Upper and Lower Bounds for Competitive Online Routing on Delaunay Triangulations*

Nicolas Bonichon[1], Prosenjit Bose[2], Jean-Lou De Carufel[2], Ljubomir Perković[3], and André van Renssen[4]

1 **LaBRI, Bordeaux University, Bordeaux, France**
`bonichon@labri.fr`

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

3 **School of Computing, DePaul University, Chicago, USA**
`lperkovic@cs.depaul.edu`

4 **National Institute of Informatics, Tokyo, Japan**
`andre@nii.ac.jp`

Abstract

Consider a weighted graph G where vertices are points in the plane and edges are line segments. The weight of each edge is the Euclidean distance between its two endpoints. A routing algorithm on 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. If the length of the path is at most c times the Euclidean distance from s to t, we say that the routing algorithm on G has a *routing ratio* of c.

We present an online routing algorithm on the Delaunay triangulation with competitive and routing ratios of 5.90. This improves upon the best known algorithm that has competitive and routing ratio 15.48. The algorithm is a generalization of the deterministic 1-local routing algorithm by Chew on the L_1-Delaunay triangulation. When a message follows the routing path produced by our algorithm, its header need only contain the coordinates of s and t. This is an improvement over the currently known competitive routing algorithms on the Delaunay triangulation, for which the header of a message must additionally contain partial sums of distances along the routing path.

We also show that the routing ratio of any deterministic k-local algorithm is at least 1.70 for the Delaunay triangulation and 2.70 for the L_1-Delaunay triangulation. In the case of the L_1-Delaunay triangulation, this implies that even though there exists a path between two points x and y whose length is at most $2.61|[xy]|$ (where $|[xy]|$ denotes the length of the line segment $[xy]$), it is not always possible to route a message along a path of length less than $2.70|[xy]|$. From these bounds on the routing ratio, we derive lower bounds on the competitive ratio of 1.23 for Delaunay triangulations and 1.12 for L_1-Delaunay triangulations.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling, G.2.2 Graph Theory

Keywords and phrases Delaunay triangulation, online routing

* N. Bonichon partially supported by ANR grant JCJC EGOS ANR-12-JS02-002-01 and LIRCO. P. Bose partially supported by NSERC. A. van Renssen supported by JST, ERATO, Kawarabayashi Large Graph Project.

1 Introduction

Navigation in networks or in graphs is fundamental in computer science. It leads to applications in a number of fields, namely geographic information systems, urban planning, robotics, and communication networks to name only a few. Navigation often occurs in a geometric setting that can be modeled using a *geometric graph* which is defined as a weighted graph G whose vertices are points in the plane and edges are line segments. Let the weight of each edge be the Euclidean distance between its two endpoints. Navigation is then simply the problem of finding a path in G from a source vertex s to a target vertex t. When complete information about the graph is available, numerous path finding algorithms exist for weighted graphs (e.g., Dijkstra's algorithm [17]).

The problem is more challenging when only *local* information is provided. To illustrate this, suppose that a message is traveling along edges of G. We are interested in a routing algorithm that makes forwarding decisions based on limited information related to the current position of the message in G. If the message is currently at vertex v, such information could be limited to the coordinates of v and its neighbors in the graph. Such an algorithm is called *local* or 1-local. More generally, when the coordinates of neighbours that are at most k hops away from v are available, we say that the algorithm is k-*local*. A routing algorithm that uses only local information is called an *online* routing algorithm. Given a constant $c \geq 1$, an online routing algorithm has a *competitive ratio* of c or is c-competitive 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. If the length of the path is at most $c|[st]|$, where $|[st]|$ is the Euclidean length of the line segment $[st]$, we say that the routing algorithm has a *routing ratio* of c. Since $|[st]|$ is a lower bound on the length of the shortest path from s to t in G, the routing ratio is an upper bound on the competitive ratio.

Competitive online routing is not always possible (refer to [4, 19] for instance) and even when it is, a small competitive ratio may not be. In this paper, we are interested in competitive online routing algorithms for classes of geometric graphs that have "good paths". A graph G is a κ-*spanner* (or has a *spanning ratio* of κ) when for any pair of vertices u and v in G, there exists a path in G from u to v with length at most $\kappa|[uv]|$ (see [1, 2, 5, 7, 8, 13, 14, 20] for instance). In several cases, the proof of existence of these paths rely on full knowledge of the graph. Therefore, a natural question is to ask whether we can construct these paths using only local information. We do not have a general answer to that question. Nevertheless, there exist c-competitive online routing algorithms for several of these classes of geometric graphs (see [6, 9, 11, 14, 15, 16]). In this paper, we focus on the most important geometric graph, the Delaunay triangulation.

A Delaunay triangulation is a geometric graph G such that there is an edge between two vertices u and v if and only if there exists a *circle* with u and v on its boundary that contains no other vertex of G. Dobkin et al. [18] were the first to prove that the Delaunay triangulation is a spanner. Xia [20] proved that the spanning ratio of the Delaunay triangulation is at most 1.998 which currently best known the best upper bound on the spanning ratio. The best lower bound, by Xia et al. [21], is 1.593. If *circle* is replaced with *equilateral triangle* in the definition of the Delaunay triangulation, then a different triangulation is defined: the TD-Delaunay triangulation. Chew proved that the TD-Delaunay triangulation [16] is a 2-spanner and that the constant 2 is tight. If we replace *circle* with *square* then yet another triangulation is defined: the L_1- or the L_∞-Delaunay triangulation, depending on the orientation of the square. Bonichon et al. [3] proved that the L_1- and the L_∞-Delaunay triangulations are $\sqrt{4+2\sqrt{2}}$-spanners and that the constant is also tight.

Ideally, we would like the routing ratio to be identical to the spanning ratio. In the case of TD-Delaunay triangulations, Bose et al. [11] found an online routing algorithm that has competitive and routing ratios of $\frac{5}{\sqrt{3}}$. They also showed lower bounds of $\frac{5}{\sqrt{3}}$ on the routing ratio and of $\frac{5}{3}$ on the competitive ratio. In his seminal paper, Chew [16] described an online routing algorithm on the L_1-Delaunay triangulation that has competitive and routing ratios of $\sqrt{10} \approx 3.162$. We show in this paper that there is separation between the spanning ratio and the routing ratio in the case of the L_1 and L_∞-Delaunay triangulations. We show lower bounds of 2.707 on the routing ratio and of 1.122 on the competitive ratio (Theorem 10). In this paper, we also present an online routing algorithm on the Delaunay triangulation that has competitive and routing ratios of 5.90 (Theorem 3). This improves upon the previous best known algorithm that has competitive and routing ratios of 15.48 [9]. Our algorithm is a generalization of the deterministic 1-local routing algorithm by Chew on the L_1-Delaunay triangulation [15] and the TD-Delaunay triangulation [16]. Although the generalization of Chew's routing algorithm to Delaunay triangulation is natural, the analysis of its routing ratio is non-trivial and relies on new techniques. An advantage of Chew's routing algorithm is that it does not require the message header to contain any information other than the coordinates of s and t. All previously known competitive routing algorithms on the Delaunay triangulation [9, 12] require header to store partial sums of distances along the routing path. in the header of the message. See Table 1 for a summary of these results.

Shape	triangle	square	circle
spanning ratio UB	2 [16]	2.61 [3]	1.998 [20]
spanning ratio LB	2 [16]	2.61 [3]	1.593 [21]
routing ratio UB	$5/\sqrt{3} \approx 2.89$ [11]	$\sqrt{10} \approx 3.16$ [15]	$1.185 + 3\pi/2 \approx 5.90$ (Thm 3)
routing ratio LB	$5/\sqrt{3} \approx 2.89$ [11]	2.707 (Thm 10)	1.701 (Thm 9)
competitiveness LB	$5/3 \approx 1.66$ [11]	1.1213 (Thm 10)	1.2327 (Thm 9)

Table 1 Upper and lower bounds on the spanning ratio and the routing ratio on Delaunay triangulations defined by different empty shapes. We also provide lower bounds on the competitiveness of k-local deterministic routing algorithms on Delaunay triangulations.

2 Chew's Routing Algorithm

In this section we present the routing algorithm. This algorithm is a natural adaptation to Delaunay triangulations of Chew's routing algorithm originally designed for L_1-Delaunay triangulations [15] and subsequently adapted for TD-Delaunay triangulations [16].

We consider the Delaunay triangulation defined on a finite set of points P in the plane. In this paper, we denote the source of the routing path by $s \in P$ and its destination by $t \in P$. We assume that an orthogonal coordinate system consisting of a horizontal x-axis and a vertical y-axis exists and we denote by $x(p)$ and $y(p)$ the x- and y-coordinates of any point p in the plane. We denote the line supported by two points p and q by pq, and the line segment with endpoints p and q by $[pq]$. Without loss of generality, we assume that $y(s) = y(t) = 0$ and $x(s) < x(t)$.

When routing from s to t, we consider only (the vertices and edges of) the triangles of the Delaunay triangulation that intersect $[st]$. Without loss of generality, if a vertex (other than s and t) is on $[st]$, we consider it to be slightly above st. Therefore, the triangles that intersect $[st]$ can be ordered from left to right. Notice that all vertices (other than s and t)

from this ordered set of triangles belong to at least 2 of these triangles.

The routing algorithm can be described as follows. When we reach a vertex p_i (initially $p_0 = s$), we consider the *rightmost* triangle T_i that has p_i as a vertex. Let x and y be the other two vertices of T_i and denote by C_i the the circle circumscribing T_i. Let w_i (w as in *west*) be the leftmost point of C_i and let r_i be the rightmost intersection of C_i with $[st]$. The line segment $[w_i r_i]$ splits C_i in two arcs: the *upper* one, defined by the clockwise walk along C_i from w_i to r_i and the *lower* one, defined by the counterclockwise walk along C_i from w_i to r_i. Both arcs include points w_i and r_i. Because T_i is rightmost, x and y cannot both lie on the interior of the same arc so we can assume that x belongs to the upper arc and y belongs to the lower arc. The forwarding decision at p_i is made as follows:

- If p_i belongs to the upper arc, we walk clockwise along C_i until we reach vertex x.
- If p_i belongs to the lower arc, we walk counterclockwise along C_i until we reach y.

If $p_i = w_i$ we apply the first (upper arc) rule.

Once we reach $p_{i+1} = x$ or y, we repeat the process until we reach t. Note that because the two vertices of T_i other than p_{i+1} are not both below or both above line segment $[st]$, T_i must be the leftmost triangle that has p_{i+1} as a vertex. Unless $p_{i+1} = t$, p_{i+1} is a vertex of at least another triangle intersecting $[st]$, so T_i cannot be the rightmost triangle that has p_{i+1} as a vertex.

Figure 1 shows an example of a route computed by this algorithm.

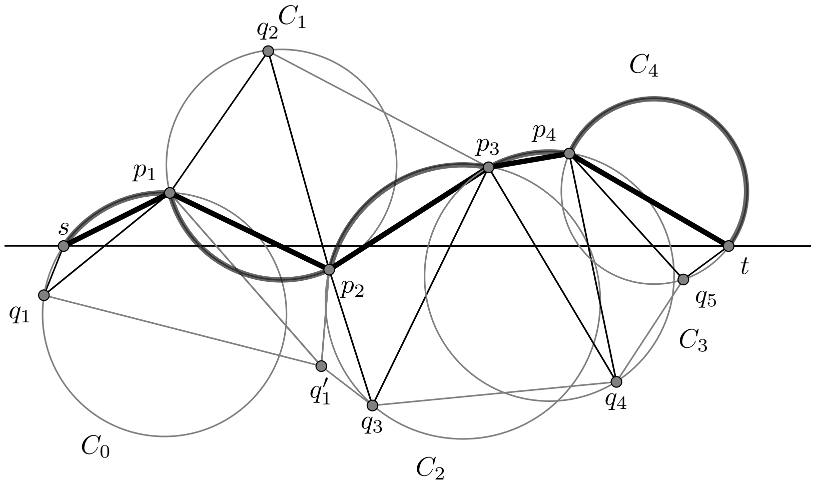

Figure 1 Illustration of Chew's routing algorithm. The empty circles of the *rightmost triangles* are drawn in gray and their edges are drawn in black. The edges of the obtained path and the associated arcs are thicker.

Because the routing decision can always be applied, because the decision is based on the rightmost triangle and progress is made from left to right, and because P is finite, we can conclude that the following results by Chew from [15] extend to Delaunay triangulations. The following is Lemma 2 in [15]:

▶ **Lemma 1.** *The triangles used ($T_0, T_1 \ldots, T_k$) are ordered along $[st]$. Although not all Delaunay triangulation triangles intersecting $[st]$ are used, those used appear in their order along $[st]$.*

In Figure 1 the triangles T_i are drawn with blue edges. The following corollary is in [15] as well:

▶ **Corollary 2.** *The algorithm terminates, producing a path from s to t.*

3 Routing Ratio

In this section, we prove the main theorem of this paper.

▶ **Theorem 3.** *The Chew's routing algorithm on the Delaunay triangulation has a routing ratio of at most* $(1.185043874 + 3\pi/2) \approx 5.89743256$.

As shown in Figure 2, Chew's algorithm has routing ratio at least 5.7282 (see Section 4).

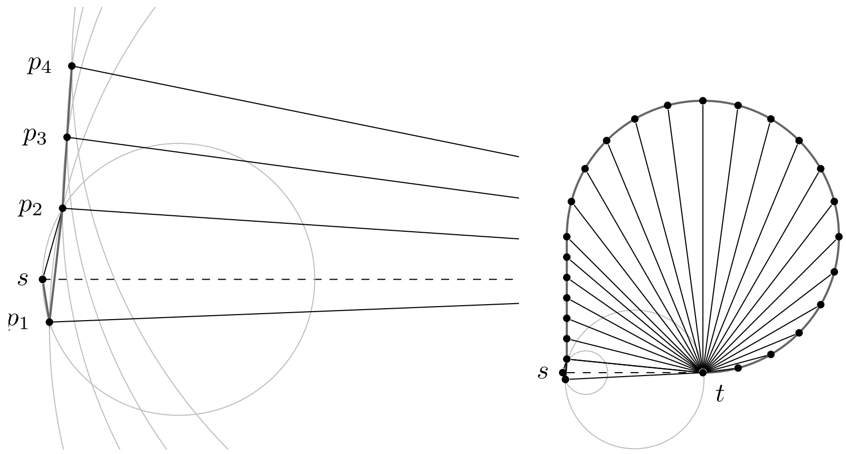

Figure 2 On the right is a Delaunay triangulation that illustrates the lower bound on the routing ratio of Chew's algorithm. The path obtained by the algorithm is shown in dark gray; it has length $5.7282|[st]|$, a bit greater than $(1 + 3\pi/2)|[st]|$. The left image zooms in on what happens close to point s.

We devote this section to the proof of Theorem 3.

3.1 Preliminaries

We start by introducing additional definitions, notations, and structural results about Chew's routing algorithm. Some of the notations are illustrated in Figure 3.

We denote by $|[pq]|$ the Euclidean length of the line segment $[pq]$, and by $|\mathcal{P}|$ the length of a path $\mathcal{P}$ in the plane. Given a path $\mathcal{P}$ from p to q and a path $\mathcal{Q}$ from q to r, $\mathcal{P} + \mathcal{Q}$ denotes the concatenation of $\mathcal{P}$ and $\mathcal{Q}$. We say that the path $\mathcal{P}$ from p to q is *inside* a path $\mathcal{Q}$ that also goes from p to q if the path $\mathcal{P}$ is inside the region delimited by $\mathcal{Q} + [qp]$. Note that if $\mathcal{P}$ is convex and inside $\mathcal{Q}$ then $|\mathcal{P}| \leq |\mathcal{Q}|$. Given a path $\mathcal{P}$ and two points p and q on $\mathcal{P}$, we denote by $\mathcal{P}\langle p, q\rangle$ the sub-path of $\mathcal{P}$ that goes from p to q.

Let $s = p_0, p_1, \ldots, p_k = t$ be the sequence of vertices visited by Chew's routing algorithm. If some p_i other than s or t lies on the segment $[st]$, we can separately analyze the routing ratio of the paths from s to p_i and from p_i to t. We assume, therefore, that no p_i, other than $s = p_0$ and $t = p_k$, lies on segment $[st]$.

For every edge (p_i, p_{i+1}), there is a corresponding oriented arc of C_i used by the algorithm which we refer to as $\mathcal{R}\langle p_i, p_{i+1}\rangle$ (shown in gray in Figures 1 and 3). The orientation (clockwise or counterclockwise) of $\mathcal{R}\langle p_i, p_{i+1}\rangle$ is the orientation taken by the routing algorithm when going from p_i to p_{i+1}. Let $\mathcal{R}$ be the union of these arcs. We call $\mathcal{R}$ the routing path from s to t. The length of the path $s = p_0, p_1, \ldots, p_{k-1}, p_k = t$ along the edges of the

Delaunay triangulation is smaller than the length of $\mathcal{R}$. In the remainder of this section, we analyze the length of $\mathcal{R}$ to obtain an upper bound on the length of the path along the edges of the Delaunay triangulation.

3.2 Worst Case Circles C'_i

In order to bound the length of $\mathcal{R}$, we work with the circles C'_i defined as follows. Let A_1 and A_2 be two circles that go through p_i and p_{i+1} such that A_1 is tangent to $[st]$ and the tangent of A_2 at p_i is vertical. We define C'_i to be A_2 if A_2 intersects $[st]$ twice and A_1 otherwise. Let w'_i be the leftmost vertex of C'_i and O'_i the center of C'_i. In the example of Figure 3, $w'_1 = p_1$ and $w'_3 \neq p_3$.

Note that if $[p_ip_{i+1}]$ crosses $[st]$, then C'_i is A_2. We consider three types of circles C'_i:

- Type A_1: $p_i \neq w'_i$ and $[p_{i-1}p_i]$ does not cross $[st]$.
- Type A_2: $p_i = w'_i$ and $[p_{i-1}p_i]$ does not cross $[st]$.
- Type B: $[p_{i-1}p_i]$ crosses $[st]$.

In Figure 3, $C'_0, C'_1 \dots C'_4$ are respectively of type A_2, B, B, A_1, A_2. Note that if C'_i is of type B, then $p_i = w'_i$. We use the expression "type A" instead of "type A_1 or A_2".

Given two points p, q on C'_i, let $\mathcal{A}_i{<}p, q{>}$ be the arc on C'_i from p to q whose orientation (clockwise or counterclockwise) is the same as the orientation of $\mathcal{R}{<}p_i, p_{i+1}{>}$ around C_i. Notice that $|\mathcal{R}{<}p_i, p_{i+1}{>}| \leq |\mathcal{A}_i{<}p_i, p_{i+1}{>}|$.

Let t_i be the first point p_j after p_i such that $[p_ip_j]$ intersects st. Notice that $t_{k-1} = t$. We also set $t_k = t$. In Figure 3, $t_0 = p_1$, $t_1 = p_2$, $t_2 = p_3$ and $t_3 = t_4 = t_5 = t$. In addition, let $t'_i = (x(t_i), 0)$ and $s_i = (x(w'_i), 0)$.

▶ **Lemma 4.** *For all* $0 < i \leq k$*:*

$$x(s_{i-1}) \leq x(s_i), \tag{1}$$

$$x(s_i) \leq x(t_{i-1}) \leq x(t_i). \tag{2}$$

Proof. We first prove (1). If $p_i = w'_i$ then $x(s_i) = x(p_i)$. Since p_i lies on C'_{i-1} and $x(s_{i-1})$ is the x-coordinate of the leftmost point of C'_{i-1}, we have that $x(s_{i-1}) \leq x(p_i) = x(s_i)$. If $p_i \neq w'_i$ then C'_i is of type A_1. We assume without loss of generality that p_i lies above st. If w'_i is on or in the interior of C'_{i-1} then (1) holds. Otherwise, the rightmost intersection of C'_{i-1} with st must be to the left of the intersection of w'_ip_i and st. This, in turn, implies (1)

We now prove (2). We first observe that $t_{i-1} = p_j$ and $t_i = p_{j'}$ for some $i \leq j \leq j'$. Using inequality (1), we have that $x(s_i) \leq x(s_j) \leq x(p_j) = x(t_{i-1})$, so the first inequality in (2) holds. The second inequality trivially holds when $j = j'$, so we assume otherwise. In that case, $p_j, p_{j+1}, \dots, p_{j'-1}$ must all be on the same side of st. Without loss of generality, we assume that $p_{j'}$ lies above st. This implies that $[p_{j'-1}p_{j'}]$ crosses $[st]$ and therefore C'_{j-1} is of type B. Moreover, $p_{j'-1} = w'_{j'-1}$ and $p_{j'-1}$ lies below st. On the other hand, p_j lies below st and on C'_{j-1}, which is of type B and whose center O'_{j-1} is above st. Note that $p_{j'-1}$ and $p_{j'}$ lie outside of C'_{j-1} and that $x(w'_{j-1}) \leq x(w'_{j'-1})$ (recall that $w'_{j'-1} = p_{j'-1}$). Therefore, if q is the intersection of $\mathcal{A}_{j'-1}{<}p_{j'-1}, p_{j'}{>}$ and st, no point of C'_{j-1} below st and outside of $C'_{j'-1}$ has an x-coordinate larger than $x(q)$. Since $x(q) < x(p_{j'})$, the second inequality in (2) holds. ◀

3.3 Proof of Theorem 3

In this section, we prove our main theorem. Given two points p and q such that $x(p) < x(q)$ and $y(p) = y(q)$, we define the path $\mathcal{S}_{p,q}$ as follows. Let C be the circle above pq that is

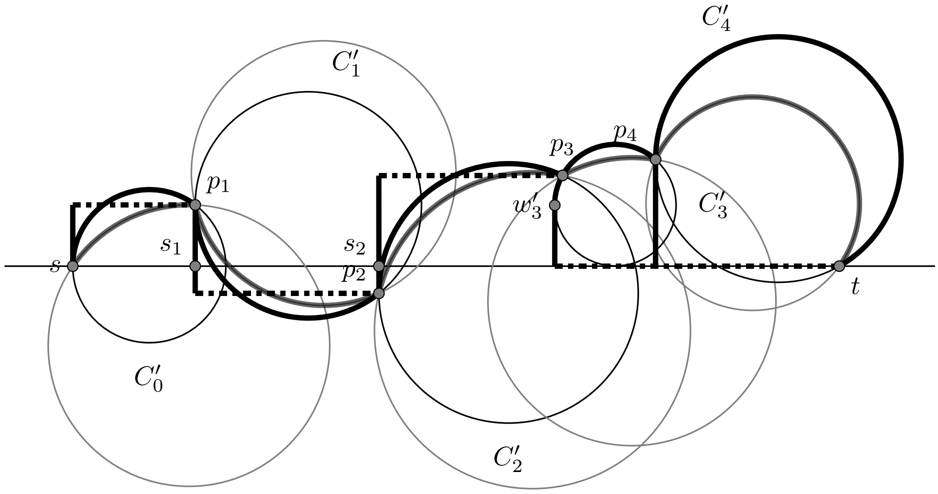

Figure 3 Illustration of Lemma 6 on the example of Figure 1. The empty circles of the Delaunay triangulation and the routing path $\mathcal{R}$ are drawn in gray. Worst cast circles C'_i, paths $\mathcal{P}_i$, and segments of height $|y(t_i)|$ are drawn in black. Lengths $|[t'_{i-1}t'_i]|$ are represented by dashed horizontal segments.

tangent to pq at q and tangent to the line $x = x(p)$ at a point that we denote by p'. The path $\mathcal{S}_{p,q}$ consists of $[pp']$ together with the clockwise arc from p' to q on C. We call $\mathcal{S}_{p,q}$ the *snail curve* from p to q. Note that $|\mathcal{S}_{p,q}| = (1 + 3\pi/2)(x(q) - x(p))$. We also define the path $\mathcal{P}_i$ to be $[s_i w'_i] + \mathcal{A}_i{<}w'_i, p_{i+1}{>}$ for $0 \le i \le k-1$ (see Figure 3).

We start with a lemma that motivates these definitions.

▶ **Lemma 5.** $|\mathcal{P}_{k-1}| \le |\mathcal{S}_{s_{k-1},t}|$

Proof. This follows from the fact that $\mathcal{P}_{k-1}$ from s_{k-1} to t is convex and inside $\mathcal{S}_{s_{k-1},t}$. ◀

The following lemma is the key to proving Theorem 3.

▶ **Lemma 6.** *For all* $0 < i < k$ *and* $\delta = 0.185043874$*,*

$$|\mathcal{P}_{i-1}| + |y(t_{i-1})| \le |\mathcal{P}_i{<}s_i, p_i{>}| + |\mathcal{S}_{s_{i-1},s_i}| + |y(t_i)| + \delta|[t'_{i-1}t'_i]|. \tag{3}$$

Moreover, if C'_{i-1} *is of type A (then* $t_{i-1} = t_i$*), the previous inequality is equivalent to*

$$|\mathcal{P}_{i-1}| \le |\mathcal{P}_i{<}s_i, p_i{>}| + |\mathcal{S}_{s_{i-1},s_i}|. \tag{4}$$

This lemma is illustrated in Figure 3. We first show how to use Lemma 6 to prove Theorem 3, then we prove Lemma 6 in Section 3.4.

Proof of Theorem 3. By Lemma 4, $\sum_{i=1}^{k-1} |[t'_{i-1}t'_i]| < |[st]|$ and $\sum_{i=1}^{k} |\mathcal{S}_{s_{i-1},s_i}| = |\mathcal{S}_{s,t}|$. By summing the $k-1$ inequalities from Lemma 6 and the inequality from Lemma 5, we get

$$\sum_{i=1}^{k} |\mathcal{P}_{i-1}| + |y(t_0)| < \sum_{i=1}^{k-1} |\mathcal{P}_i{<}s_i, p_i{>}| + |\mathcal{S}_{s,t}| + |y(t_{k-1})| + \delta|[st]|.$$

The fact that $t_{k-1} = t$ implies $y(t_{k-1}) = 0$. Therefore, since $\mathcal{P}_{i-1} = \mathcal{A}_{i-1}{<}p_{i-1}p_i{>} + \mathcal{P}_{i-1}{<}s_{i-1}, p_{i-1}{>}$, we have

$$|\mathcal{R}| \leq \sum_{i=1}^{k} \mathcal{A}_{i-1}{<}p_{i-1}p_i{>} < |\mathcal{S}_{s,t}| + \delta|[st]| \leq (1.185043874 + 3\pi/2)|[st]|$$

which completes the proof. ◀

3.4 Proof of the Key Lemma

In this section, we prove Lemma 6. We will make use of the following lemma whose proof is in the Appendix.

▶ **Lemma 7.** *Let $\theta = \angle w'_{i-1}O'_{i-1}p_i$ and $\alpha = \angle w'_iO'_ip_i$ be the angles defined using the orientations $\mathcal{A}_i{<}p_{i-1}, p_i{>}$ and $\mathcal{A}_i{<}p_i, p_{i+1}{>}$, respectively. Then*

$$0 \leq \alpha < \theta < 3\pi/2. \tag{5}$$

Proof of Lemma 6. Notice that if C'_{i-1} is of type A, then $|y(t_{i-1})| = |y(t_i)|$. Hence, in this case, it sufficient to prove

$$|\mathcal{P}_{i-1}| \leq |\mathcal{P}_i{<}s_i, p_i{>}| + |\mathcal{S}_{s_{i-1},s_i}|. \tag{6}$$

For the rest of the proof, we consider three cases depending on the types of C'_{i-1} and C'_i.
• **C'_{i-1} is of type A and C'_i is of type A_2 or B.** In this case, $p_i = w'_i$ from which $x(s_i) = x(p_i)$ follows. Let X be the orthogonal projection of p_i onto $s_{i-1}w'_{i-1}$. Then

$$|[s_{i-1}, X] + \mathcal{S}_{X,p_i}| = |\mathcal{P}_i{<}s_i, p_i{>}| + |\mathcal{S}_{s_{i-1},s_i}|. \tag{7}$$

Since the path $\mathcal{P}_{i-1}$ is convex and inside the path $[s_{i-1}, X] + \mathcal{S}_{X,p_i}$ (see Figure 4), we get

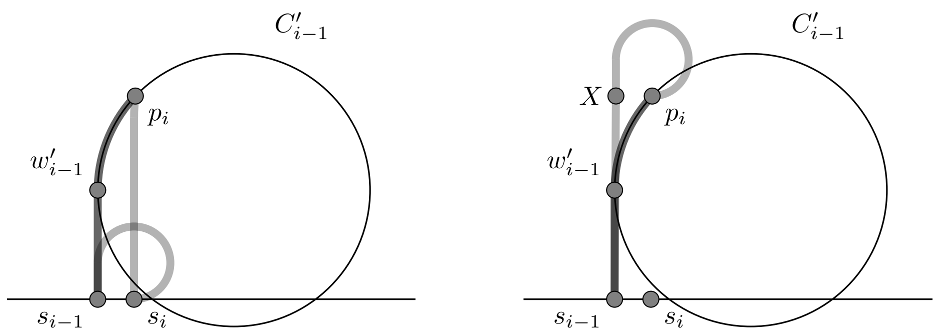

Figure 4 Illustration of the case when C'_{i-1} is of type A and C'_i is of type A_2 or B.

$|\mathcal{P}_{i-1}| \leq |[s_{i-1}, X] + \mathcal{S}_{X,p_i}|$. Applying Inequality (7) completes the proof of this case.
• **C'_{i-1} is of type A and C'_i is of type A_1.** (See Figure 5.) Let b_i the lowest point of C'_i. Let X and Y be respectively the projections of s_{i-1} and b_i on the line $y = y(b_{i-1})$. We consider the snail curve $\mathcal{S}_{s_i,b_i} = \mathcal{P}_i{<}s_i, p_i{>} + \mathcal{A}_i{<}p_i, b_i{>}$ ($\mathcal{A}_i{<}p_i, b_i{>}$ see Figure 5).

Let $\mathcal{G} = \mathcal{S}_{X,Y}$. We have:

$$|\mathcal{P}_i{<}s_i, p_i{>}| + |\mathcal{S}_{s_{i-1},s_i}| + |\mathcal{A}_i{<}p_i, b_i{>}| = |\mathcal{S}_{s_{i-1},b_i}| = |\mathcal{G}|. \tag{8}$$

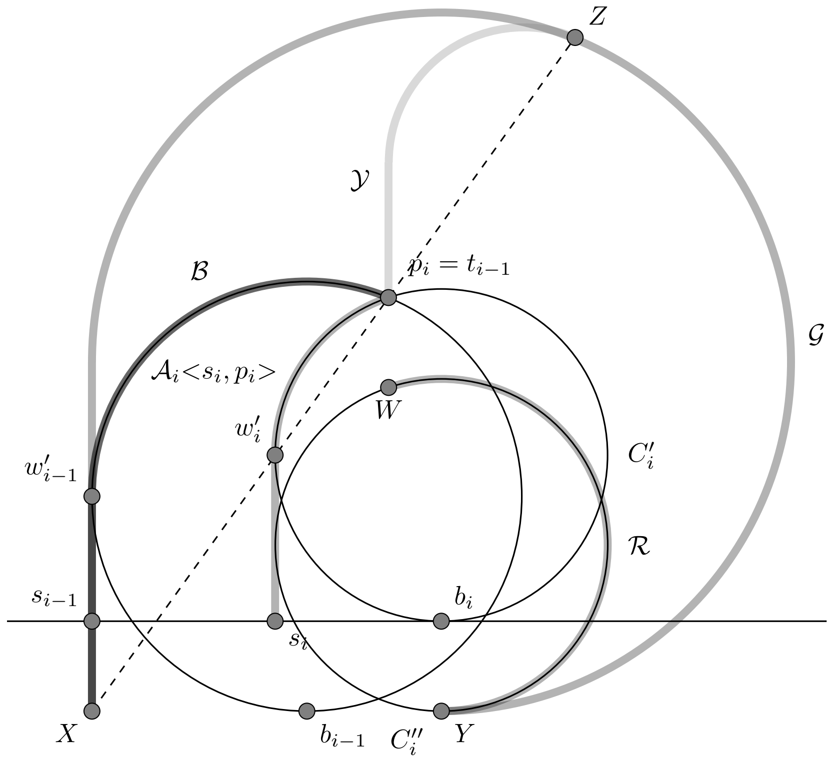

Figure 5 Illustration of the case when C_{i-1}' is of type A and C_i' is of type A_1.

Let $\mathcal{B} = [Xs_{i-1}] + \mathcal{P}_{i-1}$ and $Z \neq X$ be the intersection of Xp_i with $\mathcal{G}$. Denote by $\mathcal{Y}$ the path from p_i to Z that is homothetic to $\mathcal{B}$. Note that $\mathcal{B}$ and $\mathcal{Y}$ are both homothetic to $\mathcal{G}<X, Z>$. Hence,

$$|\mathcal{B} + \mathcal{Y}| = |\mathcal{G}<X, Z>|. \tag{9}$$

Let C_i'' be the curve obtained by translating down C_i' until b_i lies on Y. Denote by $\mathcal{R}$ and W the images of $\mathcal{A}_i<p_i, b_i>$ and p_i by the same translation, respectively. The circle C_i'' is tangent to the circular section of $\mathcal{S}_{s_{i-1},b_i}$ at Y. Moreover, the radius of C_i'' is smaller than the radius of the circular section of $\mathcal{S}_{s_{i-1},b_i}$. Hence, it does not intersect $\mathcal{S}_{s_{i-1},b_i}$. This implies that if $\mathcal{R}$ intersects the line p_iZ, the intersection points must be in $[p_iZ]$. We can show that $\mathcal{R}$ does not intersect $\mathcal{Y}$. Since $\mathcal{A}_i<p_i, b_i>$ is convex and lies inside $[Wp_i] + \mathcal{Y} + \mathcal{G}<Z, Y>$, we have

$$|\mathcal{R}| \leq |[Wp_i] + \mathcal{Y} + \mathcal{G}<Z, Y>|. \tag{10}$$

Summing (9) and (10), and removing $|\mathcal{Y}|$ and $|[b_iY]| = |[Xs_{i-1}]|$ from both sides, we get $|\mathcal{P}_{i-1}| + |\mathcal{A}_i<p_i, b_i>| \leq |\mathcal{G}|$. Using (8) and removing $|\mathcal{A}_i<p_i, b_i>|$ from both sides, we get (6).

• C_{i-1}' **is of type** B **and** C_i' **is of type** A **or** B. In this case, $w_{i-1}' = p_{i-1}$ and $t_{i-1} = p_i \neq t_i$. We consider the case where $|y(p_{i-1}) - y(p_i)| < |y(p_{i-1}) - y(t_i)|$ first (refer to Figure 6). Recall that $\mathcal{P}_{i-1} = \mathcal{A}_{i-1}<p_{i-1}p_i> + [s_{i-1}p_{i-1}]$ and let $\mathcal{P}^*$ be the curve obtained by translating

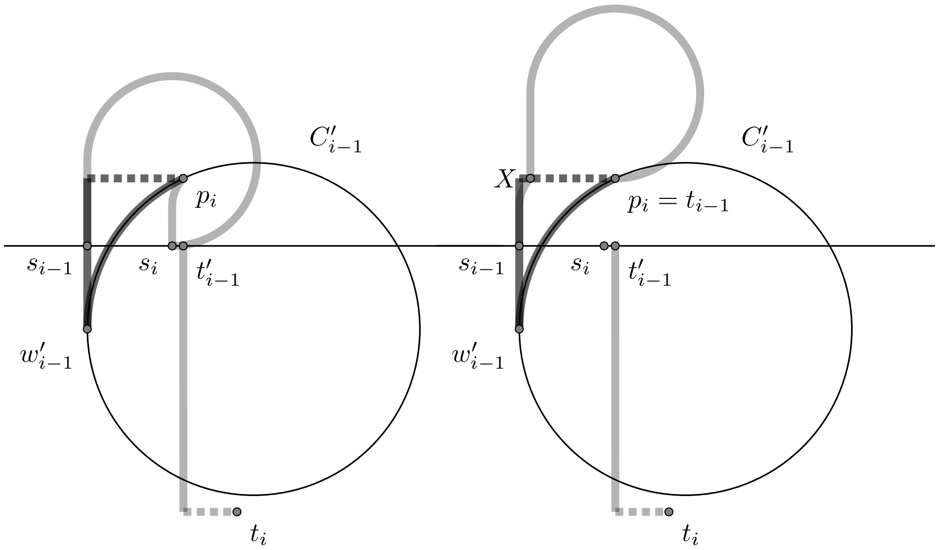

Figure 6 Illustration of the case when C'_{i-1} is of type B and $|y(p_{i-1}) - y(p_i)| < |y(p_{i-1}) - y(t_i)|$.

$\mathcal{P}_i{<}s_i, p_i{>}$ to the left until s_i lies on s_{i-1}. Denote the highest point of $\mathcal{P}^*$ by X. Notice that $x(p_i) - x(X) = x(s_i) - x(s_{i-1})$, $\mathcal{A}_{i-1}{<}p_{i-1}p_i{>}$ is convex and $\mathcal{A}_{i-1}{<}p_{i-1}p_i{>}$ is inside $[p_{i-1}s_{i-1}] + \mathcal{P}^* + \mathcal{S}_{X,p_i}$, which is also convex. Consequently, we get

$$|\mathcal{A}_{i-1}{<}p_{i-1}p_i{>}| \leq |[p_{i-1}s_{i-1}]| + |\mathcal{S}_{s_{i-1},s_i}| + |\mathcal{P}_i{<}s_i, p_i{>}|. \tag{11}$$

Moreover, since $|y(p_{i-1}) - y(p_i)| < |y(p_{i-1}) - y(t_i)|$, we have

$$|[s_{i-1}p_{i-1}]| + y(t_{i-1}) \leq |y(t_i)| - |[s_{i-1}p_{i-1}]|. \tag{12}$$

Summing (11) and (12), we get

$$|\mathcal{P}_{i-1}| + |y(t_{i-1})| \leq |\mathcal{P}_i{<}s_i, p_i{>}| + |\mathcal{S}_{s_{i-1},s_i}| + |y(t_i)|.$$

Notice that we did not need the additional potential $\delta|[t'_{i-1}t'_i]|$ in this case.

For the rest of the proof, we can assume that $|y(p_{i-1}) - y(p_i)| \geq |y(p_{i-1}) - y(t_i)|$. If we assume that p_i lies above st, then t_i must lie below st. The point t_i is outside of C_{i-1}. By Lemma 4, $x(p_i) = x(t_{i-1}) < x(t_i)$. Moreover, all points p on C'_{i-1} or inside of it, and such that $x(p_i) < x(p)$ are in the interior of C_{i-1}. Therefore, t_i is outside of C'_{i-1}.

Recall that by Lemma 7, if $\theta = \angle w'_{i-1}O'_{i-1}p_i$ and $\alpha = \angle w'_iO'_ip_i$, then $0 \leq \alpha < \theta < 3\pi/2$. Without loss of generality, assume that the radius of C'_{i-1} is 1 and the radius of C'_i is R. Then we have $|\mathcal{P}_i{<}s_i, p_i{>}| = (1+\alpha)R$ and $[s_it'_{i-1}] = (1 - \cos(\alpha))R$. Let D be the difference between the left-hand side and the right-hand side of inequality (3). We can write D as

$$\begin{aligned} D &= |\mathcal{S}_{s_{i-1},s_i}| + |\mathcal{P}_i{<}s_i, p_i{>}| + \delta|[t'_{i-1}t'_i]| + |y(t_i)| - |\mathcal{P}_{i-1}| - |y(t_{i-1})| \\ &= (1 + 3\pi/2)(1 - \cos(\theta) - (1 - \cos(\alpha))R) + (1 + \alpha)R + \delta|[t'_{i-1}t'_i]| + |y(t_i)| - \theta - \sin(\theta). \end{aligned}$$

It remains to prove that $D > 0$.

We first consider the case where $\theta \leq \pi/4$, which implies that $\alpha < \pi/4$ as well. Let $p'_i \neq p_i$ be the intersection of C'_{i-1} with the horizontal line through p_i. Since $\theta \leq \pi/4$, we have $x(p'_i) > x(O'_i)$.

Since $|y(p_{i-1}) - y(p_i)| < |y(p_{i-1}) - y(p)|$ for all points p outside of C'_{i-1} such that $x(p_i) \leq x(p) \leq x(p'_i)$, it follows that $x(t_i) > x(p'_i)$. Note that $\angle w'_{i-1}O'_{i-1}p'_i = \pi - \theta$, as illustrated in Figure 7. Since $|[t'_{i-1}t'_i]| > |[t_{i-1}p'_i]| = 2\cos(\theta)$ (recall that $p_i = t_{i-1}$), we have

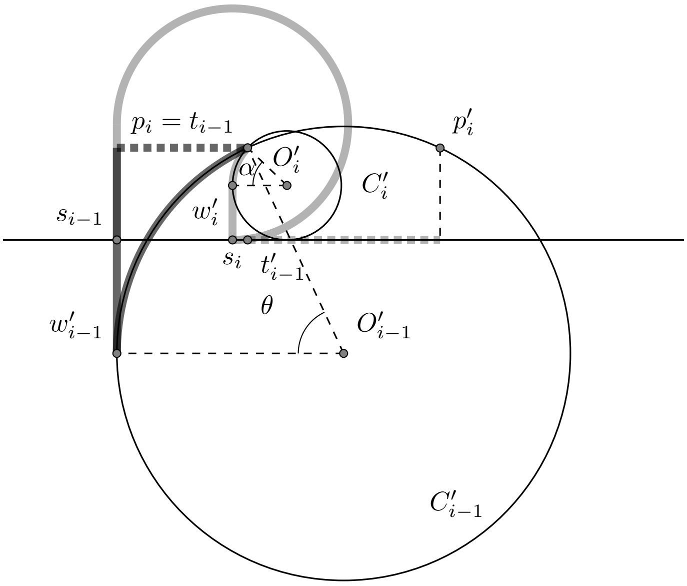

Figure 7 Illustration of the case of when C'_{i-1} is of type B and $|y(p_{i-1})-y(p_i)| \geq |y(p_{i-1})-y(t_i)|$.

$$
\begin{aligned}
D &\geq (1 + 3\pi/2)(1 - \cos(\theta) - (1 - \cos(\alpha))R) + (1 + \alpha)R + 2\delta\cos(\theta) - \theta - \sin(\theta) \\
&\geq R[1 + \alpha - (1 + 3\pi/2)(1 - \cos(\alpha))] + (1 + 3\pi/2)(1 - \cos(\theta)) + 2\delta\cos(\theta) - \theta - \sin(\theta).
\end{aligned}
$$

There exists an $\alpha_0 > \pi/4$ such that $1+\alpha-(1+3\pi/2)(1-\cos(\alpha))$ is positive for all $\alpha \in [0, \alpha_0]$. Therefore, to prove that $D > 0$ (and therefore that inequality (3) holds), it is sufficient to prove that

$$0 \leq (1 + 3\pi/2)(1 - \cos(\theta)) + 2\delta\cos(\theta) - \theta - \sin(\theta).$$

If we take $\delta = 0.185043874$, we can show that this inequality is true using elementary calculus arguments.

To complete the proof, it remains to consider the case where $\theta \in [\pi/4, \pi]$. If $\alpha \leq \alpha_0$, we have $D \geq (1 + 3\pi/2)(1 - \cos(\theta)) - \theta - \sin(\theta)$, which is positive for all $\theta \in [\pi/4, \pi]$. If $\alpha \in [\alpha_0, \pi]$, $1 + \alpha - (1 + 3\pi/2)(1 - \cos(\alpha))$ is negative and decreasing. Thus, since $\alpha \leq \theta$ and $R < 1$, we obtain

$$
\begin{aligned}
D &\geq 1 + \theta - (1 + 3\pi/2)(1 - \cos(\theta)) + (1 + 3\pi/2)(1 - \cos(\theta)) + \delta|[t'_{i-1}t'_i]| + |y(t_i)| - \theta - \sin(\theta) \\
&\geq 1 - \sin(\theta) + \delta|[t'_{i-1}t'_i]| + |y(t_i)|.
\end{aligned}
$$

This lower bound is trivially positive, hence inequality (3) holds in all cases. ◀

4 Lower Bounds

In this section, we provide lower bounds on the routing ratio and the competitive ratio of any k-local routing algorithm on the Delaunay triangulation and the L_1- and L_∞-Delaunay triangulation.

▶ **Theorem 8.** *The routing ratio of Chew's routing algorithm on Delaunay triangulations is at least* 5.7282*.*

Proof. Let C_0 and C_1 be two circles such that the west point of C_0 lies on the x-axis and the west point w_1 of C_1 lies on C_0 and below the x-axis. Let s be the west point of C_0 and let t the rightmost intersection of C_1 with the x-axis. Let $p_1 = w_1$ and p_2 be the intersections of C_0 and C_1. We perturb the configuration such that s lies slightly below the x-axis and p_1 lies slightly above the horizontal line through w_1 (see Figure 2). This implies that the first two edges of the path computed by Chew's algorithm are $[sp_1]$ and $[p_1p_2]$.

Next, we add circles C_i with west point w_i such that t lies on C_i, p_i lies slightly above w_i, and point p_{i+1} lies slightly above p_i. We place circles until t is the lowest point of C_j for some j. Finally, we add points $p_j, p_{j+1}, ..., p_k$ (for some integer k) on the $\mathcal{A}_j{<}p_jt{>}$. We slightly perturb the configuration such that all chords reach t (see Figure 2). Observe that by placing sufficiently many vertices between p_2 and p_j, we create an almost vertical path from p_2 to p_j. The routing path computed by Chew's algorithm tends to $[sp_1]+[p_1p_2]+\mathcal{S}_{p_2,t}$.

We now pick C_0 to be the circle with center at $O_0 = (-0.7652277146, 0)$ and radius 0.2369448832 and we pick C_1 to be the circle with center $O_1 = (0, -0.0320133045)$ and radius 1. This leads to a routing path whose length approaches 11.4660626 as j and k approach infinity. Since the distance between s and t is 2.00166, this implies that the routing ratio of Chew's algorithm is at least 5.7282.[1] ◀

We note that this lower bound is strictly larger that $|\mathcal{S}_{s,t}|/|[st]| = 1 + 3\pi/2$. Next we show that no deterministic k-local routing algorithm on Delaunay triangulations can have routing ratio less than 1.7018.

▶ **Theorem 9.** *There exist no deterministic k-local routing algorithm on Delaunay triangulations with routing ratio at most 1.7018 or that is* 1.2327*-competitive.*

Proof. Let C_0 be the circle with center $O_0 = (1, 0)$ and radius 1 and let C_1 be the circle with center $O_1 = (1.4804533538, 0.2990071425)$ and radius 1.2285346394. Let s be the leftmost intersection of C_0 with the x-axis and let t be the rightmost intersection of C_1 with the x-axis. Let points A and B be the intersections of C_0 and C_1 such that A lies above the x-axis. Let A' and B' be the vertices on O_0A and O_0B outside C_0 such that $|[AA']| = |[BB']| = 0.0718725166$ (see Figure 8). These points (referred to as *shield vertices* by Bose et al. [10]) will ensure that no shortcuts between points on C_0 and points on C_1 are created.

Next, we place points densely on the arcs of C_0 and C_1 that are not contained in the other circle. To ensure that AA' and BB' are edges of the Delaunay triangulation, we leave small gaps along the arcs close to each shield vertex. Since all points of C_0 (respectively C_1) are co-circular, any planar triangulations of them is a valid Delaunay triangulation. Next, we perturb the points in order to both break co-circularity and to choose the triangulation of the interior of the circles (see Figure 8). We compute a triangulation where the shortest paths between t and any point of point set P does not use any chord except $[AB]$. Let q be the point on C_0 such that qA' is the tangent of C_0 at q and let q' the reflection of q over the x-axis.

Now let us consider any deterministic k-local routing algorithm. We consider two point sets: The first one is described above (see Figure 8) and the second one is obtained from

[1] GeoGebra files that describe the 2 first examples presented in this section are available at the following url: `http://www.labri.fr/perso/bonichon/DelaunayRouting/`

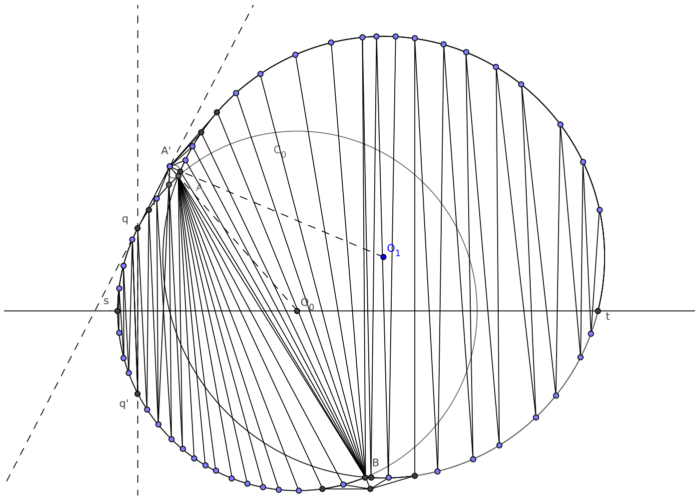

Figure 8 One of the two point sets used to provide a lower bound on the routing ratio on Delaunay triangulations.

the first one by reflecting the part of the point set that lies to the right of qq' over the x-axis. No deterministic k-local routing algorithm can distinguish between the two instances before it reaches q'', a k-neighbor of q or q', depending on if the routing algorithm followed the arc towards q or q'. Since vertices are densely placed on the arc of C_0, q'' is arbitrary close to q or q'. Hence, any deterministic k-local algorithm must route to the same vertex q'' in both instances. Either q'' is close to q or q'. In one of the two instances this leads to a non-optimal route: On the instance of Figure 8, the length of the arc from s to q is 0.477998 and the shortest paths from q to t go via vertex A and have length 4.0693551467. The Euclidean distance from s to t is 2.6720456033. Hence, on one of the two instances the length of the computed path is at least $1.7018 \cdot |[st]|$. The shortest path between s and t in this configuration is of length 3.6888, this configuration gives a lower bound on the competitiveness of any routing algorithm of 1.2327. ◀

Finally, we look at the routing ratio and competitiveness of any deterministic k-local routing algorithm on the L_1- and L_∞-Delaunay triangulations.

▶ **Theorem 10.** *There exists no deterministic k-local routing algorithm for the L_1- and L_∞-Delaunay triangulations that has routing ratio less than $(2+\sqrt{2}/2) \approx 2.7071$ or that is $\frac{2+\sqrt{2}/2}{1+\sqrt{2}} \approx 1.1213$-competitive.*

Proof. The proofs for the L_1- and L_∞-Delaunay triangulations are very similar as one can be created from the other by rotating the point set. We present only the point set of the L_∞-Delaunay triangulation. First, we place the source vertex s at the origin. Given some values $\epsilon > 0$ and $d < 1$, we then place k vertices close to point $q = (\epsilon, (2-\sqrt{2})/4)$ and $q' = (2\epsilon, -(2-\sqrt{2})/4)$ (see Figure 9). Next, we place a vertex A at $(3\epsilon, 1-d+\epsilon)$, vertex B at $(1+2\epsilon, 1-d)$, and destination t at $(1+3\epsilon, 0)$. Finally, we place vertices densely on $[Bt]$ and on $[q'B']$, where B' is picked such that q', B, t, B' forms a parallelogram. As ϵ approaches 0, the resulting L_∞-Delaunay triangulation approaches the one shown in Figure 9.

Now, consider any deterministic k-local routing algorithm. We consider two point sets: The first one is described above (see Figure 9) and the second one is obtained from the first one by reflecting the part of the point set that lies to the right of qq' over the x-axis. Since

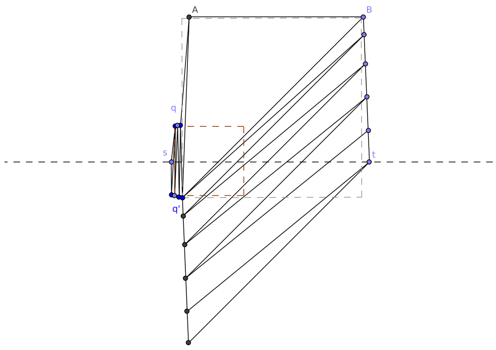

Figure 9 One of the two point sets used to provide a lower bound on the routing ratio on L_∞-Delaunay triangulations.

there are k points between s and q and between s and q', the only information the k-local routing algorithm has before getting close to q or q' consists of the vertices to the left of qq'. If the first step made by the algorithm is towards a vertex close to q, we consider the point set shown in Figure 9. Otherwise, we consider the reflected point set instead. We note that $|[sq]| = (2-\sqrt{2})/4$ and that the shortest paths from q to t have length $\min(|[qA]+[AB]+[Bt]|, |[qq']+[q'B]+[Bt]|) = \min(1-d-(2-\sqrt{2})/4+1+1-d, (2-\sqrt{2})/2+\sqrt{2}+1-d)$. If we pick $d = (2-\sqrt{2})/4)$, the length of both paths is equal to $3-3(2-\sqrt{2})/4$. This leads to a path from s to t of length $2+\sqrt{2}/2$. Since the Euclidean distance between s and t approaches 1 as ϵ approaches 0, this gives a lower bound on the routing ratio of any deterministic k-local routing algorithm on the L_1- and L_∞-Delaunay triangulations.

Finally, we observe that on the point set shown in Figure 9, the length of shortest path from s to t is $1+\sqrt{2}$. This gives a lower bound of $\frac{2+\sqrt{2}/2}{1+\sqrt{2}}$ on the competitive ratio of any deterministic k-local routing algorithm. ◀

References

1 L. Barba, P. Bose, J. D. Carufel, A. van Renssen, and S. Verdonschot. On the stretch factor of the Theta-4 graph. In *Algorithms and Data Structures - 13th International Symposium, WADS 2013, London, ON, Canada, August 12-14, 2013. Proceedings*, pages 109–120, 2013.

2 L. Barba, P. Bose, M. Damian, R. Fagerberg, W. L. Keng, J. O'Rourke, A. van Renssen, P. Taslakian, S. Verdonschot, and G. Xia. New and improved spanning ratios for Yao graphs. In *30th Annual Symposium on Computational Geometry, SOCG'14, Kyoto, Japan, June 08 - 11, 2014*, page 30, 2014.

3 N. Bonichon, C. Gavoille, N. Hanusse, and L. Perković. The stretch factor of L_1- and L_∞-Delaunay triangulations. In *Proceedings of the 20th Annual European Symposium on Algorithms (ESA)*, volume 7501 of *Lecture Notes in Computer Science*, pages 205–216. Springer, 2012.

4 P. Bose, A. Brodnik, S. Carlsson, E. D. Demaine, R. Fleischer, A. López-Ortiz, P. Morin, and J. I. Munro. Online routing in convex subdivisions. *Int. J. Comput. Geometry Appl.*, 12(4):283–296, 2002.

5 P. Bose, P. Carmi, S. Collette, and M. H. M. Smid. On the stretch factor of convex Delaunay graphs. *JoCG*, 1(1):41–56, 2010.

6 P. Bose, P. Carmi, M. Couture, M. H. M. Smid, and D. Xu. On a family of strong geometric spanners that admit local routing strategies. *Comput. Geom.*, 44(6-7):319–328, 2011.

7 P. Bose, J. D. Carufel, P. Morin, A. van Renssen, and S. Verdonschot. Optimal bounds on Theta-graphs: More is not always better. In *Proceedings of the 24th Canadian Conference on Computational Geometry, CCCG 2012, Charlottetown, Prince Edward Island, Canada, August 8-10, 2012*, pages 291–296, 2012.

8 P. Bose, M. Damian, K. Douïeb, J. O'Rourke, B. Seamone, M. H. M. Smid, and S. Wuhrer. $\pi/2$-angle Yao graphs are spanners. *Int. J. Comput. Geometry Appl.*, 22(1):61–82, 2012.

9 P. Bose, J.-L. De Carufel, S. Durocher, and P. Taslakian. Competitive online routing on Delaunay triangulations. In *14th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, 2014.

10 P. Bose, L. Devroye, M. Löffler, J. Snoeyink, and V. Verma. Almost all Delaunay triangulations have stretch factor greater than $\pi/2$. *Computational Geometry*, 44(2):121–127, 2011.

11 P. Bose, R. Fagerberg, A. van Renssen, and S. Verdonschot. Competitive routing in the half-θ_6-graph. In *Proceedings of the 23rd ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)*, pages 1319–1328, 2012.

12 P. Bose and P. Morin. Online routing in triangulations. *SIAM J. Comp.*, 33(4):937–951, 2004.

13 P. Bose, P. Morin, A. van Renssen, and S. Verdonschot. The θ_5-graph is a spanner. *Comput. Geom.*, 48(2):108–119, 2015.

14 P. Bose, A. van Renssen, and S. Verdonschot. On the spanning ratio of Theta-graphs. In *Algorithms and Data Structures - 13th International Symposium, WADS 2013, London, ON, Canada, August 12-14, 2013. Proceedings*, pages 182–194, 2013.

15 L. P. Chew. There is a planar graph almost as good as the complete graph. In *Proceedings of the Second Annual Symposium on Computational Geometry (SoCG)*, pages 169–177, 1986.

16 L. P. Chew. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2):205–219, 1989.

17 E. W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, 1:269–271, 1959.

18 D. P. Dobkin, S. J. Friedman, and K. J. Supowit. Delaunay graphs are almost as good as complete graphs. *Discrete & Computational Geometry*, 5(4):399–407, Dec. 1990.

19 S. Durocher, D. G. Kirkpatrick, and L. Narayanan. On routing with guaranteed delivery in three-dimensional ad hoc wireless networks. *Wireless Networks*, 16(1):227–235, 2010.

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

21 G. Xia and L. Zhang. Toward the tight bound of the stretch factor of Delaunay triangulations. In *CCCG*, 2011.

Appendix

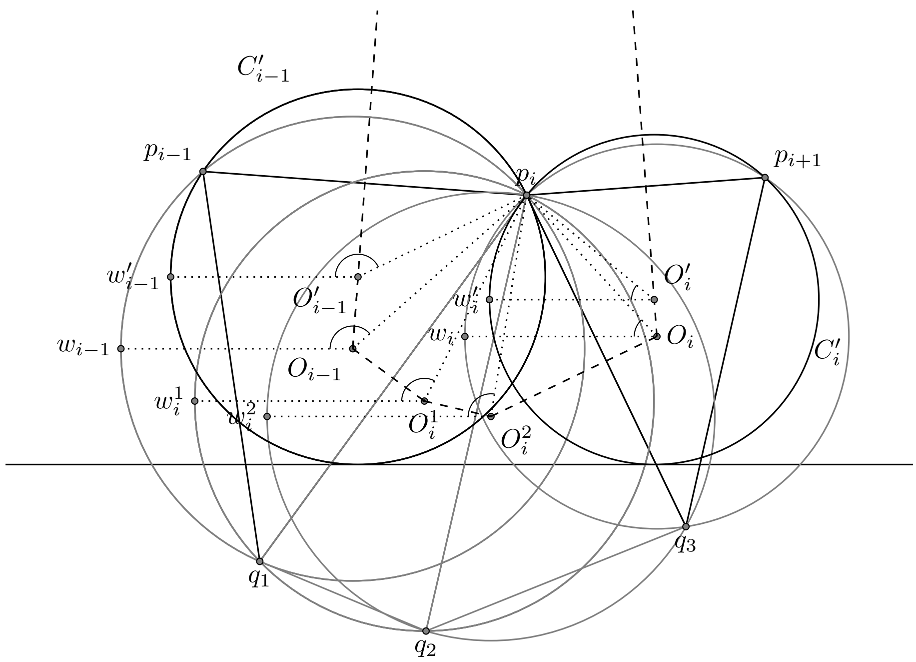

Figure 10 Illustration of Lemmas 11 and 7.

▶ **Lemma 11.** *Let O_i be the center of C_i and let $\angle w_{i-1}O_{i-1}p_i$ and $\angle w_iO_ip_i$ be the angles defined using orientations of $\mathcal{R}{<}p_{i-1}, p_i{>}$ and $\mathcal{R}{<}p_i, p_{i+1}{>}$, respectively. Then (as illustrated in Figure 10), for every $1 \leq i \leq k$,*

$$0 \leq \angle w_iO_ip_i < \angle w_{i-1}O_{i-1}p_i < 3\pi/2.$$

Proof. Without loss of generality, we assume that p_{i-1} and p_i belong to the upper arc from w_{i-1} to r_{i-1} of C_{i-1}. This implies that the orientation of $\mathcal{R}{<}p_{i-1}, p_i{>}$ is clockwise and that p_i lies above st. Because r_{i-1} is the rightmost intersection of C_{i-1} with st, it follows that $\angle w_{i-1}O_{i-1}r_{i-1} < 3\pi/2$, which implies the third inequality. Also, the first inequality holds because the angles are defined to be non-negative.

Consider the triangles which intersect $[st]$ that have p_i as a vertex. Let $C_{i-1} = C_i^0, C_i^1, \ldots, C_i^l = C_i$ be the circles circumscribing these triangles, ordered from left to right. Let O_i^j be the center and w_i^j be the leftmost point of circle C_i^j. The lemma will follow if we show that $\angle w_i^jO_i^jp_i < \angle w_i^{j-1}O_i^{j-1}p_i$ for every $j = 1, \ldots, l$. Let p_i, q_j and q'_j be the vertices of the triangle circumscribed by C_i^{j-1}, in clockwise order. Then q_j must lie below the x-axis and O_i^j must lie on the perpendicular bisector of segment $[p_iq_j]$ and in the half-plane defined by $p_iO_i^{j-1}$ not including w_i^{j-1}. This implies that $\angle p_iO_i^{j-1}w_i^{j-1} > \angle p_iO_i^jw_i^j$. ◀

Proof of Lemma 7 This lemma is illustrated in Figure 10. Consider the circles C_{i-1} and C_i with centers O_{i-1} and O_i, respectively. Without loss of generality, we assume that p_{i-1} and p_i belong to the upper arc from w_{i-1} to r_{i-1} of C_{i-1}. This implies that the orientation of $\mathcal{R}{<}p_{i-1}, p_i{>}$ is clockwise and that $y(p_i) \geq 0$. The center O'_{i-1} of C'_{i-1} lies on the half-line defined by the perpendicular bisector of $[p_{i-1}p_i]$, starting at

O_{i-1} and intersecting $[p_{i-1}p_i]$. This implies that a) $\angle w'_{i-1}O'_{i-1}p_i > \angle w_{i-1}O_{i-1}p_i$ and b) $\angle w'_{i-1}O'_{i-1}p_{i-1} < \angle w_{i-1}O_{i-1}p_{i-1}$. In the context of circle C_i, inequality b) becomes $\angle w'_iO'_ip_i < \angle w_iO_ip_i$ and the second inequality in (5) follows from Lemma 11. Let r be the rightmost intersection of C'_{i-1} with st. It follows that $\angle w'_{i-1}O'_{i-1}r \leq 3\pi/2$. Since p_i lies on $\mathcal{A}_{i-1}{<}w'_{i-1}, r{>}$, the third inequality in (5) holds. Finally, the first inequality in (5) holds since either $p_i = w'_i$ or p_i lies on $\mathcal{A}_i{<}w'_i, p_{i+1}{>}$ that is clockwise oriented. ◀