the subdivision is a *triangulation.* When each internal face is an x-monotone polygon (but not necessarily convex) and the boundary of the outer face is also an x-monotone polygon, the drawing is a *monotone subdivision.* Recall that a polygon is x-monotone if the intersection of its interior with any vertical line gives a connected region (i.e., a line segment). Every convex subdivision is also a monotone subdivision.

When G contains a spanning subgraph that is a convex subdivision, (respectively, a monotone subdivision), then we say G is an *edge-augmented convex subdivision* (respectively, an *edge-augmented monotone subdivision*); in this case, G corresponds to a convex subdivision to which zero or more edges have been added joining pairs of vertices in the underlying convex subdivision, possibly creating edge crossings. Edge-augmented convex subdivisions are not planar in general. Any routing algorithm that guarantees delivery on edge-augmented convex subdivisions also guarantees delivery on convex subdivisions.

Using notation similar to that previously defined [3, 10, 11], a local geometric routing algorithm can be expressed as a *routing function* $f : V(G) \times V(G) \times \mathscr{P}(V(G)) \to V(G)$, where $\mathscr{P}()$ denotes the power set, with arguments $f(u, t, N(u))$ such that $u \in V(G)$ is the vertex for which a forwarding decision is being made (i.e., the node presently holding the message), $t \in V(G)$ is the target vertex, and $N(u) \subseteq V(G)$ is the set of neighbours of u in G. Upon receiving a message destined for a node t, a node u forwards the message to its neighbour $w = f(u, t, N(u))$.

If u knows which of its neighbours forwarded the message, then we say the routing algorithm is *predecessor-aware* and represent the corresponding routing function as $f(u, v, t, N(u))$, where $v \in V(G)$ denotes the neighbour of u that last forwarded the message. Otherwise, we say the routing algorithm is *predecessor-oblivious*. Furthermore, if c state bits are passed with the message then we say the routing algorithm is *c-bit* local and the routing function becomes $f(u, t, N(u), e)$ (or $f(u, v, t, N(u), e)$ if predecessor-aware), where $e \in \{0, 1\}^c$. We focus on the case $c = 1$. If no bits are passed with the message then we say the routing algorithm is *stateless*. Note that no state information is stored at a node after it has forwarded a message; that is, the network is *memoryless*. When a message is forwarded, its destination t and the c state bits are passed with it. All other information is available locally at node u. Randomized solutions exists (e.g., [10]); in this work we restrict attention to deterministic routing algorithms.

1.3 Related Work

When applying a local geometric algorithm that is stateless and predecessor-oblivious, every time a node u receives a message destined for a given target node t, u always forwards the message to its same neighbour. Consequently, stateless predecessor-oblivious routing algorithms that guarantee delivery are limited to restricted classes of geometric graphs. These include greedy routing [12] and compass routing [15], both of which succeed on any Delaunay triangulation but fail on more general triangulations [6], as well as greedy-compass routing [2], which succeeds on any triangulation. In a triangulation each node knows the complete set of edges bounding every face on which it is adjacent. Beyond triangulations are convex subdivisions, where faces remain convex, but a node only knows two edges bounding every face on which it is adjacent. Every stateless and predecessor-oblivious local geometric routing algorithm fails on some convex subdivision [2]. Conseqently, local routing algorithms require additional reference beacons, or the ability to store learned route information in state bits, to support successful navigation on convex subdivisions or, more generally, on planar subdivisions.

Face routing [15] succeeds on any planar subdivision, but requires both predecessor-awareness and $\Theta(\log n)$ state bits (assuming vertex coordinates can be stored using $\Theta(\log n)$ bits per vertex). Variants of face routing succeed on unit disc graphs [7] and some quasi unit disc graphs [16]. Some local geometric algorithms define a route (or a graph traversal) on planar and near-planar subdivisions by performing a depth-first traversal of a locally defined spanning tree [1, 4, 9, 17]; all such algorithms known require $\Theta(\log n)$ state bits. For graphs drawn in three-dimensional space, every stateless and predecessor-aware local geometric routing algorithm fails on some unit ball graph [11].

If $\Theta(\log n)$ state bits are available, then geometric information is not necessary to support local routing: by storing an index into a polynomial-length universal traversal sequence, predecessor-oblivious routing is possible on any graph, not restricted to belonging to any particular class of drawings [8]; this requires each node to be able to reconstruct the traversal sequence. Without geometric information, a stateless routing algorithm requires knowledge of a large neighbourhood around each node to guarantee delivery. Specifically, a predecessor-aware stateless routing algorithm requires each node to have knowledge of the induced subgraph within graph distance $n/3$ of itself; for predecessor-oblivious algorithms the distance increases to $n/2$ [3]. There exists a small set of graphs such that every stateless routing algorithm whose knowledge is limited to a smaller neighbourhood around each node fails on one or more of these graphs.

In addition to knowing the target node t, knowledge of the source node s also determines whether local routing is possible. Applying the right-hand rule along the edges of the sequence of faces that intersect the line segment from s to t gives a stateless predecessor-aware local geometric routing that succeeds on convex subdivisions (requiring knowledge of s) [17]; this is essentially face routing applied to a specific class of graphs that does not require backtracking. To succeed on planar subdivisions that are not convex, face routing is occasionally forced to backtrack [7, 15], requiring $\Theta(\log n)$ state bits. Knowledge of s is significant even when geometric information is not available. For example, given s, a stateless predecessor-aware local routing algorithm only requires knowledge of the induced subgraph within graph distance $n/4$ of each node to guarantee delivery in any graph, instead of distance $n/3$ without knowledge of s [3].

Although similar to local routing, *online routing* [5] differs by the fact that each node u along the route has complete information about the subgraph explored prior to arriving at u. Storing such information in a message requires $\Theta(n \log n)$ state bits in general.

Guaranteeing delivery on geometric graphs beyond triangulations requires state information or predecessor-awareness. In this paper we seek to bridge the gap between stateless predecessor-oblivious local routing algorithms, which cannot guarantee delivery even on convex subdivisions, and $\Theta(\log n)$-bit local routing algorithms. Specifically, we examine whether navigation is possible when a local routing algorithm is provided a single state bit and whether it is possible when enabled with predecessor awareness. In each case we seek to define a routing algorithm and to identify broad classes of geometric graphs on which the algorithm guarantees delivery. For surveys on local geometric routing, see Morin [17], Guan [14], Urrutia [18], and Frey et al. [13].

1.4 Overview of Results

No stateless predecessor-oblivious local geometric routing algorithm can guarantee delivery on convex subdivisions [2]; to succeed on convex subdivisions and, therefore, on more general classes of graphs, such as planar subdivisions, a local geometric routing algorithm must have the ability to store state information or be provided with predecessor awareness. To the authors' knowledge, prior to this work no predecessor-oblivious c-bit local geometric routing algorithm was known to guarantee delivery on convex subdivisions for any $c \in o(\log n)$. Similarly, no predecessor-aware stateless local geometric routing algorithm was

bits	predecessor oblivious	predecessor aware
0	triangulations [2] impossible on convex subdiv. [2]	**(edge-aug.) convex subdiv.,** **(edge-aug.) monotone subdiv.**
1	**convex subdiv.**	beyond edge-aug. monotone subdiv.: unknown
$O(1)$	beyond convex subdiv.: unknown	
$O(\log n)$	all graphs [8]	planar [15], unit disc [7], all graphs [8]

Table 1. Graphs on which local routing is possible. New contributions appear in bold.

known to guarantee delivery on convex subdivisions. This paper presents the first predecessor-aware stateless local geometric routing algorithm and the first 1-bit predecessor-oblivious local geometric routing algorithm to guarantee delivery on any non-trivial class of geometric graphs beyond triangulations. See Table 1. In Section 2 we present a predecessor-oblivious local geometric routing algorithm that uses one state bit ($c = 1$) and guarantees delivery on any convex subdivision. In Section 3, we present a stateless predecessor-aware local geometric routing algorithm that guarantees delivery on any edge-augmented monotone subdivision. We conclude with a discussion in Section 4.

2 Using One State Bit

We describe a predecessor-oblivious one-bit geometric local routing algorithm, called OneBit, that guarantees delivery on any convex subdivision. Let u denote the node holding the message, i.e., the node making a forwarding decision. As in compass routing [15] and greedy-compass routing [2], we refer to the *clockwise* (respectively, *counterclockwise*) *neighbour* of u relative to t, denoted $cw(u)$ (respectively, $ccw(u)$) defined as the node $v \in N(u)$ that forms the smallest clockwise (counterclockwise) angle $\angle tuv$. Let H_s denote the closed half-plane containing u whose boundary is the vertical line ℓ_t through t. See Figure 1A. Algorithm OneBit never forwards the message to a node outside H_s, enabling all nodes along the route to identify H_s consistently relative to ℓ_t, regardless of whether the source node s is left or right of ℓ_t.

Algorithm OneBit uses one state bit, denoted c, to determine whether node u should forward the message from u to $cw(u)$ or to $ccw(u)$. The state bit c can be initialized arbitrarily at the source node s, e.g., $c \leftarrow 0$. Node s does not need to know it is the source; the algorithm can initialize c arbitrarily if it is not assigned a value in $\{0, 1\}$. See Algorithm 1 and Figure 1B. The resulting route corresponds to a sequence of clockwise forwarding decisions (when $c = 0$), which we call an *clockwise chain*, followed by a sequence of counterclockwise forwarding decisions (when $c = 1$), which we call a *counterclockwise chain*, followed by another clockwise chain (when $c = 0$ again), and so on, until the message reaches the target node t. Note that a clockwise chain proceeds in a counterclockwise direction relative to t, and vice versa. The algorithm toggles the state bit whenever continuing the chain would send the message outside H_s. As we show, each chain in the resulting sequence is contained within a region bounded

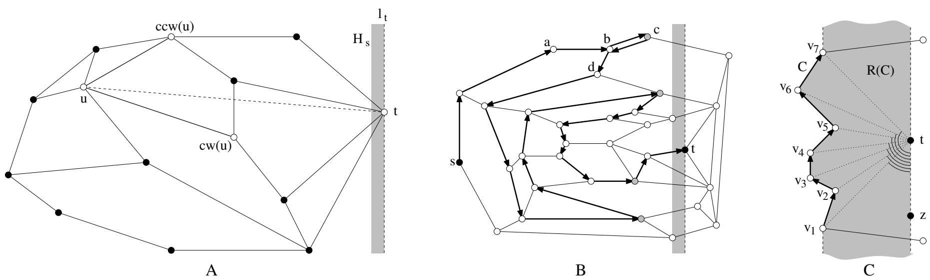

Fig. 1. **(A)** The clockwise and counterclockwise neighbours of u are defined relative to the line segment from u to t. The state bit c remains unchanged since both $ccw(u)$ and $cw(u)$ are in H_s; node u forwards the message to $ccw(u)$ if $c = 0$ and to $cw(u)$ if $c = 1$. **(B)** The bold arrows denote the sequence of local forwarding decisions made by Algorithm OneBit from s to t on this convex subdivision. Nodes at which the state bit toggles are shaded grey. Some edges can be traversed once in each direction on a route; e.g., see the subsequence $a \to b \to c \to b \to d$. **(C)** The increasing sequence of angles and the region $R(C)$ determined by a clockwise chain C

by the preceding chain, giving a convergence towards t. We refer to the first and last vertices on a chain according to the chronological order of the sequence of forwarding decisions as its *head* and *tail*, respectively, where the tail of the ith chain is the head of the $(i+1)$st chain.

Algorithm 1 OneBit(u, c, t)

Preconditions: u is the node holding the message, $N(u)$ is its set of neighbours, $c \in \{0, 1\}$ is the state bit passed with the message, t is the target node, and $cw(u)$ and $ccw(u)$ denote the clockwise and counterclockwise neighbours of u, respectively.

Postconditions: Forward the message from u to w with state bit c', where $w \in N(u)$.

```
c' ← c
if [c' = 0 and ccw(u) ∉ H_s] or [c' = 1 and cw(u) ∉ H_s] then
    c' ← not c' (The current chain cannot be continued in H_s: change states.)
end if
if t ∈ N(u) then (The target node t is adjacent to u.)
    w ← t (Forward the message to the target node t.)
else if c' = 0 then (State 0)
    w ← ccw(u) (Forward the message along a counterclockwise chain.)
else (State 1)
    w ← cw(u) (Forward the message along a clockwise chain.)
end if
```

Lemma 1. *For every node u in a convex subdivision, if $u \neq t$, then $cw(u) \in H_s$ or $ccw(u) \in H_s$.*

Proof. The lemma follows from the fact that the half-plane H_s is closed and is bounded by the vertical line ℓ_t through t and that every face is convex. □

Lemma 2. *Every clockwise (respectively, counterclockwise) chain C terminates, either at t or at the head of an oppositely oriented chain.*

Proof. Without loss of generality, suppose C is a clockwise chain corresponding to the sequence of vertices $v_1, \ldots, v_k$. By construction, the nodes $v_1, \ldots, v_k$ are all contained in H_s and corresponds to a sequence of increasing angles $\angle v_1 t z < \cdots < \angle v_k t z$, where z is any point that lies below t on ℓ_t. See Figure 1C. As defined in Algorithm 1, the chain C terminates when the tail node u has no clockwise neighbour. By Lemma 1, u must have a counterclockwise neighbour v, which defines the head of the subsequent counterclockwise chain. □

Lemma 2 implies that the forwarding sequence cannot continue indefinitely (i.e., it cannot cycle) without a change of state. Consequently, every chain C has a head and a tail. Given a chain C, let $R(C)$ denote the region bounded by C, the respective vertical rays emanating away from its head and tail, and ℓ_t. See Figure 1C. We say a chain is *complete* if it originated and terminated as a result of toggling the state bit. Consequently, all chains are complete, except the first (whose head is the source node s) and the last (whose tail is the target node t).

Lemma 3. *If C_i and C_j are any two chains in a route such that C_i is complete and C_i precedes C_j, then $R(C_j) \subseteq R(C_i)$*

Proof. The result follows by induction on the sequence of chains between C_i and C_j. Consider the case when C_i and C_j are consecutive chains. Without loss of generality, suppose C_i is a clockwise chain. Let u denote any node in C_i other than the tail. Node u forwards the message to its neighbour $v = cw(u)$. Therefore, $ccw(v)$ exists. That is, either $ccw(v) = u$ or $ccw(v) = u'$ such that $\angle tvu' < \angle tvu$. See Figure 2A. That is, no two chains can cross. They can, however, share a common sequence of adjacent vertices. □

Lemma 4. *If C_i and C_j are any two oppositely oriented chains, then $C_j \neq C_i$.*

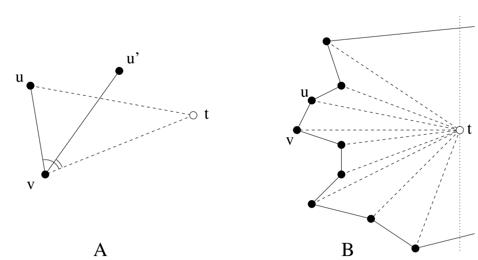

Fig. 2. Illustrations in support of Lemmas 3 **(A)** and 4 **(B)**

Proof. We prove the lemma by contradiction. By definition of $R(C)$ and Lemma 2, no point of C lies in the interior of $R(C)$ and, consequently, for any chains C_i and C_j, $R(C_i) \neq R(C_j)$ if and only if $C_i \neq C_j$. Suppose $C_i = C_j$, where C_i and C_j are two oppositely oriented chains. Therefore, for every edge $\{u, v\}$ in both C_i and C_j, $u = cw(v)$ and $v = ccw(u)$ (or vice versa). Consequently, no internal vertex on the chains can have an edge into the interior of $R(C_i) = R(C_j)$. See Figure 2B. Since every face is convex, t must have a neighbour in H_s. Therefore, some node on C_i must have a neighbour that is t or that lies in the interior of $R(C_i)$, deriving a contradiction. □

Theorem 1. *Given any convex subdivision G and any vertices $\{s, t\} \subseteq V(G)$, Algorithm OneBit is a predecessor-oblivious geometric local routing algorithm that uses one state bit to determine a sequence of forwarding decisions from s to t in G.*

Proof. Each chain is finite by Lemma 2. In particular, the first chain terminates. Each pair of subsequent chains, C_i and C_j, is complete, except for the last chain which terminates at t. Lemmas 3 and 4 imply that $R(C_j)$ is a proper subset of $R(C_i)$. Consequently, the sequence of chains converges towards ℓ_t. Since the vertices of each chain are vertices of G, of which there are a finite number, the result follows. □

3 Using Predecessor Awareness

We describe a predecessor-aware stateless geometric local routing algorithm, called PredAware(u, v, t), that guarantees delivery on any edge-augmented monotone subdivision.

Let G be an edge-augmented monotone subdivision. We define a partial order $\mathcal{P}$ over the vertex set $V(G)$ as follows. For each $u \in V(G)$, let ℓ_u denote the vertical line through u, let z'_u denote a point on ℓ_u above u, and let H^-_u and H^+_u denote the respective left and right half-planes bounded by ℓ_u. Let the *ith left neighbour* of u, denoted $\text{left}_u(i)$, be the node in $v \in N(u) \cap H^-_u$ that forms the ith smallest convex angle $\angle vuz'_u$. The *parent* of u is its first left neighbour, $\text{left}_u(1)$. Similarly, let the *ith right neighbour* of u, denoted $\text{right}_u(i)$, be the node $v \in N(u) \cap H^+_u$ that forms the ith smallest convex angle $\angle vuz'_u$. See Figure 3A. If u has no left neighbours, then u is a *root*. If u has no right neighbours, then u is a *leaf*. For all nodes u and v, $u = \text{left}_v(i)$ for some i if and only if $v = \text{right}_u(j)$ for some j; in particular, this inverse relationship exists if and only if u and v are neighbours and u lies to the left of ℓ_v. The left neighbour relation, $\prec$ (or equivalently, the right neighbour relation) assigns an orientation to each edge in $E(G)$ such that $u \prec v$ if $\{u, v\} \in E(G)$ and $u_x < v_x$, where a_x denotes the x-coordinate of point a. That is, each (previously undirected) edge $\{u, v\} \in E(G)$, where $u_x < v_x$, is assigned the orientation (u, v). Since x-coordinates belong to a total order, the corresponding directed graph is acyclic, which defines the partial order $\mathcal{P}$ on the vertex set $V(G)$. See Figure 3C.

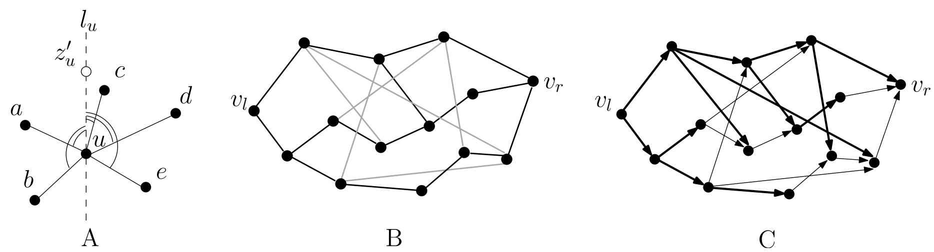

Fig. 3. **(A)** Nodes a and b are the respective 1st and 2nd left neighbours of u. Node a is the parent of u. Nodes c, d, and e are the respective 1st, 2nd, and 3rd right neighbours of u. **(B)** An edge-augmented monotone subdivision G, where the underlying monotone subdivision is shown in bold. **(C)** The corresponding directed acyclic graph on G with parent edges defining a spanning tree in bold.

Lemma 5. *$\mathcal{P}$ defines a single-source (single root) single-sink (single leaf) directed acyclic graph over G.*

Proof. Let M be the monotone subdivision underlying G. Recall that the boundary of the exterior face of any monotone subdivision is monotone. Hence M has a leftmost and a rightmost node, which we denote by v_l and v_r, respectively. Since these nodes remain incident to the outer face even after edge augmentation, they are also the leftmost and rightmost vertices in G. Observe that for each vertex $v \notin \{v_l, v_r\}$ of G, there is a monotone path from v_l to v_r that passes through v. Hence every vertex $v \notin \{v_l, v_r\}$ has a left and a right neighbour.

The existence of a directed cycle in G would imply some edge oriented from right to left. By definition, all edges are oriented from left to right. Therefore, the resulting edge orientations on G determine a directed acyclic graph with a unique source v_l and a unique sink v_r. □

The term "source" as used in Lemma 5 refers to a vertex with in-degree zero and non-zero out-degree. Throughout the rest of the paper, a source node refers to the initial node s from which a message is routed to a target node t.

Algorithm PredAware traverses every edge of G using the partial order $\mathcal{P}$ defined on G. The orientation of each edge and, consequently, the partial order $\mathcal{P}$, can be determined locally by any node u that knows its coordinates and those of its set of neighbours $N(u)$. The route corresponds to a depth-first traversal of a spanning tree of G, resulting in a complete traversal of the graph's vertices (implying guaranteed delivery). Specifically, for each node $u \in V(G)$ that is not the leftmost node (which is the tree root) the edge $\{u, \text{left}_u(1)\} \in E(G)$ (i.e., the edge from u to its parent) corresponds to a tree edge.

Lemma 6. *The set of parent edges defines a spanning tree on G.*

Proof. By Lemma 5, the graph G is a directed acyclic graph with a single source. Each vertex other than the root has a single parent edge, which is an in-edge. The result follows. □

Let u denote the node holding the message, i.e., the node making a forwarding decision. Let $v \in N(u)$ denote the neighbour of u that last forwarded the message to u. At the start of the route (when $u = s$ initially), suppose $v = \varnothing$. The tree traversal algorithm is described in Algorithm 2.

Algorithm 2 PredAware(u, v, t)

Preconditions: u is the node holding the message, $N(u)$ is its set of neighbours, $v \in N(u)$ is u's neighbour that last forwarded the message, t is the target node, $\text{left}_u(i)$ and $\text{right}_u(i)$ denote the ith left and right neighbours of u, respectively.

Postconditions: Forward the message from u to w, where $w \in N(u)$.

1: **if** $t \in N(u)$ **then** (The target node t is adjacent to u.)
2: $w \leftarrow t$ (Forward the message to the target node t.)
3: **else if** $v = \varnothing$ **then** (There is no predecessor: initiate the route.)
4: **if** $\text{right}_u(1) \neq \varnothing$ **then** (u has a right neighbour.)
5: $w \leftarrow \text{right}_u(1)$ (Forward the message to u's first right neighbour.)
6: **else** (u has no right neighbour.)
7: $w \leftarrow \text{left}_u(1)$ (Forward the message to u's parent.)
8: **end if**
9: **else if** $v = \text{left}_u(1)$ **then** (u's parent passed the message into a new subtree of u.)
10: **if** $\text{right}_u(1) \neq \varnothing$ **then** (u has a right neighbour.)
11: $w \leftarrow \text{right}_u(1)$ (Forward the message to u's first right neighbour.)
12: **else** (u has no right neighbour.)
13: $w \leftarrow v$ (Return the message to u's parent.)
14: **end if**
15: **else if** $v = \text{left}_u(i)$ for some $i \geq 2$ **then** (This edge is not in the spanning tree; return the message.)
16: $w \leftarrow v$ (Return the message to the sender v.)
17: **else** (Traversal of u's ith subtree is complete. Traverse u's $(i+1)$st subtree.)
18: **if** $\text{right}_u(i+1) \neq \varnothing$ **then** (u has an $(i+1)$st right neighbour.)
19: $w \leftarrow \text{right}_u(i+1)$ (Forward the message to u's $(i+1)$st right neighbour.)
20: **else if** $\text{left}_u(1) \neq \varnothing$ **then** (u has no $(i+1)$st right neighbour but has a parent.)
21: $w \leftarrow \text{left}_u(1)$ (Forward the message to u's parent.)
22: **else** (u has neither an $(i+1)$st right neighbour nor a parent: u is the root.)
23: $w \leftarrow \text{right}_u(1)$ (Forward the message to u's first right neighbour.)
24: **end if**
25: **end if**

Theorem 2. *Given any edge-augmented monotone subdivision G and any vertices $\{s, t\} \subseteq V(G)$, Algorithm PredAware is a predecessor-aware stateless geometric local routing algorithm that determines a sequence of forwarding decisions from s to t in G. Furthermore, Algorithm PredAware performs a traversal of G.*

Proof. Upon receiving the message from its parent, each node u sequentially forwards the message to each of its right neighbours in clockwise order (see lines 9–14 and 17–24 in Algorithm 2). Upon receiving the message from its ith right neighbour, u forwards the message to its $(i+1)$st right neighbour. If u has no $(i+$

1)st neighbour, then u returns the message to its parent. If u receives the message from a left neighbour other than its parent, then u returns the message to the sender; this indicates that the message was sent along a non-tree edge, and the message is returned immediately. Therefore, the route is extended only when a node u receives the message from its parent, by forwarding the message to each of u's right neighbours. A node's set of right neighbours includes all of its children in the spanning tree on the set of parent edges. Algorithm 2 generates a depth-first recursive traversal of the set of parent edges, which, by Lemma 6, corresponds to a spanning tree of G. The resulting sequence of forwarding decisions is a preorder (depth-first) traversal of the spanning tree. □

Although local algorithms exist for various classes of geometric graphs that construct a spanning tree on which a depth-first tree traversal determines a graph traversal sequence (e.g., [1, 17]), these all require $\Theta(\log n)$ state bits. Algorithm PredAware is stateless. Its ability to guarantee delivery on a monotone subdivision is due to predecessor awareness.

4 Discussion and Directions for Future Research

The algorithms presented in this paper reduce the gap between the classes of geometric graphs on which guaranteed delivery is possible without state bits and those on which it is possible with $O(\log n)$ state bits. Several questions remain to be answered to close this gap. See Table 1.

If nodes have distinct labels, then identifying t in the message requires $\Omega(\log n)$ bits. That data is *static* and is not modified by the routing algorithm. The goal of this research is to minimize the state bits *modified dynamically* by the algorithm. With sufficient state bits, a routing algorithm can record the complete partial graph that has been explored (e.g., $O(n \log n)$ state bits). Braverman's local routing algorithm [8] guarantees delivery on any graph using $\Theta(\log n)$ state bits, regardless of geometry, and without requiring predecessor awareness. In many cases, $\Theta(\log n)$ bits is an allowable cost.

We seek to identify and characterize classes of geometric graphs on which delivery can be guaranteed using few states. In this paper we showed that guaranteed delivery is possible on convex subdivisions using only one state bit and without predecessor awareness, and on edge-augmented monotone subdivisions using only predecessor awareness and no state bits. Routing in planar subdivisions (and other classes of geometric graphs) allows a local routing algorithm to capitalize on the inherent *geometry* to guarantee delivery using fewer states than are necessary on arbitratry graphs. This leads to some natural open questions. Is geometric local routing on planar subdivisions possible using c state bits, where $c \in o(\log n)$ or $c \in O(1)$? On what classes of geometric graphs can a geometric local routing algorithm guarantee delivery using $O(1)$ state bits? On what classes of geometric graphs can a stateless geometric local routing algorithm guarantee delivery using predecessor awareness? With both predecessor awareness and $O(1)$ state bits, can a local routing algorithm guarantee delivery on more general classes of graphs than if it were predecessor-aware and stateless?

Finally, measuring and bounding a local routing algorithm's dilation (worst-case ratio of actual route length to shortest path length) is of interest. Can $O(1)$ dilation be guaranteed on convex subdivisions with one state bit?

References

1. M. de Berg, M. van Kreveld, R. van Oostrum, and M. Overmars. Simple traversal of a subdivision without extra storage. *Int. J. Geog. Inf. Sci.*, 11(4):359–373, 1997.
2. P. Bose, A. Brodnik, S. Carlsson, E. D. Demaine, R. Fleischer, A. López-Ortiz, P. Morin, and I. Munro. Online routing in convex subdivisions. *Int. J. Comp. Geom. & Appl.*, 12(4):283–295, 2002.
3. P. Bose, P. Carmi, and S. Durocher. Bounding the locality of distributed routing algorithms. *Dist. Comp.*, 26(1):39–58, 2013.
4. P. Bose and P. Morin. An improved algorithm for subdivision traversal without extra storage. *Int. J. Comp. Geom. & Appl.*, 12(4):297–308, 2002.
5. P. Bose and P. Morin. Competitive online routing in geometric graphs. *Theor. Comp. Sci.*, 324:273–288, 2004.
6. P. Bose and P. Morin. Online routing in triangulations. *SIAM J. Comp.*, 33(4):937–951, 2004.
7. P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Net.*, 7(6):609–616, 2001.
8. M. Braverman. On ad hoc routing with guaranteed delivery. In *Proc. ACM PODC*, volume 27, page 418, 2008.
9. E. Chavez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, and J. Urrutia. Traversal of a quasi-planar subdivision without using mark bits. *J. Interconn. Net.*, 5(4):395–408, 2004.
10. D. Chen, L. Devroye, V. Dujmović, and P. Morin. Memoryless routing in convex subdivisions: Random walks are optimal. *Comp. Geom.: Theory & Appl.*, 45(4):178–185, 2012.
11. S. Durocher, D. Kirkpatrick, and L. Narayanan. On routing with guaranteed delivery in three-dimensional ad hoc wireless networks. *Wireless Net.*, 16:227–235, 2010.
12. G. G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, Information Sciences Institute, 1987.
13. H. Frey, S. Ruehrup, and I. Stojmenovic. Routing in wireless sensor networks. In S. Misra, I. Woungag, and S. Misra, editors, *Guide to Wireless Ad Hoc Networks*, chapter 4, pages 81–111. Springer-Verlag, 2009.
14. X. Guan. *Face routing in wireless ad-hoc networks.* PhD thesis, Univ. Toronto, 2009.
15. E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proc. CCCG*, volume 11, pages 51–54, 1999.
16. F. Kuhn, R. Wattenhofer, and A. Zollinger. Ad-hoc networks beyond unit disk graphs. In *Proc. ACM DIALM-POMC*, pages 69–78, 2003.
17. P. Morin. *Online routing in geometric graphs.* PhD thesis, Carleton Univ., 2001.
18. J. Urrutia. Handbook wireless net. & mob. comp. In I. Stojmenovic, editor, *Routing with Guaranteed Delivery in Geometric and Wireless Networks*, pages 393–406. John Wiley & Sons, Inc., 2002.
19. M. A. Wahid. Local geometric routing algorithms for edge-augmented planar graphs. Master's thesis, Univ. Manitoba, 2013.