

Gabriel Triangulations and Angle-Monotone Graphs: Local Routing and Recognition

Nicolas Bonichon¹, Prosenjit Bose², Paz Carmi³,
Irina Kostitsyna⁴, Anna Lubiw⁵, and Sander Verdonschot⁶

¹ LaBRI, Univ. Bordeaux, France. bonichon@labri.fr

² School of Computer Science, Carleton University, Ottawa, Canada.
jit@scs.carleton.ca

³ Department of Computer Science, Ben-Gurion University of the Negev, Israel.
carmip@cs.bgu.ac.il

⁴ Université Libre de Bruxelles. irina.kostitsyna@ulb.ac.be

⁵ David R. Cheriton School of Computer Science, University of Waterloo, Canada.
alubiw@uwaterloo.ca

⁶ School of Electrical Engineering and Computer Science, University of Ottawa,
Ottawa, Canada. sander@cg.scs.carleton.ca

Abstract. A geometric graph is *angle-monotone* if every pair of vertices has a path between them that—after some rotation—is *x-* and *y-* monotone. Angle-monotone graphs are $\sqrt{2}$ -spanners and they are increasing-chord graphs. Dehkordi, Frati, and Gudmundsson introduced angle-monotone graphs in 2014 and proved that Gabriel triangulations are angle-monotone graphs. We give a polynomial time algorithm to recognize angle-monotone geometric graphs. We prove that every point set has a plane geometric graph that is *generalized angle-monotone*—specifically, we prove that the half- θ_6 -graph is generalized angle-monotone. We give a local routing algorithm for Gabriel triangulations that finds a path from any vertex s to any vertex t whose length is within $1 + \sqrt{2}$ times the Euclidean distance from s to t . Finally, we prove some lower bounds and limits on local routing algorithms on Gabriel triangulations.

1 Introduction

A geometric graph has vertices that are points in the plane, and edges that are drawn as straight-line segments, with the weight of an edge being its Euclidean length. A geometric graph need not be planar. Geometric graphs that have relatively short paths are relevant in many applications for routing and network design, and have been a subject of intense research. A main scenario is that we are given a point set and must construct a sparse geometric graph on that point set with good shortest path properties.

If the shortest path between every pair of points has length at most t times the Euclidean distance between the points, then the geometric graph is called a *t-spanner*, and the minimum such t is called the *spanning ratio*. Since their introduction by Paul Chew in 1986 [10], spanners have been heavily studied [18].

Besides the existence of short paths, another issue is *routing*—how to find short paths in a geometric graph. One goal is to find paths using *local routing* where the path is found one vertex at a time using only local information about the neighbours of the current vertex plus the coordinates of the destination. A main example of such a method is *greedy routing*: from the current vertex u take any edge to a vertex v that is closer (in Euclidean distance) to the destination than u is. The geometric graphs for which greedy routing succeeds in finding a path are called *greedy drawings*. These have received considerable attention because of their potential ability to replace routing tables for network routing, and because of the noted conjecture of Papadimitriou and Ratajczak [19] (proved in [5, 16]) that every 3-connected planar graph has a greedy drawing. One drawback is that a path found by greedy routing may be very long compared to the Euclidean distance between the endpoints. Of course this is inevitable if the geometric graph has large spanning ratio.

When a geometric graph is a t -spanner, we can ideally hope for a local routing algorithm that finds a path whose length is at most k times the Euclidean distance between the endpoints, for some k , where, of necessity, $k \geq t$. The maximum ratio, k , of path length to Euclidean distance is called the *routing ratio*. For example, the Delaunay triangulation, which is a t -spanner for $t \leq 1.998$ [21], permits local routing with routing ratio $k \leq 5.90$ [7]. It is an open question whether the spanning ratio and routing ratio are equal, though there is a provable gap for L_1 -Delaunay triangulations [7] and TD-Delaunay triangulations [9].

Other “good” paths. Recently, a number of other notions of “good” paths in geometric graphs have been investigated. Alamdari et al. [2] introduced *self-approaching* graphs, where any two vertices s and t are joined by a *self-approaching path*—a path such that a point moving continuously along the path from s to any intermediate destination r on the path always gets closer to r in Euclidean distance. In an *increasing-chord* graph, this property also holds for the reverse path from t to s . The self-approaching path property is stronger than the greedy path property in two ways: it applies to every intermediate destination r , and it requires that continuous motion (not just the vertices) along the path to r always gets closer to r . The significance of the stronger property is that self-approaching and increasing-chord graphs have bounded spanning ratios of 5.333 [15] and 2.094 [20], respectively. An important characterization is that a path is self-approaching if and only if at each point on the path, there is a 90° wedge that contains the rest of the path [15].

Angelini et al. [4] introduced *monotone drawings*, where any two vertices s and t are joined by a path that is monotone in some direction. This is a natural desirable property, but not enough to guarantee a bounded spanning ratio.

Angle-monotone paths. In this paper we explore properties of another class of geometric graphs with good path properties. These are the *angle-monotone graphs* which were first introduced by Dehkordi, Frati, and Gudmundsson [12] as a tool to investigate increasing-chord graphs. (We note that Dehkordi et al. [12] did not give a name to their graph class.)

A polygonal path with vertices v_0, v_1, \dots, v_n is β -monotone for some angle β if the vector of every edge (v_i, v_{i+1}) lies in the closed 90° wedge between $\beta - 45^\circ$ and $\beta + 45^\circ$. (In the terminology of Dehkordi et al. [12] this is a θ -path.) In particular, an x - y -monotone path (where x and y coordinates are both non-decreasing along the path) is a β -monotone path for $\beta = 45^\circ$ (measured from the positive x -axis). A path is *angle-monotone* if there is some angle β for which it is β -monotone. To visualize this, note that a path is angle-monotone if and only if it can be rotated to be x - y -monotone. An angle-monotone path is a special case of a self-approaching path where the wedges containing the rest of the path all have the same orientation. See Figure 1. This implies that an angle-monotone path is also angle-monotone when traversed in the other direction, and thus, has the increasing-chord property. Observe that angle-monotone paths have spanning ratio $\sqrt{2}$ —this is because x - y -monotone paths do.

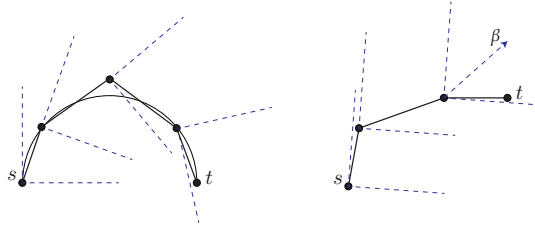


Fig. 1. The difference between a self-approaching st path (left) with 90° wedges each containing the rest of the path, and an angle-monotone path (right) where the 90° wedges all have the same orientation β .

A geometric graph is *angle-monotone* if for every pair of vertices u, v , there is an angle-monotone path from u to v . Note that the angle β may be different for different pairs u, v . Dehkordi et al. [12] introduced angle-monotone graphs, and proved that they include the class of Gabriel triangulations (triangulations with no obtuse angle). Their main goal was to prove that any set of n points in the plane has a planar increasing-chord graph with $O(n)$ Steiner points and $O(n)$ edges. Given their result that Gabriel graphs are increasing chord, this follows from a result of Bern et al. [6] that any point set can be augmented with $O(n)$ points to a point set whose Delaunay triangulation is Gabriel.

The notion of angle-monotone graphs can be generalized to wedges of angle γ different from 90° . (A precise definition is given below.) We call these *angle-monotone graphs with width γ* , or *generalized angle-monotone graphs*. For $\gamma < 180^\circ$, they still have bounded spanning ratios.

Results. The main themes we explore are: Which geometric graphs are angle-monotone? Can we create a sparse (generalized) angle-monotone graph on any given point set? Do angle-monotone graphs permit local routing?

Our first main result is a polynomial time algorithm to test if a geometric graph is angle-monotone. This is significant because it is not known whether increasing chord graphs can be recognized in polynomial time (or whether the problem is NP-hard). Our algorithm extends to generalized angle-monotone graphs for any width $\gamma < 180^\circ$.

Our next result is that for any point set in the plane, there is a plane geometric graph on that point set that is angle-monotone with width 120° . In particular, we prove that the *half- θ_6 -graph* has this property. Width 90° cannot always be achieved because it would imply spanning ratio $\sqrt{2}$ which is known to be impossible for some point sets, as discussed below under Further Background.

The rest of the paper is about local routing algorithms, where we concentrate on a subclass of angle-monotone graphs, namely the Gabriel triangulations. We give a local routing algorithm for Gabriel triangulations that achieves routing ratio $1 + \sqrt{2} \approx 2.41$. This is better than the best known routing ratio for Delaunay triangulations of 5.90 [7]. Also, our algorithm is simpler. The algorithm succeeds, i.e. finds a path to the destination, for any triangulation, and we prove that the algorithm has a bounded routing ratio for triangulations with maximum angle less than 120° . For Delaunay triangulations, we prove a lower bound on the routing ratio of 5.07, but leave as an open question whether the algorithm ever does worse. Finally, we give some lower bounds on the routing ratio of local routing algorithms on Gabriel triangulations, and we prove that no local routing algorithm on Gabriel triangulations can find self-approaching paths.

As is clear from this outline, we leave many interesting open questions, some of which are listed in the Conclusions section.

Further Background. The standard Delaunay triangulation is not self-approaching in general [2], and therefore not angle-monotone.

The *Gabriel graph* of point set P is a graph in which for every edge (u, v) the circle with diameter uv contains no points of P . A Gabriel graph that is a triangulation is called a *Gabriel triangulation*. Any Gabriel triangulation is a Delaunay triangulation. Observe that a triangulation is Gabriel if and only if it has no obtuse angles. Not every point set has a Gabriel triangulation, e.g. three points forming an obtuse triangle.

There are several results on constructing self-approaching/increasing-chord graphs on a given set of points. Alamdari et al. [2] constructed an increasing chord network of linear size using Steiner points, and Dehkordi et al. [12] improved this to a plane network. It is an open question whether every point set admits a plane increasing-chord graph without adding Steiner points. However, for the more restrictive case of angle-monotone graphs, the answer is no: any angle-monotone graph has spanning ratio $\sqrt{2}$ but there is a point set (specifically, the vertices of a regular 23-gon) for which any planar geometric graph has spanning ratio at least 1.4308 [13]. An earlier example was given by Mulzer [17].

Preliminaries and Definitions. A polygonal path with vertices v_0, v_1, \dots, v_n is β -monotone with width γ for some angles β and γ with $\gamma < 180^\circ$ if the vector of every edge (v_i, v_{i+1}) lies in the closed wedge of angle γ between $\beta - \frac{\gamma}{2}$ and $\beta + \frac{\gamma}{2}$. When we have no need to specify β , we say that the path is

angle-monotone with width γ , or “generalized angle-monotone”. A path that is generalized angle-monotone is a *generalized self-approaching path* [1] and thus has bounded spanning ratio depending on γ [1]. But in fact, we can do better:

Observation 1 [proof in Appendix A] *The spanning ratio of an angle-monotone path with width $\gamma < 180^\circ$ is at most $1/\cos \frac{\gamma}{2}$.*

A geometric graph is *angle-monotone with width γ* if for every pair of vertices u, v , there is an angle-monotone path with width γ from u to v . When we have no need to specify γ , we say that the graph is “generalized angle-monotone”.

Note that in an angle-monotone path (with width 90°) the distances from v_0 to later vertices form an increasing sequence. Furthermore, any β -monotone path from u to v lies in a rectangle with u and v at opposite corners and with two sides at angles $\beta \pm 45^\circ$, and the union of such rectangles over all $\beta \in [0, 360^\circ)$ forms the disc with diameter uv . (See Figure 5 in Appendix.) This implies:

Lemma 1. *Any angle-monotone path from u to v lies inside the disc with diameter uv .*

2 Recognizing Angle-monotone Graphs

In this section we give an $O(nm^2)$ time algorithm to test if a geometric graph with n vertices and m edges is angle-monotone. The idea is to look for angle-monotone paths from a node s to all other nodes, and then repeat over all choices of s . For a given source vertex s , the algorithm explores nodes u in non-decreasing order of their distance from s . At each vertex u we store information to capture all the possible angles β for which there is a β -monotone path from s to u . We show how to propagate this information along an edge from u to v .

We begin with some notation. We will measure angles counterclockwise from the positive x -axis, modulo 360° . To any ordered pair u, v of vertices (points) of our geometric graph we associate the vector $v - u$ and we denote its angle by $\alpha(u, v)$. If S is a set of angles that lie within a wedge of angle less than 180° , then we define the *minimum* of S to be the most clockwise angle, and the *maximum* of S to be the most counter-clockwise angle. More formally, α is the *minimum* of S if for any other $\beta \in S$, $\beta - \alpha \in [0, 180^\circ)$, and similarly for *maximum*.

Although there may be exponentially many angle-monotone paths from s to u , each such path has two extreme edges. More precisely, if P is an angle-monotone path from s to u , then the angles, $\alpha(e), e \in P$, lie in a 90° wedge, and so this set has a minimum and maximum that differ by at most 90° . We will store a list of all such min-max pairs with vertex u . Each pair defines a wedge of at most 90° . Since each pair is defined by two edges, there are at most $O(m^2)$ such pairs (though we will show below that we only need to store $O(m)$ of them).

The algorithm starts off by looking at every edge (s, u) and adding the pair $(\alpha(s, u), \alpha(s, u))$ to u 's list. Then the algorithm explores vertices $u \neq s$ in non-decreasing order of their distance from s . To explore vertex u , consider each edge (u, v) and each pair $(\alpha(e), \alpha(f))$ stored with u , and update the list of pairs for

vertex v as follows. If $\alpha(u, v)$ is within 90° of $\alpha(e)$ and within 90° of $\alpha(f)$ then add to v 's list the pair $(\min\{\alpha(u, v), \alpha(e)\}, \max\{\alpha(u, v), \alpha(f)\})$.

If ever the algorithm tries to explore a vertex that has no pairs stored with it, then halt—the graph is not angle-monotone. To justify correctness we prove:

Lemma 2. *When the algorithm has explored all the vertices closer to s than v , then there exists an angle-monotone path from s to v with extreme edges e and f if and only if the pair $(\alpha(e), \alpha(f))$ is in v 's list.*

Proof. The proof is by induction on the distance from s to v .

For the “only if” direction, let P be an angle-monotone path from s to v with extreme edges e and f , and let u be the penultimate vertex of P . The subpath of P from s to u is an angle-monotone path. Suppose its extreme edges are e' and f' where $e = e'$ or $f = f'$ or both. Now, u is closer to s so by induction the pair $(\alpha(e'), \alpha(f'))$ is in u 's list. Because P is angle-monotone, $\alpha(u, v)$ is within 90° of $\alpha(e')$ and $\alpha(f')$. Thus the update step applies. During the update step we add the angle $\alpha(u, v)$ to the pair $(\alpha(e'), \alpha(f'))$, which gives the pair $(\alpha(e), \alpha(f))$. Thus we add the pair $(\alpha(e), \alpha(f))$ to v 's list.

For the “if” direction, suppose that the pair $(\alpha(e), \alpha(f))$ is in v 's list. This pair was added to v 's list because of an update from some vertex u closer to s applied to some pair $(\alpha(e'), \alpha(f'))$ in u 's list. By induction, there exists an angle-monotone path P from s to u with extreme edges e' and f' , and because the update is only performed when $\alpha(u, v)$ is within 90° degrees of $\alpha(e')$ and $\alpha(f')$ therefore the edge (u, v) can be added to P to produce an angle-monotone path with extreme edges e and f . \square

To improve the efficiency of the algorithm we observe that it is redundant to store at a vertex v a pair whose wedge contains the wedge of another pair. Therefore, we only need to store $O(m)$ pairs at each vertex, at most one pair whose first element is $\alpha(e)$ for each edge e . We can simply keep with each vertex v a vector indexed by edges e , in which we store the minimal pair $(\alpha(e), \alpha(f))$ (if any) associated with v so far. Finally, observe that during the course of the algorithm, each edge (u, v) is handled once in an update step. With the refinement just mentioned, handling an edge costs $O(m)$. Therefore the algorithm runs in time $O(m^2)$ for a single choice of s , and in time $O(nm^2)$ overall.

The algorithm can be generalized to recognize angle-monotone graphs of width γ for fixed $\gamma < 180^\circ$. It is no longer legitimate to explore vertices in order of distance from s , since a generalized angle-monotone path will not necessarily respect this ordering. However, we can run the algorithm in phases, where phase i captures all the angle-monotone paths of width γ that start at s and have at most i edges. Since no angle-monotone path can repeat a vertex, there are at most $n - 1$ edges in any angle-monotone path. Thus we need $n - 1$ phases. In each phase, for each directed edge (u, v) we update each pair $(\alpha(e), \alpha(f))$ stored at u as follows. If $\alpha(u, v)$ is within γ of $\alpha(e)$ and within γ of $\alpha(f)$ then add to v 's list the pair $(\min\{\alpha(u, v), \alpha(e)\}, \max\{\alpha(u, v), \alpha(f)\})$. In this way, each of the $n - 1$ phases takes time $O(m^2)$, so the total run-time of the algorithm over all choices of s becomes $O(n^2m^2)$.

3 A Class of Generalized Angle-Monotone Graphs

In this section we show that every point set in the plane has a plane geometric graph that is angle-monotone with width 120° . In particular, we will prove that the *half- θ_6 -graph* has this property. As noted in the Introduction, there are point sets for which no plane graph is angle-monotone with width 90° . It is an open question to narrow this gap and find the minimum angle γ for which every point set has a plane graph that is angle-monotone with width γ (and thus spanning ratio $1/\cos \frac{\gamma}{2}$).

We first define the half- θ_6 -graph. For each point $u \in P$, partition the plane into 60° cones with apex u , with each cone defined by two rays at consecutive multiples of 60° from the positive x -axis. Label the cones $C_0, C_1, C_2, C_3, C_4,$ and C_5 in clockwise order around u , starting from the cone containing the positive y -axis. See Figure 2(a).

For two vertices u and v the *canonical triangle* T_{uv} is the triangle bounded by: the cone of u that contains v ; and the line through v perpendicular to the bisector of that cone. See Figure 2(b). Notice that if v is in an even cone of u , then u is in an odd cone of v . We build the half- θ_6 -graph as follows. For each vertex u and each even $i = 0, 2, 4$, add the edge uv provided that v is in the C_i cone of u and T_{uv} is empty. We call v the *C_i -neighbour* of u . For simplicity, we assume that no two points lie on a line parallel to a cone boundary, guaranteeing that each vertex connects to exactly one vertex in each even cone. Hence the graph has at most $3n$ edges in total. The half- θ_6 -graph is a type of Delaunay triangulation where the empty region is an equilateral triangle in a fixed orientation as opposed to a disk [8]. It can be computed in $O(n \log n)$ time [18].

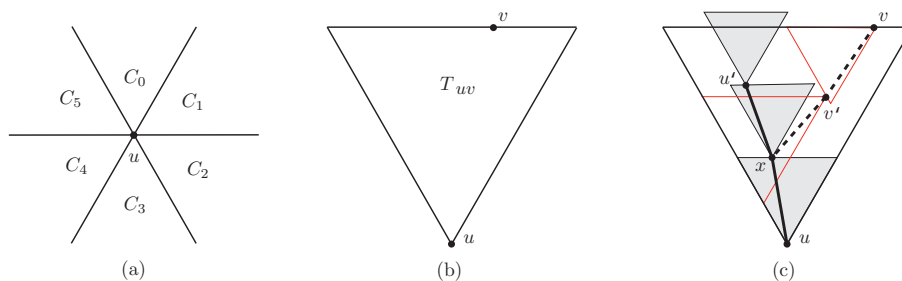


Fig. 2. (a) 6 cones originating from point u , (b) Canonical triangle T_{uv} , (c) path σ_u (solid) with its empty canonical triangles shaded, path σ_v (dashed) and their common vertex x .

To prove angle-monotonicity properties of the half- θ_6 -graph, we use an idea like the one used by Angelini [3]. His goal was to show that every abstract triangulation has an embedding that is monotone, i.e. angle-monotone with width 180° . (The same result was obtained in [14] with a different proof.) Angelini did this by showing that the Schnyder drawing of any triangulation is monotone,

and in fact, upon careful reading, his proof shows that any Schnyder drawing is angle-monotone with the smaller width 120° . Schnyder drawings are a special case of half- θ_6 -graphs [8] so it is not surprising that Angelini’s proof idea extends to the half- θ_6 -graph in general.

Theorem 1. *The half- θ_6 -graph is angle-monotone with width 120° .*

Proof. We must prove that for any points u and v , there is an angle-monotone path from u to v of width 120° . Assume without loss of generality that v is in the C_0 cone of u . See Figure 2(b).

Our path from u to v will be the union of two paths, each of which is angle-monotone of width 60° . We begin by constructing a path σ_u from u in which each vertex is joined to its C_0 neighbour. This is a β -monotone path of width 60° for $\beta = 90^\circ$. If the path contains v we are done, so assume otherwise. Let u' be the last vertex of the path that lies in T_{uv} . Note that v cannot lie in the C_0 cone of u' . Let S be the subpath of σ_u from u to u' , together with the C_0 cone of u' . Then S separates T_{uv} into two parts. Suppose that v lies in the right-hand part (the other case is symmetric). See Figure 2(c).

Next, construct a path σ_v from v in which each vertex is joined to its C_4 neighbour. This is a β -monotone path of width 60° for $\beta = 210^\circ$.

We now claim that σ_u and σ_v have a common vertex x . Then as our final path from u to v we take the portion of σ_u from u to x followed by the portion of σ_v backwards from x to v . Since the reverse of σ_v is β -monotone with width 60° for $\beta = 30^\circ$, the final path is β -monotone with width 120° for $\beta = 60^\circ$.

It remains to prove that x exists. Let v' be the last vertex of σ_v that lies strictly to the right of S . Let u'' be the last vertex of σ_u that lies below v' . We claim that u'' is the C_4 neighbour of v' , and thus that u'' provides our vertex x . Let T be the empty canonical triangle from u'' to its C_0 -neighbour (or the empty C_0 cone of u'' in case u'' has no C_0 -neighbour). First note that u'' is in the C_4 cone of v' —otherwise v' would be in T . Next note that $T_{v'u''}$ is empty—otherwise v would have a C_4 -neighbour that is in T or is to the right of S . \square

Theorem 1 implies that the spanning ratio of the half- θ_6 -graph is 2, which was already known [11]. The best routing ratio achievable for the half- θ_6 -graph is $5/\sqrt{3} \approx 2.887$ [9]. (This was the first proved separation between spanning ratio and routing ratio.) Since angle-monotone paths of width 120° have spanning ratio 2, this implies that no local routing algorithm can compute angle-monotone paths with width 120° on the half- θ_6 -graph.

4 Local Routing in Gabriel Triangulations

In this section we give a simple local “angle” routing algorithm that finds a path from s to t in any triangulation. Like previous algorithms, the path walks only along edges of triangles that intersect the line segment st . The novelty is that the next edge of the path is chosen based on angles relative to the vector st .

The details of the algorithm are in Section 4.1. In Section 4.2 we prove that the algorithm has routing ratio $1 + \sqrt{2}$ on Gabriel graphs, and discuss its behaviour on Delaunay triangulations. In Section 4.3 we give lower bounds on the routing and competitive ratios of local routing algorithms on Gabriel graphs.

4.1 Local Angle Routing

Our algorithm is simple to describe: Suppose we want a route from s to t in a triangulation. Orient st horizontally, t to the right. Suppose we have reached vertex p . Consider the last (rightmost) triangle that is incident to p and intersects the line segment st . The triangle has two edges incident to p . Of these two edges, take the one that has the minimum angle to the horizontal ray from p to the right. See Figure 3. Pseudo-code can be found below in Algorithm 1. Note that in the pseudo-code, the angle test is equivalently replaced by two tests, identifying steps of type A and B for easier case analysis. For an example of a path computed by the algorithm, see Figure 4. Observe that the algorithm always succeeds in finding a route from s to t because it always advances rightward in the sequence of triangles that intersect line segment st .

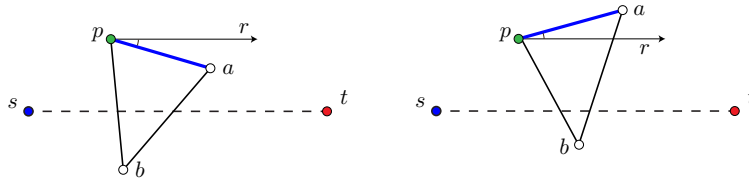


Fig. 3. Local routing from s to t . At vertex p , with pab being the rightmost triangle incident to p that intersects line segment st , we route from p to a because the (unsigned) angle apr is less than angle bpr . A step of type A is shown on the left and a step of type B on the right.

4.2 Analysis of the Algorithm

In this section we will prove that the above algorithm has routing ratio exactly $1 + \sqrt{2}$ on Gabriel triangulations, which have maximum angle at most 90° . In the last part of the section we generalize the analysis to triangulations with a larger maximum angle, and we show that the routing ratio is at least 5.07 on Delaunay triangulations.

The intuition for bounding the routing ratio on Gabriel triangulations is to replace each segment of the route by the most extreme segment possible. See Figure 4. Any step of type B is replaced by a 45° segment plus a horizontal segment. Any step of type A is replaced by a vertical segment plus a horizontal segment. Vertical segments are the bad ones, but each vertical must be preceded

Algorithm 1: Local angle routing

```

1  $p \leftarrow s$ 
2 while  $p \neq t$  do
3   Let  $T = pab$  be the rightmost triangle containing  $p$  that intersects segment
    $st$ , with  $p$  and  $a$  on the same side of line  $st$ .
4   if  $a$  is closer to line  $st$  than  $p$  then                                /* step of type A */
5      $p \leftarrow a$ 
6   else                                                                    /* step of type B */
7     if  $|\text{slope}(pa)| \leq |\text{slope}(pb)|$  then
8        $p \leftarrow a$ 
9     else
10       $p \leftarrow b$ 

```

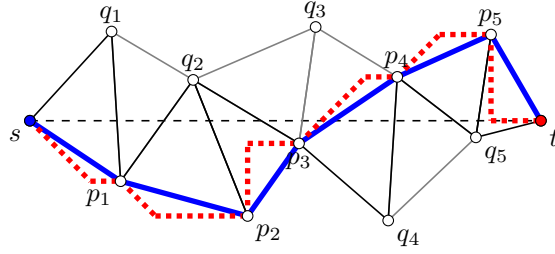


Fig. 4. Example of route computed by Algorithm 1 (heavy blue path). In dotted red, a longer route obtained by replacing each segment of the route by the most extreme angle. Both routes are within $(1 + \sqrt{2})$ of $\|st\|$.

by 45° segments, which means that instead of travelling 1 unit horizontally (the optimum route) we have travelled $\sqrt{2}$ along a 45° segment plus 1 vertically, giving us the $1 + \sqrt{2}$ ratio. We now give a more formal proof.

For each edge $e = (p_i, p_{i+1})$ of the path, let $d_x(e) = \|x(p_i) - x(p_{i+1})\|$ and $d_y(e) = \|y(p_i) - y(p_{i+1})\|$. Let A (resp. B) be the set of edges of the path where the algorithm makes a step of type A (resp. type B). (Context will distinguish edge sets from steps.) Let $x_B = \sum_{e \in B} d_x(e)$ and $x_A = \sum_{e \in A} d_x(e)$.

Lemma 3. *On any Gabriel triangulation the path computed by Algorithm 1 is x -increasing.*

Proof. Let us show that each step is x -increasing. Consider a step from p , with a and b as defined in Algorithm 1. Assume without loss of generality that p and a are above line st and b is below. Since T is the last triangle incident to p that intersects st , the clockwise ordering of T is pab . Refer to Figure 3.

If the algorithm takes a step of type B then a is above p (in y coordinate) and b is below p . Since $\angle bpa \leq 90^\circ$, thus $x(a)$ and $x(b)$ are greater than $x(p)$. If the algorithm takes a step of type A then since b is below st and a is above st and $\angle bap \leq 90^\circ$, thus $x(a)$ is greater than $x(p)$. \square

Theorem 2. *On any Gabriel triangulation, Algorithm 1 has a routing ratio of $1 + \sqrt{2}$ and this bound is tight.*

Proof. We first bound $\sum_{e \in B} \|e\|$. Observe that each edge in B forms an angle with the horizontal line through p that is at most 45° . Thus $\sum_{e \in B} d_y(e) \leq x_B$ and $\sum_{e \in B} \|e\| \leq \sqrt{2}x_B$.

We next bound $\sum_{e \in A} \|e\|$. Observe that edges in A move us closer to the line st , and must be balanced by previous steps (of type B) that moved us farther from the line st . This implies that $\sum_{e \in A} d_y(e) \leq \sum_{e \in B} d_y(e) \leq x_B$ (where the last step comes from the first observation). Since $\|e\| \leq d_x(e) + d_y(e)$, thus $\sum_{e \in A} \|e\| \leq x_A + \sum_{e \in A} d_y(e) \leq x_A + x_B$.

Putting these together, the length of the path is bounded by $\sum_{e \in A} \|e\| + \sum_{e \in B} \|e\| \leq x_A + x_B + \sqrt{2}x_B \leq (1 + \sqrt{2})(x_A + x_B)$. Finally, by Lemma 3, $x_A + x_B = \|st\|$, so this proves that the routing ratio is at most $(1 + \sqrt{2})$.

An example to show that this analysis is tight is given in Appendix B. \square

We conclude this section with two results on the behaviour of the routing algorithm on other triangulations. Proofs are deferred to Appendix B.

Theorem 3. *In a triangulation with maximum angle $\alpha < 120^\circ$ Algorithm 1 has a routing ratio of $(\sin \alpha + \sin \frac{\alpha}{2}) / \sin \frac{3\alpha}{2}$ and this bound is tight.*

Theorem 4. *The routing ratio of Algorithm 1 on Delaunay triangulation is greater than 5.07.*

We believe that the routing ratio of Algorithm 1 on Delaunay triangulations is close to 5.07, but leave that as an open question. We remark that Algorithm 1 is different from the generalization of Chew's Routing Algorithm for Delaunay triangulations [10] (cf. the algorithm described in [7]).

4.3 Limits of Local Routing Algorithms on Gabriel Triangulations

In this section we prove some limits on local routing on Gabriel triangulations. Proofs are deferred to Appendix B.

A routing algorithm on a geometric graph G has a *competitive ratio* of c if the length of the path produced by the algorithm from any vertex s to any vertex t is at most c times the length of the shortest path from s to t in G , and c is the minimum such value. (Recall that the routing ratio compares the length of the path produced by the algorithm to the Euclidean distance between the endpoints. Thus the competitive ratio is less than or equal to the routing ratio.)

A routing algorithm is *k-local* (for some integer constant $k > 0$) if it makes forwarding decisions based on: (1) the k -neighborhood in G of the current position of the message; and (2) limited information stored in the message header.

Theorem 5. *Any k -local routing algorithm on Gabriel triangulations has routing ratio at least 1.4966 and competitive ratio at least 1.2687.*

Although Gabriel triangulations are angle-monotone [12], Theorem 5 shows that no local routing algorithm can compute angle-monotone paths since that would give routing ratio $\sqrt{2}$. The following theorem tells us that even less constrained paths cannot be computed locally:

Theorem 6. *There is no k -local routing algorithm on Gabriel triangulations that always finds self-approaching paths.*

5 Conclusions

We conclude this paper with some open questions.

1. What is the minimum angle γ for which every point set has a plane geometric graph that is angle-monotone with width γ (and thus has spanning ratio $1/\cos \frac{\gamma}{2}$)? We proved $\gamma \leq 120^\circ$, and it is known that $\gamma > 90^\circ$.
2. Is there a local routing algorithm with bounded routing ratio for any angle-monotone graph? Any increasing-chord graph?
3. We bounded the routing ratio of our local routing algorithm on triangulations based on the maximum angle in the triangulation, but how does this relate to the property of being generalized angle-monotone? If a triangulation has bounded maximum angle, is it generalized angle-monotone? The only thing known is that maximum angle 90° implies angle-monotone with width 90° [12].
4. Is the standard Delaunay triangulation generalized angle-monotone? In particular, proving that the Delaunay triangulation is angle-monotone with width strictly less than 120° would provide a different proof that the Delaunay triangulation has spanning ratio less than 2 [21]. It is known that the Delaunay triangulation is not angle-monotone with width 90° (see Section 1).
5. How does our local routing algorithm behave on standard Delaunay triangulations? We proved a lower bound of 5.07 on the routing ratio. We believe the routing ratio is close to this value, but have no upper bound.

Acknowledgements

This work was begun at the CMO-BIRS Workshop on Searching and Routing in Discrete and Continuous Domains, October 11–16, 2015. We thank the other participants of the workshop for many good ideas and stimulating discussions. We thank an anonymous referee for helpful comments.

Funding acknowledgements: A.L. thanks NSERC (Natural Sciences and Engineering Council of Canada). S.V. thanks NSERC and the Ontario Ministry of Research and Innovation. N.B. thanks French National Research Agency (ANR) in the frame of the “Investments for the future” Programme IdEx Bordeaux - CPU (ANR-10-IDEX-03-02). I.K. was supported in part by the NWO under project no. 612.001.106, and by F.R.S.-FNRS.

References

1. Aichholzer, O., Aurenhammer, F., Icking, C., Klein, R., Langetepe, E., Rote, G.: Generalized self-approaching curves. *Discrete Applied Mathematics* 109(1–2), 3–24 (2001)
2. Alamdari, S., Chan, T.M., Grant, E., Lubiw, A., Pathak, V.: Self-approaching graphs. In: Didimo, W., Patrignani, M. (eds.) *Proc. Graph Drawing (GD)*, LNCS, vol. 7704, pp. 260–271. Springer (2013)
3. Angelini, P.: Monotone drawings of graphs with few directions. In: *6th Int. Conf. Information, Intelligence, Systems and Applications (IISA)*. pp. 1–6. IEEE (2015)
4. Angelini, P., Colasante, E., Battista, G.D., Frati, F., Patrignani, M.: Monotone drawings of graphs. *J. Graph Algorithms Appl.* 16(1), 5–35 (2012)
5. Angelini, P., Frati, F., Grilli, L.: An algorithm to construct greedy drawings of triangulations. *J. Graph Algorithms Appl.* 14(1), 19–51 (2010)
6. Bern, M., Eppstein, D., Gilbert, J.: Provably good mesh generation. In: *Proc. 31st Symp. on Foundations of Computer Science (FOCS)*. pp. 231–241. IEEE (1990)
7. Bonichon, N., Bose, P., De Carufel, J.L., Perković, L., Van Renssen, A.: Upper and lower bounds for online routing on Delaunay triangulations. In: Finocchi, I. (eds.) *Proc. 23rd European Symp. on Algorithms (ESA)*. LNCS, vol. 9294, pp. 203–214. Springer (2015)
8. Bonichon, N., Gavaille, C., Hanusse, N., Ilcinkas, D.: Connections between theta-graphs, Delaunay triangulations, and orthogonal surfaces. In: Thilikos, D.M. (ed.) *Proc. 36th Int. Workshop Graph Theoretic Concepts in Computer Science (WG)*. LNCS, vol. 6410, pp. 266–278 (2010)
9. Bose, P., Fagerberg, R., van Renssen, A., Verdonschot, S.: Optimal local routing on Delaunay triangulations defined by empty equilateral triangles. *SIAM J. Comput.* 44(6), 1626–1649 (2015)
10. Chew, L.P.: There is a planar graph almost as good as the complete graph. In: *Proc. 2nd Annual Symp. Computational Geometry (SoCG)*. pp. 169–177 (1986)
11. Chew, L.P.: There are planar graphs almost as good as the complete graph. *J. Computer and System Sciences* 39(2), 205–219 (1989)
12. Dehkordi, H.R., Frati, F., Gudmundsson, J.: Increasing-chord graphs on point sets. *J. Graph Algorithms Appl.* 19(2), 761–778 (2015)
13. Dumitrescu, A., Ghosh, A.: Lower bounds on the dilation of plane spanners (2015), <http://arxiv.org/pdf/1509.07181v3.pdf>
14. He, X., He, D.: Monotone drawings of 3-connected plane graphs. In: Bansal, N., Finocchi, I. (eds.) *Proc. 23rd European Symp. on Algorithms (ESA)*. LNCS, vol. 9294, pp. 729–741. Springer (2015)
15. Icking, C., Klein, R., Langetepe, E.: Self-approaching curves. *Math. Proc. Cambridge Philosophical Society* 125, 441–453 (1995)
16. Leighton, T., Moitra, A.: Some results on greedy embeddings in metric spaces. *Discrete Comput. Geom.* 44, 686–705 (2010)
17. Mulzer, W.: Minimum Dilation Triangulations for the Regular n -Gon. Master’s thesis, Freie Universität Berlin (2004)
18. Narasimhan, G., Smid, M.: *Geometric Spanner Networks*. Cambridge University Press (2007)
19. Papadimitriou, C.H., Ratajczak, D.: On a conjecture related to geometric routing. *Theor. Comput. Sci.* 344, 3–14 (2005)
20. Rote, G.: Curves with increasing chords. *Math. Proc. Cambridge Philosophical Society* 115, 1–12 (1994)

21. Xia, G.: The stretch factor of the Delaunay triangulation is less than 1.998. SIAM J. Comput. 42(4), 1620–1659 (2013)

A Omitted Proofs for Section 1

Proof (Proof of Observation 1). In the worst case we travel the two equal sides of an isosceles triangle with base length 1 and two angles of $\gamma/2$. If ℓ is the side length, the ratio is 2ℓ , and we have $\cos \frac{\gamma}{2} = \frac{1}{2}/\ell$. Thus the ratio is $1/\cos \frac{\gamma}{2}$. \square

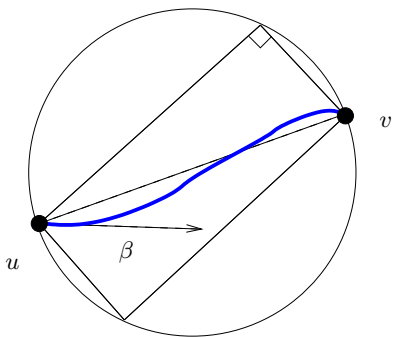


Fig. 5. Illustration of Lemma 1. A β -monotone path (in blue) inside the rectangle with both sides at angles $\beta \pm 45^\circ$. This rectangle lies inside the disc of diameter uv .

B Omitted Proofs for Section 4

Proof (Proof of Theorem 2). To complete the proof, we show an example for which our algorithm gives a routing ratio of $1 + \sqrt{2}$. Consider the configuration shown in Figure 6. It is a Gabriel triangulation and the route computed by the algorithm is as shown. Observe that the size of the leftmost circle can be made arbitrarily small compared to $\|st\|$. Hence, when $s = (0, 0)$ and $t = (1, 0)$, the route can be arbitrarily close to the polyline $s \rightarrow (1, 1) \rightarrow t$. Thus we can build a point set such that the length of the computed route is as close to $1 + \sqrt{2}$ as we want. \square

Proof (Proof sketch for Theorem 3). Following the intuitive justification for the routing ratio of Algorithm 1 on Gabriel triangulations, lengthen the route by replacing each segment of the route by the most extreme segment possible. Any step of type B is replaced by a segment at angle $\frac{\alpha}{2}$ plus a horizontal segment. Any step of type A is replaced by a segment at angle α plus a horizontal segment. In all cases angles are measured from the forward horizontal. See Figure 7. Segments of type A are the bad ones, but each such segment must be preceded

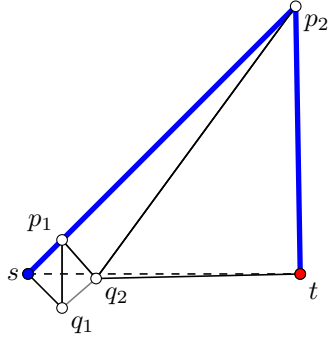


Fig. 6. Example that gives a lower bound on the routing ratio of our routing algorithm on Gabriel triangulations. The route found by the algorithm is drawn as a heavy blue path.

by angle $\frac{\alpha}{2}$ segments, which means that instead of travelling 1 unit horizontally (the optimum route) we have travelled on a segment of angle $\frac{\alpha}{2}$ and then on a segment of angle $-\alpha$ (both angles measured w.r.t the forward horizontal). Let these segments have lengths ℓ_1 and ℓ_2 respectively. In the triangle formed by these three segments, the ℓ_1 segment is opposite angle α , the ℓ_2 segment is opposite angle $\frac{\alpha}{2}$ and the unit horizontal is opposite angle $180^\circ - \frac{3\alpha}{2}$. We need $180^\circ - \frac{3\alpha}{2} > 0$, i.e. $\alpha < 120^\circ$. By the sine law, $\ell_1 = \sin \alpha / \sin \frac{3\alpha}{2}$ and $\ell_2 = \sin \frac{\alpha}{2} / \sin \frac{3\alpha}{2}$. Thus the distance travelled is $\ell_1 + \ell_2 = (\sin \alpha + \sin \frac{\alpha}{2}) / \sin \frac{3\alpha}{2}$.

To show that the bound is tight, we generalize the example of Figure 6. The resulting example is shown in Figure 8. \square

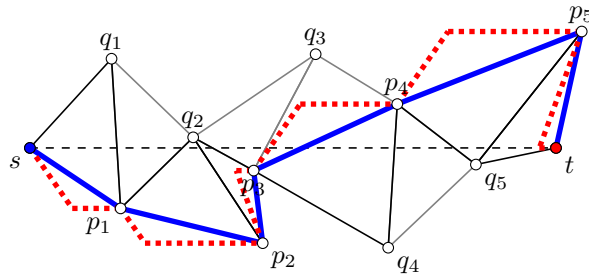


Fig. 7. Intuition for general routing.

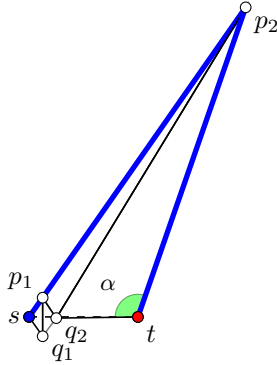


Fig. 8. The worst case situation for general routing.

Proof (Proof of Theorem 4). Let us explain the example of Figure 9. This Delaunay triangulation is defined in the following way: The first triangle sp_1q_1 is such that the slope of line sp_1 is slightly smaller than the slope of sq_1 , so we route to p_1 . Let q_2 be a point on the empty circle C_0 containing s , p_1 and q_1 that is slightly below the x -axis. Let C_1 be the circle that goes through p_1 and q_2 such that the tangent of C_1 at q_2 is horizontal. Let p_2 be a point on C_1 such that the slope of p_1p_2 is slightly smaller than the slope of p_1q_2 . We place point t at the rightmost intersection of C_1 and the x -axis, and we place vertices densely on the arc of C_1 between p_2 and t . The route in the example of Figure 9 has length about $5\|st\|$. Moving p_1 closer and closer to s leads to 5.07 as a lower bound on the routing ratio of Algorithm 1 on Delaunay triangulations. \square

Proof (Proof of Theorem 5).

Let us consider the triangulation of Figure 10. This triangulation is defined as follows: all the triangles intersecting the segment st are right triangles. The first one is isosceles and symmetric with respect to the x -axis. Then we have a fan of $2k - 2$ triangles each having a horizontal side and pointing alternately upward and downward. Let q and q' be respectively the upper rightmost and lower rightmost points of this set of triangles. The next triangle $qq'B$ is such that the angle $\angle q'qB = 22.5^\circ$. The point t is on the intersection of the line $q'B$ and the x -axis. We complete the triangulation with two triangles, qBA and ABt having common hypotenuse AB . Finally we make the fan of $2k - 2$ triangles arbitrarily thin and we assume that $\|qq'\| = 2$.

Now let us consider any deterministic k -local routing algorithm. We consider two triangulations: The first is the one described above (and shown in Fig. 10) and the second one is obtained from the first by reflecting over the x -axis the part of the triangulation that lies to the right of qq' . No deterministic k -local routing algorithm computing a path from s to t can distinguish between the two

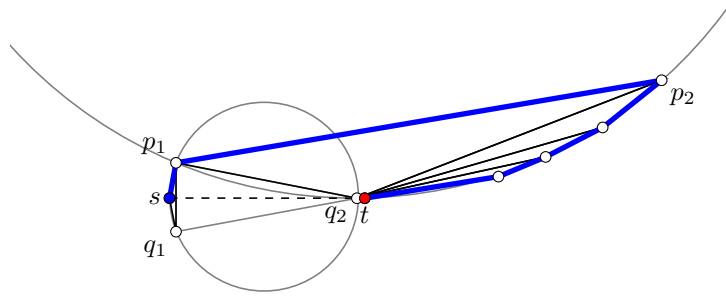


Fig. 9. Example that gives a 5.0 lower bound on the routing ratio of Algorithm 1 on Delaunay triangulations.

point sets until a vertex less than k hops away from q or q' is reached. Let q'' be the vertex k hops away from q or q' that is reached by the algorithm on either triangulation.

Since the fan is arbitrarily thin, q'' can be assumed to be arbitrarily close to q or to q' .

Each case, q or q' , leads to a non-optimal path for one of the point sets; we only consider the first case as the second will follow by symmetry. If q'' is arbitrarily close to q then, for the point set shown in Fig. 10, the shortest paths from q'' to t go through A or B and are of length $\|qB\| + \|Bt\| = 2\|qB\| = 4 \cos(22.5^\circ)$. Moreover $\|sq\| = \sqrt{2}$ and $\|st\| = 1 + 1/\tan(22.5^\circ)$. Hence the length of the complete path computed by the algorithm is at least $\frac{\sqrt{2} + 4 \cos(22.5^\circ)}{1 + 1/\tan(22.5^\circ)} \|st\| \approx 1.496605761$, which proves the routing ratio lower bound. The shortest route from s to t goes through q' and is of length $\|sq'\| + \|q'B\| + \|q't\| = \sqrt{2} + (1 - \cos(45^\circ))/\sin(22.5^\circ) + 2 \cos(22.5^\circ)$. Thus a lower bound on the competitive ratio is $\frac{\sqrt{2} + 4 \cos(22.5^\circ)}{\sqrt{2} + (1 - \cos(45^\circ))/\sin(22.5^\circ) + 2 \cos(22.5^\circ)} \approx 1.268761101$. \square

Proof (Proof sketch for Theorem 6). We apply reasoning as in the previous proof, but this time on the triangulation of Figure 11, where the fat segment qq' represents a fan of $2k - 2$ thin triangles. As before we assume that the algorithm is routing through q (if not we consider the symmetric triangulation with respect to the x -axis). Moving from s to q the distance toward B is not decreasing. Hence a self approaching path that goes through the edge sq cannot go through the vertex B . Hence once at vertex q the only possibility is to use the edge qA . But moving along the edge qA the distance toward t is not decreasing. Hence, there is no self approaching path from s to t that goes through q .

So for any deterministic k -local routing algorithm, there exists a triangulation on which the algorithm will not find a self-approaching path. \square

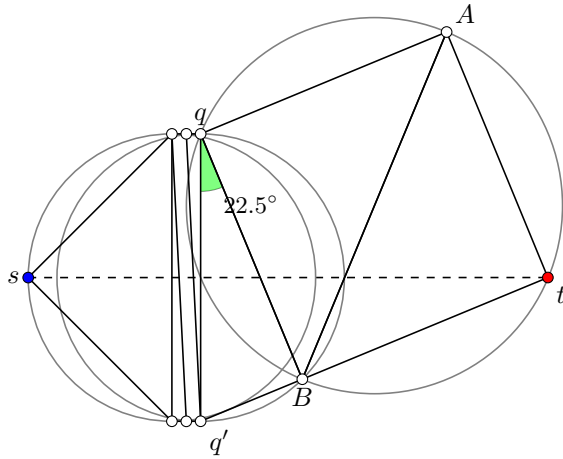


Fig. 10. Example for lower bounds on the routing ratio and competitive ratio of any k -local routing algorithm on Gabriel triangulations.

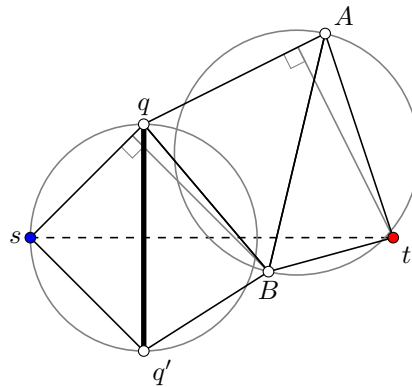


Fig. 11. Example of Gabriel triangulation used to show that no k -local routing algorithm can compute self-approaching paths.