



HAL
open science

Expected Complexity of Routing in Θ_6 and Half- Θ_6 Graphs

Prosenjit Bose, Jean-Lou de Carufel, Olivier Devillers

► **To cite this version:**

Prosenjit Bose, Jean-Lou de Carufel, Olivier Devillers. Expected Complexity of Routing in Θ_6 and Half- Θ_6 Graphs. EuroCG 2020 - 36th European Workshop on Computational Geometry, Mar 2020, Würzburg, Germany. hal-02479502

HAL Id: hal-02479502

<https://inria.hal.science/hal-02479502v1>

Submitted on 14 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Expected Complexity of Routing in Θ_6 and Half- Θ_6 Graphs*

Prosenjit Bose¹, Jean-Lou De Carufel², and Olivier Devillers³

1 Carleton University, Ottawa, Canada jit@scs.carleton.ca

2 University of Ottawa, Ottawa, Canada jdecaruf@uottawa.ca

3 Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
Olivier.Devillers@inria.fr

Abstract

We study online routing algorithms on the Θ_6 -graph and the half- Θ_6 -graph (which is equivalent to a variant of the Delaunay triangulation). Given a source vertex s and a target vertex t in the Θ_6 -graph (resp. half- Θ_6 -graph), there exists a deterministic online routing algorithm that finds a path from s to t whose length is at most $2\|st\|$ (resp. $2.89\|st\|$) which is optimal in the worst case [Bose *et al.*, SIAM J. on Computing, 44(6)]. We propose alternative, slightly simpler routing algorithms that are optimal in the worst case and for which we provide an analysis of the average routing ratio for the Θ_6 -graph and half- Θ_6 -graph defined on a Poisson point process.

1 Introduction

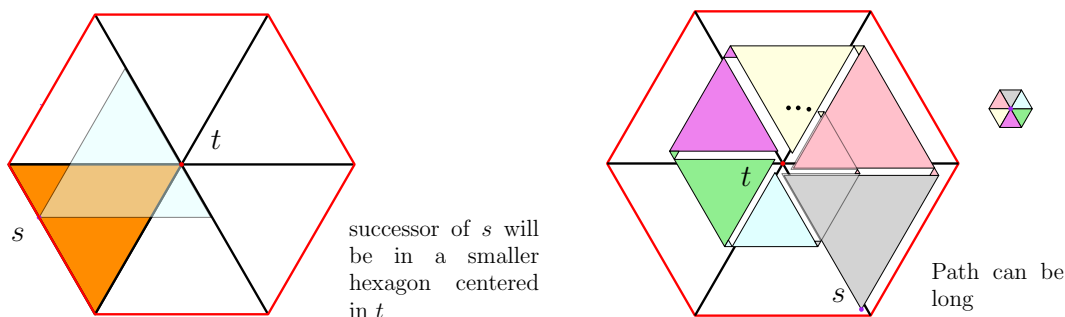
The half- Θ_6 -graph or TD-Delaunay (Triangular-Distance Delaunay [1]) is the Delaunay triangulation for the convex metric whose disk has the shape of an equilateral triangle. The Θ_6 -graph gathers the two half- Θ_6 -graphs corresponding to two symmetric equilateral triangles. Given such a graph, one may be interested in its spanning ratio, that is the worst ratio between the length of a shortest path in the graph and the Euclidean length [6, 12], algorithms to compute paths [11, 10, 9, 8] knowing the whole graph, or routing algorithms that uses only local knowledge of the graph [5].

This paper has two main contributions. The first contribution consists of the design of two new algorithms for routing in the half- Θ_6 -graph in the so called *negative-routing* case. Our new routing algorithms come in two flavors: one is memoryless and the other uses a constant amount of memory. These new negative-routing algorithms have a worst-case optimal routing ratio but are simpler and more amenable to probabilistic analysis than the known optimal routing algorithm [4]. We also provide a new point of view on routing [4] in the half- Θ_6 -graph in the *positive-routing* case.

The second contribution is the analysis of the two new negative-routing algorithms and of the positive-routing algorithm in a random setting, namely when the vertex set of the Θ_6 -graph and half- Θ_6 -graph is a point set that comes from an infinite Poisson point process X of intensity λ . The analysis is asymptotic with λ going to infinity, and gives the expected length of the shortest path between two fixed points s and t at distance one. Our results depend on the position of t with respect to s . We express our results both by taking the worst position for t and by averaging over all possible positions for t .

* This work has been supported by INRIA Associated team TRIP, NSERC and ANR Asparg (ANR-17-CE40-0017). Full version: [3]

1:2 Expected Complexity of Routing in Θ_6 and Half- Θ_6 Graphs



■ **Figure 1** Naive Θ_6 -routing terminates but its stretch is unbounded

2 Routing in Θ_6 -graph and Half- Θ_6 -graph

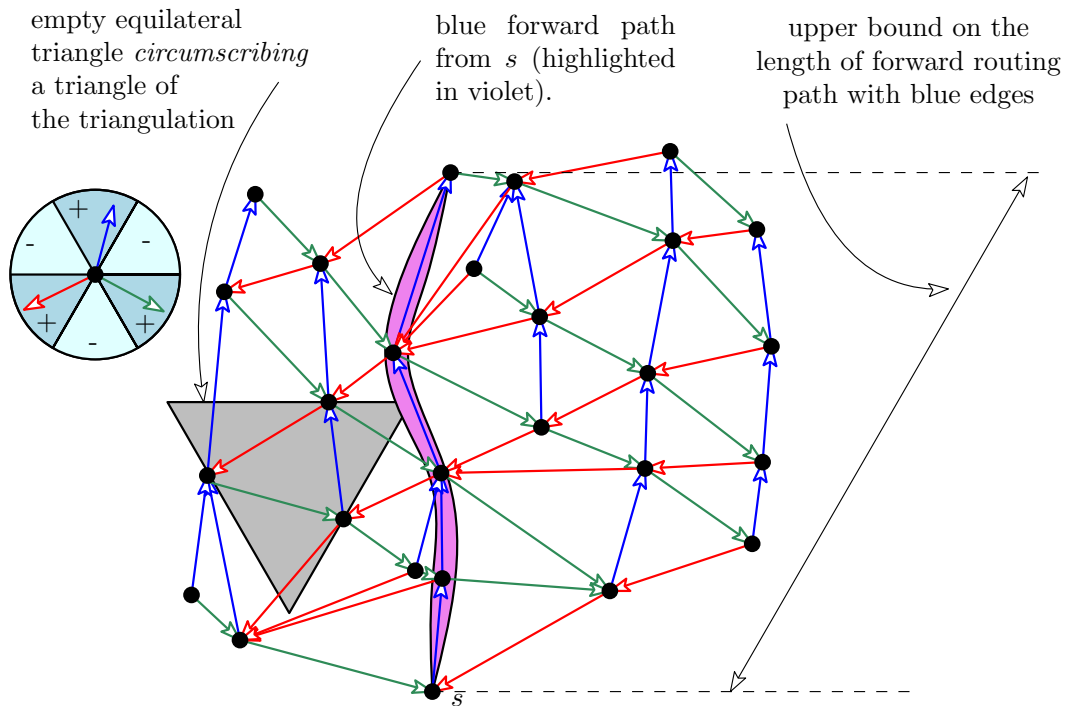
Let a *cone* be the region in the plane between two rays originating from the same point, referred to as the apex of the cone. The Θ_6 -graph is formally defined as follows. For each point p , we split the plane around p into six cones defined by rays emanating from p making an angle of 0 , $\frac{\pi}{3}$, $\frac{2\pi}{3}$, π , $\frac{4\pi}{3}$, and $\frac{5\pi}{3}$ with the horizontal axis. The cone whose bisector is an upward vertical ray emanating from p is labeled C_0^p , and $C_1^p \dots C_5^p$ are labeled in counterclockwise order. Given two vertices p and q with q in C_i^p , define the *canonical triangle* T_{pq} to be the equilateral triangle formed by the intersection of C_i^p and the half-plane that contains p , has q on its boundary, and whose boundary is a line perpendicular to the bisector of C_i^p . We call a canonical triangle T_{pq} *even* if q is in C_i^p for even i and *odd* otherwise. An edge exists in the Θ_6 -graph between two vertices p and q if T_{pq} is empty. The half- Θ_6 -graph uses the same rays to define the cone boundaries except only half the cones are used to define edges, namely only the even cones or only the odd cones. The even half- Θ_6 -graph is defined using the neighbors of p in cones C_0^p , C_2^p , and C_4^p (Fig. 2 illustrates an even half- Θ_6 -graph). The even (resp. odd) cones in the even half- Θ_6 -graph are called positive (resp. negative) and symmetrically for the odd half- Θ_6 -graph.

To route from a vertex s to a vertex t , a simple naive routing algorithm in Θ_6 -graph consists of choosing as successor for s the one in the cone C_i^s containing t and then to iterate. This procedure always terminates but may have an unbounded routing-ratio (Fig. 1).

For the even half- Θ_6 -graph Chew [7] provides a routing algorithm with a routing ratio of 2 when t is in a cone C_i^s with i even. As a consequence, the stretch ratio of the half- Θ_6 -graph is 2 using the routing from t to s when i is odd. Bose *et al.* [4] address the routing problem with i odd and provide an algorithm with routing ratio $\frac{5}{\sqrt{3}} \simeq 2.89$ which is optimal for any constant-memory online routing algorithm [4]. Bonichon and Marckert [2] analyze the naive Θ_6 -routing for Poisson distributed point sets.

3 Two Basic Routing Building Blocks on the Half- Θ_6 -graph

We introduce *forward routing* and *side routing* two routing modes on the half- Θ_6 -graph which serve as building blocks for our routing algorithms that have optimal worst-case behaviour. We consider the even half- Θ_6 -graph and for ease of reference, we color the cones C_0 , C_2 and C_4 blue, red, and green respectively.



■ **Figure 2** A TD -Delaunay triangulation (half- Θ_6 -graph).

3.1 Forward-Routing Phase

Forward-routing consists of only following edges defined by a specific type of cone (i.e., a cone with the same color) until some specified stopping condition is met. For example, suppose the specific cone selected for forward routing is the blue cone. Thus, when forward-routing is invoked at a vertex x , the edge followed is xy where y is the vertex adjacent to x in x 's blue cone. If the stopping condition is not met at y , then the next edge followed is yz where z is the vertex adjacent to y in y 's blue cone. This process continues until a specified stopping condition is met. A path produced by forward routing consists of edges of the same color since edges are selected from one specific cone as illustrated in Fig. 2.

► **Lemma 3.1.** *Suppose that forward-routing is invoked at a vertex s and ends at a vertex t . The length of the path from s to t produced by forward-routing is at most the length of one side of the canonical triangle T_{st} which is $\frac{2}{\sqrt{3}}$ times the length of the orthogonal projection of st onto the bisector of C_0^s .*

Proof. This result follows from the fact that each edge along the path makes a maximum angle of $\frac{\pi}{6}$ with the cone bisector and the path is monotone in the direction of this bisector. ◀

3.2 Side-Routing Phase in the Half- Θ_6 -graph

The *side-routing phase* is defined on the half- Θ_6 -graph by using the fact that it is the TD -Delaunay triangulation, and thus planar. Consider a line ℓ parallel to one of the cone sides. W.l.o.g., we will assume ℓ is horizontal. We call the side of the line that bounds even cones the *positive side* of ℓ . (For a horizontal line, the positive side is below ℓ , for the lines with slopes $-\sqrt{3}$ and $\sqrt{3}$, respectively, the positive side is above the line.) Let s and t be two

1:4 Expected Complexity of Routing in Θ_6 and Half- Θ_6 Graphs

vertices below ℓ and $\Delta_1, \Delta_2, \dots, \Delta_j$ be an ordered sequence of consecutive triangles of the TD -Delaunay triangulation intersecting ℓ such that s is the bottom-left vertex of Δ_1 , and t is the bottom-right vertex of Δ_j . Note that B is a path in the half- Θ_6 -graph. Side-routing invoked at vertex s along ℓ stopping at t consists of walking from s to t along B (Fig. 3 for an example).

► **Lemma 3.2.** *Side-routing on the positive side of a line ℓ parallel to a cone boundary invoked at a vertex s and stopped at a vertex t in the half- Θ_6 -graph results in a path whose length is bounded by twice the length of the orthogonal projection of st on ℓ . This path only uses edges of two colors and all vertices of the path have their successor of the third color on the other side of ℓ .*

Proof. W.l.o.g., assume ℓ is horizontal and the positive side is below ℓ . Consider the triangles Δ_i , $1 \leq i \leq j$ as defined above. The empty equilateral triangle ∇_i circumscribing Δ_i has a vertex of Δ_i on each of its side by construction (∇_i are shown in grey in Fig. 3). If Δ_i has an edge of the path (i.e., below ℓ) then the vertex on the horizontal side of ∇_i is above the line while the two others are below. Thus, such an edge of the path goes from the left to the right side of ∇_i . Based on the slopes of the edges of ∇_i , we have the following:

- a- Each edge on the path has a length smaller than twice its horizontal projection. Therefore, summing the lengths of all the projections of the edges gives the claimed bound on the length.
- b- If the slope is negative (resp. positive), the path edge is green (resp. red).
- c- The blue successor of a vertex u on the lower sides of ∇_i is above ℓ since the part of C_o^u below ℓ is inside ∇_i and thus contains no other points. Blue edges are not on the path. ◀

3.3 Positive routing in the Half- Θ_6 -graph (and the Θ_6 -graph)

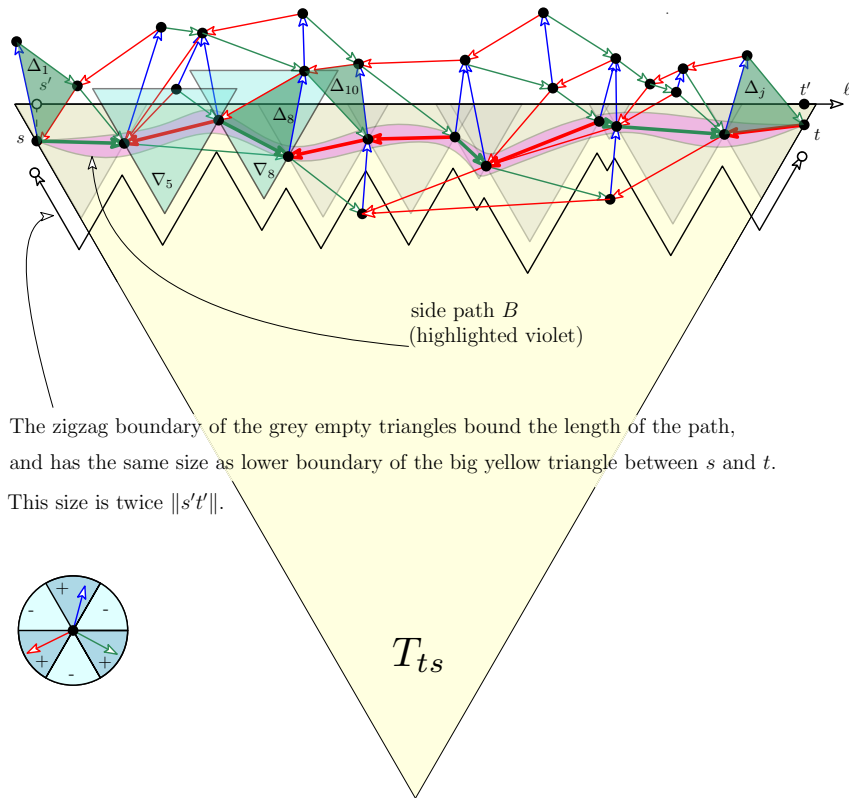
If t is in a positive cone of s , Bose et al. [4] (similar to Chew's algorithm) proposed a routing algorithm in the half- Θ_6 -graph which they called *positive routing*. This algorithm can be rephrased in two phases: a *forward-routing* phase and a *side-routing* phase. The forward-routing phase is invoked with source s and destination t . It produces a path from s to the first vertex u outside the negative cone of t that contains s . The side-routing phase is invoked with source u and destination t and finds a path along the boundary of this negative cone. (Fig. 4-left). The stretch of such a path is proven to be smaller than 2 [4].

4 Alternative Negative Routing Algorithms in the Half- Θ_6 -graph

In this section, we outline two alternatives to the negative routing algorithm described by Bose et al. [4]. Our algorithms are a little simpler to describe, have the same worst-case routing ratio, and are easier to analyze in the random setting. The lower bound of $\frac{5}{\sqrt{3}} \simeq 2.89$ [4] applies to our alternative negative routing algorithms.

4.1 Memoryless Routing

- Case 1.** If t is in the positive cone C_i^s , take one step of forward-routing towards t
- Case 2.** If t is in the negative cone C_i^s and the successor u of s in C_{i-1}^s is outside T_{ts} (red triangle empty in Fig. 4-right), take one step of side-routing along the side of T_{ts} crossed by su .
- Case 3.** If t is in the negative cone C_i^s and the successor u of s in C_{i+1}^s is outside T_{ts} (green triangle empty in Fig. 4-right), take one step of side-routing along the side of T_{ts} crossed by su .



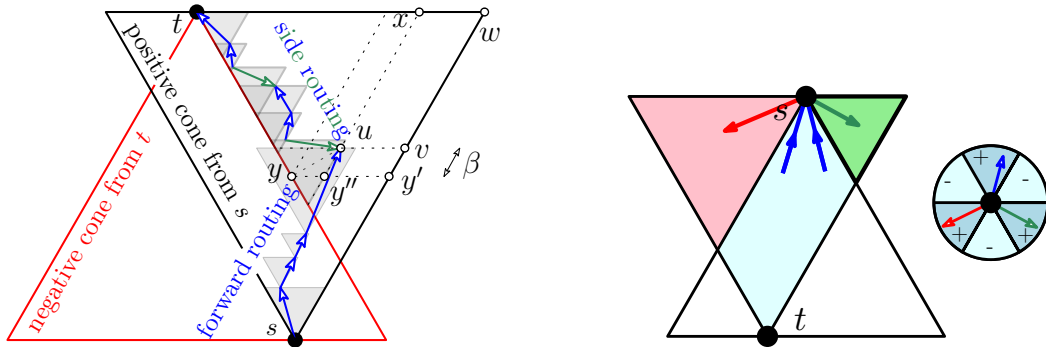
■ **Figure 3** A side path below the horizontal line ℓ .

Case 4. If t is in the negative cone C_i^s and both successors of s in C_{i-1}^s and C_{i+1}^s are inside T_{ts} (green and red triangle non empty in Fig. 4-right), take one step of forward-routing in the direction of the side of T_{ts} incident to t closest to s (go to the green successor of s in Fig. 4-right).

Beyond the presentation, our strategy differs from the one of Bose et al. in Case 4 where Bose et al. follows a blue edge if one exists. We remark that, when we reach Case 3, we enter a side-routing phase that will continue until t is reached since a side-routing step ensures that at the next iteration side-routing will also be applicable. The same argument holds in Case 2, unless we reach a point s with both successors outside T_{ts} in which case we follow the other side of T_{ts} . To summarize, if t is in a positive cone of s , this routing algorithm will produce the path described at Section 3.3. If t is in a negative cone of s we use a forward phase in the green triangle, until we reach a vertex u whose edge in the green triangle intersects T_{ts} (recall that we assume that the green triangle is the smaller one). At this point, we invoke side-routing from u to t along the boundary of T_{ts} .

► **Lemma 4.1.** *Memoryless negative routing has a worst-case routing ratio of $\frac{5}{\sqrt{3}} \simeq 2.89$.*

Proof. Assume w.l.o.g. $s \in C_0^t$. Referring to Fig. 5-left, let w be the upper right vertex of T_{ts} , v be the orthogonal projection of u on tw and x its projection parallel to tw on sw . By Lemma 3.1, the path from s to u has length bounded by $\|sx\|$ and by Lemma 3.2, the path from u to t has length bounded by $2\|vt\|$. Combining the two paths, the length is bounded by $\|sx\| + 2\|vt\| \leq \|sw\| + 2\|wt\|$. Thus the stretch is smaller than $\frac{\|sw\| + 2\|wt\|}{\|st\|}$.



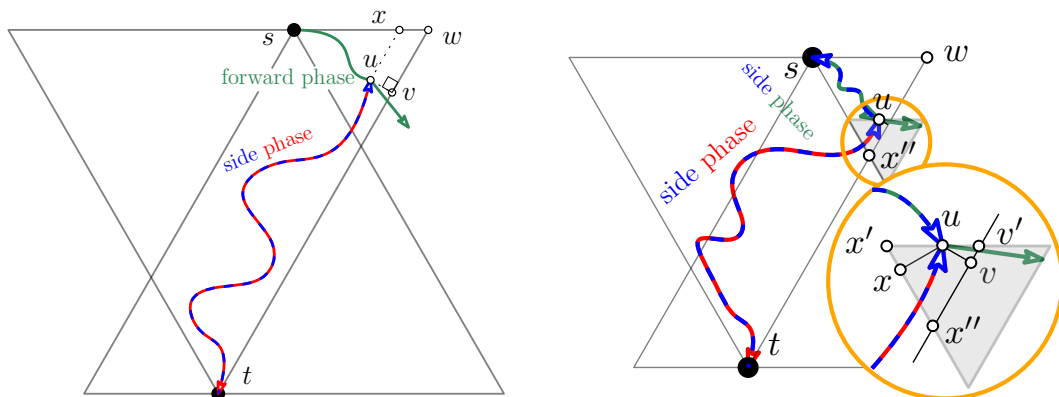
■ **Figure 4** Positive and negative routing schemes [4].

Defining $\xi = \frac{\|wt\|}{\|sw\|}$ the stretch can be expressed as a function $\xi \rightsquigarrow \frac{2+\xi}{\sqrt{\frac{3}{4}+(\xi-\frac{1}{2})^2}}$. It attains its maximum value of $\frac{5}{\sqrt{3}}$ when $\xi = \frac{1}{2}$ corresponding to s and t lie on a vertical line. ◀

4.2 Constant-Memory Negative Routing

We propose a second negative routing algorithm that has the same worst-case routing ratio, but we will prove that it has a better average routing ratio. However, it is no longer memoryless since it needs to remember the coordinates of one vertex, namely the source of the path. Let x'' be the intersection between T_{ts} and T_{st} closest to s . (Fig. 5-right). The idea is to use side-routing from s along sx'' and, just before exiting the green triangle, apply side-routing along $x''t$. This routing algorithm is identical to the one in the previous subsection, except that we replace Case 4 with the following, where u is the current vertex and s is the origin of the path whose coordinates are kept in memory:

Case 4' If t is in the negative cone C_i^u and both successors of u in C_{i-1}^u and C_{i+1}^u are inside T_{tu} (green and red triangle non empty): take one step of side-routing along the line sx'' .



■ **Figure 5** For Lemmas 4.1 and 4.2

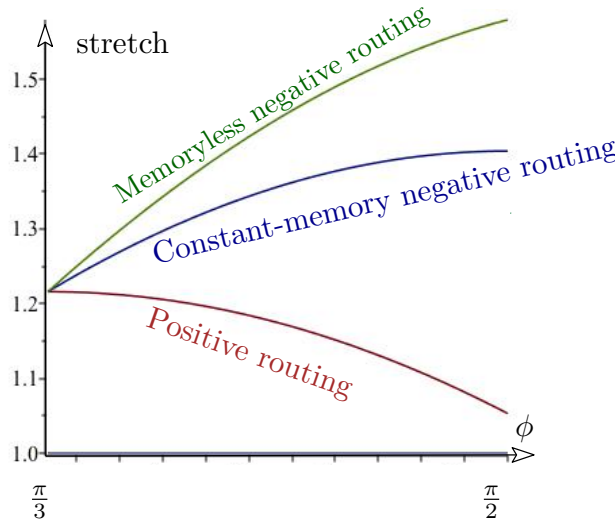
► **Lemma 4.2.** *Constant-memory negative routing has a worst-case routing ratio of $\frac{5}{\sqrt{3}} \simeq 2.89$.*

Proof. Assume w.l.o.g. $s \in C_0^t$. Referring to the Fig. 5-right, let x' and x be the horizontal and orthogonal projections of u on T_{st} , respectively, and v' and v be the horizontal and orthogonal projections of u on T_{ts} , respectively. By Lemma 3.2, the path from s to u has length bounded by $2\|sx\|$ and by Lemma 3.2 again, the path from u to t has length bounded by $2\|vt\|$. Combining the two paths the length is bounded by $2\|sx\| + 2\|vt\| \leq 2\|sx'\| + 2\|x'x\| + 2\|v't\| = 2\|wt\| + 2\|xx'\|$. Since x is the orthogonal projection of u on the side $x'x''$ of the equilateral triangle $x'x''v'$, $\|xx'\|$ is smaller than the half side of the triangle $x'x''v'$ and we get a bound on the length of $2\|wt\| + 2\|xx'\| \leq 2\|wt\| + 2\frac{1}{2}\|x'v'\| \leq 2\|wt\| + \|sw\|$. ◀

5 Probabilistic Analysis

► **Theorem 5.1.** *Let X be a Poisson point process of intensity λ , s and t two points at unit distance and ϕ the angle of st with the horizontal axis. The limits of the expected routing ratios of the different routing algorithms on the half- Θ_6 -graph defined on $X \cup \{s, t\}$, as λ tends to ∞ are given in the following table and graph (with $\tau_1 := \frac{1}{4\sqrt{3}}(3 \ln 3 + 4)$):*

Routing	$\mathbb{E}[\text{routing ratio}](\phi)$	$\max_{s,t} \mathbb{E}[\text{routing ratio}]$	$\mathbb{E}_{s,t}[\mathbb{E}[\text{routing ratio}]]$
Positive routing	$\tau_1 \left(\sin \phi + \frac{1}{\sqrt{3}} \cos \phi \right)$	$\frac{2}{\sqrt{3}}\tau_1 \simeq 1.2160$	$\frac{2\sqrt{3}}{\pi}\tau_1 \simeq 1.1612$
Constant-memory	$\frac{4}{3}\tau_1 \sin \phi$	$\frac{4}{3}\tau_1 \simeq 1.4041$	$\frac{4}{\pi}\tau_1 \simeq 1.3408$
Memoryless	$\tau_1 \left(\frac{3}{2} \sin \phi - \frac{\sqrt{3}}{6} \cos \phi \right)$	$\frac{3}{2}\tau_1 \simeq 1.5800$	$\frac{6-\sqrt{3}}{\pi}\tau_1 \simeq 1.4306$



Proof. See full paper [3] ◀

References

- 1 Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and David Ilcinkas. Connections between Theta-graphs, Delaunay triangulations, and orthogonal surfaces. In *Proceedings of the 36th International Conference on Graph Theoretic Concepts in Computer Science (WG 2010)*, pages 266–278, 2010.
- 2 Nicolas Bonichon and Jean-François Marckert. Asymptotics of geometrical navigation on a random set of points in the plane. *Advances in Applied Probability*, 43(4):899–942, 2011. doi:10.1239/aap/1324045692.
- 3 Prosenjit Bose, Jean-Lou De Carufel, and Olivier Devillers. Expected Complexity of Routing in Θ_6 and Half- Θ_6 Graphs. Research report, INRIA, 2019. URL: <https://hal.inria.fr/hal-02338733>.
- 4 Prosenjit Bose, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. Optimal local routing on Delaunay triangulations defined by empty equilateral triangles. *SIAM Journal on Computing*, 44(6):1626–1649, 2015. doi:10.1137/140988103.
- 5 Prosenjit Bose and Pat Morin. Online routing in triangulations. *SIAM Journal on Computing*, 33(4):937–951, 2004.
- 6 Prosenjit Bose and Michiel Smid. On plane geometric spanners: A survey and open problems. *Computational Geometry: Theory and Applications*, 46(7):818–830, 2013.
- 7 Paul Chew. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2):205–219, 1989.
- 8 E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- 9 J. E. Hopcroft and R. E. Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.
- 10 C. Y. Lee. An algorithm for path connection and its applications. *IRE Transaction on Electronic Computers*, EC-10(3):346–365, 1961.
- 11 E. F. Moore. The shortest path through a maze. In *Proceedings of the International Symposium on the Theory of Switching*, pages 285–292, 1959.
- 12 G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.