

Routing in Geometric Graphs

John Stuart

Thesis submitted to the University of Ottawa
in partial Fulfilment of the requirements for the
Degree of Doctorate in Philosophy Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Under the Supervision of Jean-Lou De Carufel (University of Ottawa)
and Prosenjit Bose (Carleton University)

© John Stuart, Ottawa, Canada, 2026

Contents

0	Acknowledgements	iv
1	Abstract	v
2	Introduction	1
2.1	Preliminaries	2
3	Exact Spanning and Routing Ratios for Arbitrary Triangle-Delaunay Graphs	4
3.1	Abstract	4
3.2	Introduction	4
3.3	Preliminaries	5
3.3.1	Our Contributions	7
3.4	Spanning Ratio	7
3.5	Local Routing	8
3.5.1	Local Routing Algorithm	10
3.5.2	Comparison to known routing algorithms	14
4	Improved Routing for Polygon-Delaunay Graphs	16
4.1	Abstract	16
4.2	Introduction	16
4.3	Preliminaries	17
4.4	Routing Ratio Upper Bound	19
4.4.1	Local Routing Algorithm	19
4.4.2	Extending Algorithm 2 to graphs where the convex hull is not present	29
4.4.3	Extending our result to hexagons, septagons and octagons	30
4.5	Routing Ratio Lower Bound for $\bigcirc_5$ -Delaunay graphs	31

4.6	Spanning Ratio Lower Bound for $\bigcirc_5$ -Delaunay graphs	32
4.7	Conclusions	32
5	Routing in Sparse Directed Planar Graphs: The $\vec{\text{Yao}}_4^\infty$-Graph	34
5.1	Abstract	34
5.2	Introduction	35
5.3	Preliminaries	37
5.4	A Local Routing Algorithm for $\vec{\text{Yao}}_4^\infty$ -Graphs	38
5.4.1	Analysis of DIRTY Algorithm 3	40
5.5	Lower Bounds	52
6	The Spanning Ratio of the Directed $\vec{\Theta}_6$-Graph is 5	60
6.1	Abstract	60
6.2	Introduction	60
6.3	Preliminaries	63
6.4	Lower Bound	64
6.5	Upper Bound	66
6.5.1	Main Argument	66
6.5.2	Remaining Cases for Upper Bound Proof	81
6.6	Lower Bound on the Routing Ratio of the $\vec{\Theta}_6$ -Graph	87
6.7	Conclusion	88

Chapter 0

Acknowledgements

I would like to thank the following people for their support throughout this journey.

My supervisors, Jean-Lou De Carufel and Jit Bose, for introducing me to research in Computational Geometry and for their guidance, encouragement, insightful feedback, and for making research both challenging and rewarding.

The staff of the School of Electrical Engineering and Computer Science at the University of Ottawa for their support during my studies.

My partner, Sandrine, for introducing me to computational geometry and encouraging me to pursue my studies.

Chapter 1

Abstract

Note to the reader. *This thesis is presented in an integrated (article-based) format. The core chapters consist of research based on articles published in conferences, reproduced with supplemental proofs for completeness. As a result, some introductory material, background, definitions, and notation are intentionally repeated across chapters. This repetition has been retained to preserve the integrity and readability of each article as a self-contained contribution.*

Chapter 3: Exact Spanning and Routing Ratios for Arbitrary Triangle-Delaunay Graphs

A Delaunay graph built on a planar point set has an edge between two vertices when there exists a disk with the two vertices on its boundary and no vertices in its interior. When the disk is replaced with an equilateral triangle, the resulting graph is known as a *Triangle-Distance Delaunay Graph* or TD-Delaunay for short. A generalized TD_{θ_1, θ_2} -Delaunay graph is a TD-Delaunay graph whose empty region is a scaled translate of a triangle with angles of $\theta_1, \theta_2, \theta_3 := \pi - \theta_1 - \theta_2$ with $\theta_1 \leq \theta_2 \leq \theta_3$. We prove that $\frac{1}{\sin(\theta_1/2)}$ is a lower bound on the spanning ratio of these graphs which matches the best known upper bound [36]. Then we provide a memoryless local routing algorithm for TD_{θ_1, θ_2} -Delaunay graphs with a routing ratio that is optimal in the worst case. When $\theta_1 = \theta_2 = \frac{\pi}{3}$, our expressions for the spanning ratio and routing ratio evaluate to 2 and $\frac{\sqrt{5}}{3}$, matching the known tight bounds for TD-Delaunay graphs. This chapter of the thesis is based on the joint work with Prosenjit Bose and Jean-Lou De Carufel, which was published at *The 36th Canadian Conference on Computational Geometry (CCCG 2024)* [42]. This research was subsequently submitted to the *Journal of Graph Algorithms and Applications*.

Chapter 4: Improved Routing for Polygon-Delaunay Graphs

The standard Delaunay triangulation is a geometric graph whose vertices are points in the plane, and two vertices share an edge if they lie on the boundary of an empty disk. If the disk is replaced with a homothet of a fixed convex shape C , then the resulting graph is called a C -Delaunay graph. We study the problem of local routing in C -Delaunay graphs where C is a regular polygon having five to eight sides. In particular, we generalize the routing algorithm of Chew for square-Delaunay graphs [26] in order to obtain the following approximate upper bounds of 4.640, 6.429, 8.531 and 4.054 on the spanning and routing ratios for pentagon-,

hexagon-, septagon-, and octagon-Delaunay graphs, respectively. The exact expression for the upper bounds of the routing ratio is

$$\Psi(n) := \begin{cases} \sqrt{1 + ((\cos(2\pi/n) + n - 1)/\sin(2\pi/n))^2} & \text{if } n \in \{5, 6, 7\}, \\ \sqrt{1 + ((\cos(\pi/8)\cos(3\pi/8) + 3)/(\cos(\pi/8)\sin(3\pi/8)))^2} & \text{if } n = 8. \end{cases}$$

We show that these bounds are tight for the output of our routing algorithm by providing a point set where these bounds are achieved. We also include lower bounds of 1.708 and 1.995 on the spanning and routing ratios of the pentagon-Delaunay graph.

Our upper bounds yield a significant improvement over the previous routing ratio upper bounds for this problem, which previously sat at around 400 for the pentagon, septagon, and octagon as well as 18 for the hexagon. Our routing ratios also provide significant improvements over the previously best known spanning ratios for pentagon-, septagon- and octagon-Delaunay graphs, which were around 45.

Although we also provide lower bounds, they do not match the upper bounds we obtained, hence it is open whether these gaps can be closed.

This chapter of the thesis is based on the joint work with Prosenjit Bose and Jean-Lou De Carufel. This work appeared at *The 35th International Symposium on Algorithms and Computation (ISAAC 2024)* [16]. It was also submitted to the journal, *Computational Geometry: Theory and Applications*.

Chapter 5: Routing in Sparse Directed Planar Graphs: $\vec{\text{Yao}}_4^\infty$

In this chapter, we explore local routing in sparse graphs. In particular, we aim to solve the following problem:

Problem. What is the least constant $c > 0$ such that for any point set V , we can construct a planar directed spanner G with less than $c|V|$ directed edges that supports local routing with constant routing ratio?

In 1986, Chew introduced a local routing algorithm for the square-Delaunay graph, discussed in Chapter 4 [26]. Being a triangulation, there are roughly $3|P|$ undirected edges in the worst case. Since Chew's local routing algorithm uses each edge in both directions, this construction shows that $6|V|$ directed edges suffice to support local routing. Among all local routing algorithms in the literature for planar graphs, roughly $6|V|$ directed edges are required. We lower this constant and route in sparser graphs with only $4|V|$ directed edges.

The $\vec{\text{Yao}}_4^\infty$ and $\vec{\text{Yao}}_4$ -graphs are two families of directed geometric graphs whose vertices are points in the plane, and each vertex has up to four outgoing edges. Consider a horizontal and a vertical line through each vertex v , defining four quadrants around v . Then v has an outgoing edge to the *closest* vertex in each of its four quadrants. When the distance is measured using the Euclidean norm, the resulting graph is the $\vec{\text{Yao}}_4$ -graph, whereas with the L_∞ -norm, we obtain the $\vec{\text{Yao}}_4^\infty$ -graph, which is a sub-graph of the well-known square-Delaunay graph.

In this chapter, we provide a local routing algorithm with routing ratio at most 85.22 for $\vec{\text{Yao}}_4^\infty$ -graphs. Prior to this work, no constant spanning or local routing ratios for $\vec{\text{Yao}}_4^\infty$ -graphs were known. $\vec{\text{Yao}}_4^\infty$ -graphs are the sparsest family of directed planar graphs supporting a local routing strategy with constant routing ratio. Furthermore, we show that no local routing algorithm for $\vec{\text{Yao}}_4^\infty$ -graphs can have a routing ratio lower than $7 + \sqrt{2} \approx 8.41$. Moreover, we prove that the spanning ratio is at least $5 + \sqrt{2} \approx 6.41$ for the class of $\vec{\text{Yao}}_4^\infty$ -graphs. The techniques we develop in this chapter also allow us to prove lower bounds of $7 - \sqrt{3} + \sqrt{5 - 2\sqrt{3}} \approx 6.51$ and $7 + \sqrt{2}$ for the spanning and routing ratios of $\vec{\text{Yao}}_4$, respectively.

While we give a loose upper bound of 85.22 on the routing ratio in $\vec{\text{Yao}}_4^\infty$ -graphs, we believe that a stronger upper bound is likely possible to prove via an alternative routing algorithm. On the other hand, the routing ratio lower bound that we provide is among the largest found in the literature for geometric graphs. This further highlights the difficulty of routing in $\vec{\text{Yao}}_4^\infty$ -graphs.

The spanning ratio of $\vec{\text{Yao}}_4^\infty$ -graphs was also unknown prior to this work, and while our routing ratio upper bound yields a spanning ratio upper bound, there is much improvement to be made. Improving the upper bound on the spanning ratio of $\vec{\text{Yao}}_4^\infty$ -graphs is another open problem that we propose.

This chapter is based on joint work with Prosenjit Bose and Jean-Lou De Carufel. This work was published at *The 19th Algorithms and Data Structures Symposium (WADS 2025)* [17]. It was also submitted to the *Journal of Computational Geometry*.

Chapter 6: The Spanning Ratio of the Directed Θ_6 -Graph is 5

The directed $\vec{\Theta}_6$ -graph uses a similar construction to the Yao graph. For two points $u, v \in \mathbb{R}^2$, we define ∇_u^v to be the equilateral triangle with side slopes $0, \sqrt{3}, -\sqrt{3}$ and u on a corner and v on the opposing side. Then for a point set $P \subset \mathbb{R}^2$, the directed $\vec{\Theta}_6(P)$ -graph has an edge from u to v exactly when the interior of ∇_u^v contains no vertices of P . This definition highlights the similarities with the closely-related Δ -Delaunay graph. In particular, the undirected Θ_6 -graph is the union of two rotated copies of the Δ -Delaunay graph. By inheritance, the undirected Θ_6 -graph also has spanning ratio at most 2, which is tight [27]. The directed version however does not preserve distances nearly as well. It was shown that the class of directed $\vec{\Theta}_6$ -graphs has a spanning ratio of at least 4, and at most 7 [1]. The authors conjectured that the true spanning ratio is closer to 4. We close this gap by showing that the spanning ratio is exactly 5. This result is among a short list of tight bounds on spanning ratios of geometric graphs, including the undirected θ_{4k+2} -graph, as well as the generalized Delaunay graphs based on triangles, parallelograms, and hexagons, shown in Table 1.1. The techniques we employ are novel to the area as well. In particular, we use linear programming to prove the existence of a short path in the directed $\vec{\Theta}_6$ -graph.

In addition to achieving an exact spanning ratio of 5 for $\vec{\Theta}_6$ -graphs, we also rectify a mistake in the lower bound proof of [1] on the routing ratio in $\vec{\Theta}_6$ -graphs. The lower bound of 6 on the routing ratio still holds with our proposed argument.

Reference	Graph	Spanning Ratio	Notes
[14]	Θ_{4k+2} ($k \geq 1$)	$1 + 2 \sin(\theta/2)$	Also for constrained version[24]
[27]	$\triangle$ -Delaunay	2	Also for affine transformations [36, 42]
[9]	$\square$ -Delaunay	$\sqrt{4 + 2\sqrt{2}} \approx 2.613$	Also for affine transformations [15, 43]
[40]	$\hexagon$ -Delaunay	2	

Table 1.1: Other geometric graphs with known exact spanning ratios. Notice these graphs have bidirectional edges.

Chapter 6 is based on research with Darryl Hill and my supervisors, Prosenjit Bose and Jean-Lou De Carufel. Our article was published at *The 41st International Symposium on Computational Geometry (SoCG 2026)* [18], and selected for a special issue of the Springer journal, *Discrete & Computational Geometry*.

Chapter 2

Introduction

Routing is a central problem in computational geometry, with a wide range of applications. Navigation assistance, flight planning, and internet traffic all require different forms of routing. We focus on the naturally discrete version of routing which typically comes up in problems such as navigation planning in a network, as opposed to the continuous version which arises for areas such as autonomous driving. To formalize this flavour of routing, we first introduce the notion of a *geometric graph*. The vertex set of a geometric graph is a finite set of points in the plane and each edge is a line segment joining the endpoints, weighted by the Euclidean norm. Given a geometric graph G , the *spanning ratio* of a subgraph H is a measure of how well distances of G are preserved. In particular, a subgraph H is called a c -spanner of G if for every edge $uv \in G$, the shortest path in H from u to v is at most c times the length of the edge uv in G [39]. The smallest such c is referred to as the spanning ratio of H . Throughout this thesis, we consider G to be the complete graph with edges weighted by Euclidean distance. In computational geometry, researchers study families of graphs and strive to determine whether there exists a constant upper bound on the spanning ratio of all graphs in the family. When an upper bound is successfully proven for a spanning ratio, researchers often turn to efficiently finding short paths using only local information.

A small spanning ratio guarantees the existence of reasonably short paths in a graph, however these paths are not necessarily easy to find. For this reason, we also study path-finding algorithms in various graphs. Loosely speaking, we say that such an algorithm has *routing ratio* c if the paths produced have length at most c times the straight-line distance from source to destination. We primarily focus on the constrained model of local routing, where the path-finding algorithm does not have access to the entire graph, but rather the neighbourhood of the current vertex. Our algorithms typically use memory to store information from one step to the next, however in some cases we are able to provide memory-less variants.

A well-studied graph in computational geometry is the Delaunay triangulation, and for good reason. It appears to have many desirable properties, such as small spanning ratio, while still being planar. A Delaunay triangulation can be constructed on any finite set of points in the plane by including an edge between two points exactly when there exists a disk with those two points on its boundary and no vertices in its interior [30]. To ensure planarity, we assume that no four vertices are co-circular. Bounding the exact spanning ratio of the Delaunay triangulation is a long-standing open problem, first posed in 1986 by Chew [26]. Currently, it is known to be

between 1.5932 [45] and 1.998 [44]. In other words, there exists a point set where the spanning ratio is at least 1.5932, and for any point set, the spanning ratio is at most 1.998. While the exact spanning ratio of the standard Delaunay triangulation remains unknown, several variants do have known exact spanning ratios. For example, when the empty disk is replaced with an oriented square, we obtain the L_∞ or L_1 -Delaunay graph, which is known to have a spanning ratio of exactly $\sqrt{4 + 2\sqrt{2}} \approx 2.61$ [9]. If the square is replaced with a rectangle or parallelogram, then the exact bound is also known [15, 43]. Similarly, substituting the empty disk with an oriented triangle or hexagon yields another Delaunay-type graph whose worst-case spanning ratio is exactly 2 [27, 40]. In general, one can define variants of the Delaunay triangulation by replacing the *empty disk* with any convex shape C , and the resulting generalized Delaunay graph has a constant spanning ratio depending only on the properties of the convex shape, such as ratio of the perimeter to the width [11]. Generalized Delaunay graphs are planar, assuming that no four points lie on the boundary of any scaled translate of C . The dual graph of a Delaunay triangulation is well-studied and referred to as a Voronoi diagram. Alternatively, one can define a Voronoi diagram as follows: Given a set of points S in the plane, a Voronoi diagram is the planar graph with $|S|$ faces such that for each $s \in S$, there is a face containing all points of $\mathbb{R}^2$ that are closer to s than to any other point of S . The definition of a Voronoi diagram implicitly uses the Euclidean norm, however, the author of [37] generalizes Voronoi diagrams to use an arbitrary norm. Then, taking the dual of a generalized Voronoi diagram yields a generalized Delaunay graph where the empty region C is the unit disk in the specified norm. Note that any norm in the plane corresponds to a convex shape with a specified centre, interpreted as the norm's unit disk. Interestingly, the Voronoi diagram depends on the choice of centre of the convex shape, whereas the Delaunay graph is invariant under translation of the convex shape's centre [37].

These results highlight the research community's strong understanding of the spanning properties of generalized Delaunay graphs. However, while Delaunay graphs have excellent spanning properties, they can suffer from high vertex degree. In particular, it is possible to have vertices of degree linear in the size of the point set. In the context of routing, storing the coordinates of all neighbours of a high degree vertex can use large amounts of memory. For this reason, we also study the spanning properties of graphs with constant vertex degree, such as Yao and Theta graphs.

2.1 Preliminaries

For any point $u \in \mathbb{R}^2$, we let $x(u)$ and $y(u)$ denote the x - and y -coordinates of u , respectively. We denote the line segment between points u, v as uv . A geometric graph is a graph whose vertex set is a set of points in the plane and whose edge set is a set of line segments joining pairs of vertices. Edges are weighted by the L_2 (Euclidean) lengths of the corresponding line segments. In a geometric graph, each vertex is identified with its coordinates. For two vertices u, v in a geometric graph G , the length of the shortest path, (i.e., the sum of the lengths of the edges of the shortest path) from u to v in G is denoted by $d_G(u, v)$. Then for a real number $c \geq 1$, G is said to be a c -spanner if for all points u, v in G , we have $d_G(u, v) \leq c\|uv\|_2$. The spanning ratio of a class of graphs $\mathcal{G}$ is the infimum of $\{c \in \mathbb{R} \mid \forall G \in \mathcal{G}, G \text{ is a } c\text{-spanner}\}$.

When routing in a geometric graph, each vertex is identified with its coordinates. Here, one unit of memory is $\log_2(|V(G)|)$ bits in order to store points on a polynomially-sized grid. Formally, an m -memory local routing algorithm is a function that takes as input $(s, N(s), t, M)$, and outputs some memory register M' and a vertex $p \in N(s)$ where s is the current vertex, $N(s)$ is the outgoing neighbourhood of s , t is the destination, and both M, M' are memory registers that can hold m numbers. If $m = 0$, then the local routing algorithm is referred to as *memoryless*. The routing ratio of an algorithm for a family of geometric graphs $\mathcal{G}$ is the infimum over all c such that the path output by the algorithm for any pair of vertices $s, t \in V(G)$ for $G \in \mathcal{G}$ has length at most $c\|st\|_2$. Note that the routing ratio is an upper bound on the spanning ratio.

Chapter 3

Exact Spanning and Routing Ratios for Arbitrary Triangle-Delaunay Graphs

Prosenjit Bose, Jean-Lou De Carufel, John Stuart

3.1 Abstract

A Delaunay graph built on a planar point set has an edge between two vertices when there exists a disk with the two vertices on its boundary and no vertices in its interior. When the disk is replaced with an equilateral triangle, the resulting graph is known as a *Triangle-Distance Delaunay Graph* or TD-Delaunay for short. A generalized TD_{θ_1, θ_2} -Delaunay graph is a TD-Delaunay graph whose empty region is a scaled translate of a triangle with angles of $\theta_1, \theta_2, \theta_3 := \pi - \theta_1 - \theta_2$ with $\theta_1 \leq \theta_2 \leq \theta_3$. We prove that $\frac{1}{\sin(\theta_1/2)}$ is a lower bound on the spanning ratio of these graphs which matches the best known upper bound (Lubiw & Mondal *J. Graph Algorithms Appl.*, **23**(2):345–369). Then we provide an online local routing algorithm for TD_{θ_1, θ_2} -Delaunay graphs with a routing ratio that is optimal in the worst case. When $\theta_1 = \theta_2 = \frac{\pi}{3}$, our expressions for the spanning ratio and routing ratio evaluate to 2 and $\frac{\sqrt{5}}{3}$, matching the known tight bounds for TD-Delaunay graphs.

3.2 Introduction

Geometric graphs are graphs whose vertex sets are points in the plane and whose edge weights are the corresponding Euclidean distances. A common theme in Computational Geometry is the study of shortest paths. In geometric graphs, one measure of how well a graph preserves distances is its spanning ratio. The spanning ratio of a geometric graph is the smallest upper bound on the ratio of distance in the graph to distance in the plane for all pairs of points [39]. One particular geometric graph of interest is the Delaunay triangulation, which has an edge between two vertices exactly when they lie on the boundary of a disk which contains no other vertex in its interior [30].

A long-standing open problem is to determine the worst-case spanning ratio of the Delaunay triangulation, which is known to be between 1.5932 [45] and 1.998 [44]. In other words, there exists a point set where the spanning ratio is at least 1.5932, and for any point set, the spanning

ratio is at most 1.998. While the exact spanning ratio of the standard Delaunay triangulation remains unknown, several variants do have known tight spanning ratios in the worst case. For example, when the empty disk is replaced with a square we obtain the L_∞ or L_1 -Delaunay graph, which is known to have a spanning ratio of exactly $\sqrt{4 + 2\sqrt{2}} \approx 2.61$ [9]. Similar proof techniques have been generalized to Delaunay graphs based on rectangles and parallelograms [15, 43]. In general, one can define a Delaunay graph from any convex distance function, and such a graph is known to have a constant spanning ratio where the spanning ratio depends on the ratio of the perimeter to the width of the convex shape [11]. When the unit circle in this distance is a regular hexagon, then the exact worst-case spanning ratio is 2 [40]. When the unit circle is an equilateral triangle, then exact worst-case spanning ratio is also 2 [27]. A generalized $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graph is a TD-Delaunay graph whose empty region is a scaled translate of a triangle with angles of $\theta_1, \theta_2, \theta_3 := \pi - \theta_1 - \theta_2$ with $\theta_1 \leq \theta_2 \leq \theta_3$. In this paper, we provide a lower bound of $\frac{1}{\sin(\theta_1/2)}$ that matches the best known upper bound for $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graphs [36].

The routing ratio of a geometric graph essentially captures how feasible it is to find short paths in a graph when making local decisions based only on the neighbourhood of the current vertex. The routing ratio is the smallest upper bound on the ratio of the length of the path returned by the routing algorithm and the Euclidean distance between all pairs of vertices. Routing in Delaunay triangulations is notoriously difficult, with the routing ratio of the standard Delaunay triangulation known to be between 1.70 [6] and 3.56 [6]. Variations such as the L_1 -Delaunay triangulation are known to have a routing ratio between 2.7 [6] and 3.16 [26].

For TD-Delaunay graphs, there is a gap between the spanning ratio of 2 and the routing ratio which was shown to be exactly $\frac{5}{\sqrt{3}}$ in the worst-case [20]. We show that this gap is preserved for $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graphs by extending techniques from [20] to obtain a tight routing ratio of

$$C(\theta_1, \theta_2) := \max_{\substack{j \in \{1, 2, 3\} \\ 0 \leq \alpha \leq \theta_j}} \left(\frac{\sin(\theta_j - \alpha)}{\sin(\theta_{j+1})} + \frac{\sin(\alpha)}{\sin(\theta_{j-1})} + \min \left(\frac{\sin(\alpha)}{\sin(\theta_{j-1})} + \frac{\sin(\alpha + \theta_{j-1})}{\sin(\theta_{j+1})}, \right. \right. \\ \left. \left. \frac{\sin(\theta_j - \alpha)}{\sin(\theta_{j+1})} + \frac{\sin(\alpha + \theta_{j-1})}{\sin(\theta_{j-1})} \right) \right).$$

3.3 Preliminaries

We will denote the line segment from point u to point v as uv , and the length of uv is denoted $|uv|$. For two vertices u, v in a geometric graph G , the length of the shortest path from u to v in G is denoted $d_G(u, v)$. Then for a real number $c \geq 1$, G is said to be a c -spanner if for all vertices u, v in G , we have $d_G(u, v) \leq c|uv|$. The spanning ratio of a class of graphs $\mathcal{G}$ is the infimum of $\{c \in \mathbb{R} \mid \forall G \in \mathcal{G}, G \text{ is a } c\text{-spanner}\}$. In a geometric graph, each vertex is identified with its coordinates. Here, one unit of memory is $\log_2(|V(G)|)$ bits in order to store points on a polynomially-sized grid. The k -neighbourhood of a vertex u in a graph is defined to be all the vertices v such that there is a path from u to v consisting of k or fewer edges. Formally, a k -local, m -memory routing algorithm is a function that takes as input $(s, N_k(s), t, M)$, and

outputs a vertex p where s is the current vertex, $N_k(s)$ is the k -neighbourhood of s , t is the destination, M is an m -bit memory register, and $p \in N_1(s)$. An algorithm is said to have a routing ratio of at most c for a family of geometric graphs $\mathcal{G}$ if the path output by the algorithm for any pair of vertices $s, t \in V(G)$ for $G \in \mathcal{G}$ has length at most $c|st|$. The routing ratio of an algorithm $\mathcal{A}$ is the infimum over all c for which the algorithm is c -competitive for $\mathcal{G}$.

Throughout this paper, we fix a triangle Δ in the plane with angles $\theta_1 \leq \theta_2 \leq \theta_3$. We assume that the corresponding corners of Δ are labelled τ_1, τ_2, τ_3 . In order to keep notation cleaner, we use arithmetic modulo 3 for operations on index i when referring to corners of triangles. For example, $\tau_4 = \tau_1$, and $\tau_0 = \tau_3$. By convention, the expression $\angle abc$ will refer to the smaller angle among the clockwise and counterclockwise angles between ab and bc for three non-collinear points $a, b, c \in \mathbb{R}^2$.

For any two points u, v in the plane, define the triangle $T^{u,v}$ to be the smallest scaled translate of Δ with u and v on its boundary. Note that *smallest* implies that at least one of u, v is on a corner of $T^{u,v}$. We use $\tau_i^{u,v}$ to refer to the corner of triangle $T^{u,v}$ corresponding to τ_i . Now we define the cones, depicted in Figure 3.1. In particular, for a point p and index $i \in \{1, 2, 3\}$, let $C_{p,i} := \{v \in \mathbb{R}^2 \mid p = \tau_i^{p,v}\}$ be the positive cone centred at point p corresponding to τ_i . On the other hand, define the negative cone $\overline{C}_{p,i} := \{v \in \mathbb{R}^2 \mid v = \tau_i^{p,v}\}$. Note that $\overline{C}_{p,i}$ is $C_{p,i}$ rotated by π radians about p .

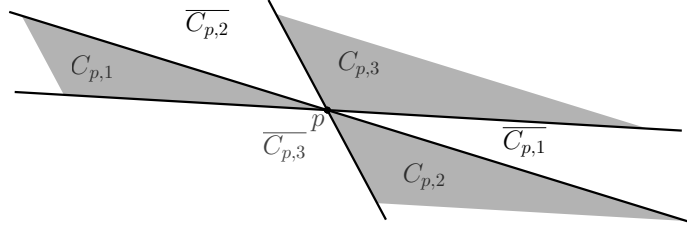


Fig. 3.1: $C_{p,1}, C_{p,2}, C_{p,3}$ are the positive cones of p and $\overline{C}_{p,1}, \overline{C}_{p,2}, \overline{C}_{p,3}$ are the negative cones of p .

The TD-Delaunay graph of a vertex set $S \subseteq \mathbb{R}^2$ has an edge between vertices u and v when there exists an equilateral triangle with u, v on its boundary and no other points of S in its interior. Note that the equilateral triangle is a scaled translate of the TD unit circle. As with any Delaunay graph based on a convex distance function, every bounded face is a triangle [11]. To define the $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graph, we replace the equilateral triangle with Δ containing angles of $\theta_1, \theta_2, \theta_3$. Equivalently, if F is the affine transformation that brings Δ to the equilateral triangle (the unit circle in the triangle distance), then there is an edge uv in the $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graph of a set $S \subseteq \mathbb{R}^2$ exactly when $F(u)F(v)$ is an edge of the TD-Delaunay graph of $F(S)$. This alternative definition immediately leads to a local routing strategy for the $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graph of a point set S : use the existing routing algorithm from [20] on the TD-Delaunay graph of $F(S)$. In Section 3.5.2, we show that this approach is not optimal.

Bonichon et al. [7] showed that the TD-Delaunay graph corresponds to the half- θ_6 -graph. Analogous to the half- θ_6 -graph, Lubiw and Mondal [36] define the 3-sweep graph, which directly corresponds to the $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graph. The 3-sweep graph G gets its name from an alternative, yet equivalent, construction. For each vertex u and each positive cone $C_{u,i}$, include in G the edge to the *nearest* vertex $v \in C_{u,i}$. By *nearest*, we mean that the triangle $T^{u,v}$ is minimal among $\{T^{u,v'} \mid v' \in C_{u,i}\}$. In this way, one can picture the leading edge $\tau_{i-1}^{u,v} \tau_{i+1}^{u,v}$ *sweeping* through cone $C_{u,i}$. Throughout the paper, we assume that no two points lie on a line parallel to a cone boundary. This ensures that each vertex has at most one neighbour in each positive cone.

One desirable property of paths is angle monotonicity. A path is angle monotone with width α if the vector of each edge on the path lies in a cone with apex angle α . Lubiw and Mondal show that the 3-sweep graph has certain angle-monotone properties which are used to upper bound the spanning ratio, see Observation 1.

Observation 1. *An angle monotone path from s to t with width α has length at most $\frac{|st|}{\cos(\alpha/2)}$ [36].*

In [36], Lubiw and Mondal also define a k -layered 3-sweep graph by combining k copies of rotated 3-sweep graphs, and provide a local routing algorithm that finds angle monotone paths in k -layered 3-sweep graphs. Note that since k is at least 4, their routing algorithm does not apply to $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graphs.

3.3.1 Our Contributions

In Section 3.4, we prove that $\frac{1}{\sin(\theta_1/2)}$ is a lower bound on the spanning ratio of $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graphs which matches the best known upper bound [36]. Then in Section 3.5, we provide a lower bound on the routing ratio by showing that there exist $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graphs for which the routing ratio of any k -local routing algorithms is at least as large as $C(\theta_1, \theta_2)$. Then, we show that our lower bound is tight by providing an online local routing algorithm for $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graphs with a routing ratio of $C(\theta_1, \theta_2)$. Finally, in Section 3.5.2, we compare our optimal routing algorithm to the previously best-known approach to routing in $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graphs.

3.4 Spanning Ratio

We present a lower bound in the following proposition.

Proposition 1. *For every $\epsilon > 0$, there exists a set of points $S \subseteq \mathbb{R}^2$ such that the $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graph of S has a spanning ratio of at least $\frac{1}{\sin(\theta_1/2)} - \epsilon$*

Proof. We will construct a point set $S = \{a, b, \tau_1, \tau_2, \tau_3\}$ such that $d_G(a, b)$ approaches $\frac{|ab|}{\sin(\theta_1/2)}$, where G is the $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graph of S . See Figure 3.2. Place two points a, b outside Δ each at a distance $\frac{\min(|\tau_1 \tau_2|, |\tau_1 \tau_3|)}{2}$ from τ_1 , with a arbitrarily close to $\tau_1 \tau_2$ and b arbitrarily close to $\tau_1 \tau_3$. By construction of S , G has edges $\tau_1 \tau_2$, $\tau_2 \tau_3$, $\tau_1 \tau_3$, $\tau_1 a$, $\tau_2 a$, $\tau_1 b$ and $\tau_3 b$.

The shortest path in G from a to b passes through τ_1 , meaning the spanning ratio is at least $\frac{|a\tau_1| + |\tau_1 b|}{|ab|}$. This value can be made arbitrarily close to $\frac{1}{\sin(\theta_1/2)}$ as a and b move closer to

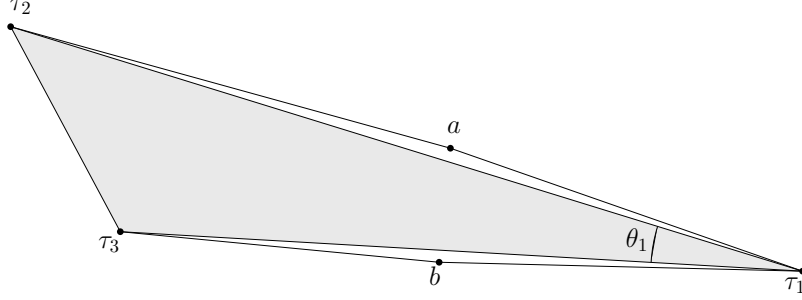


Fig. 3.2: The shortest path from a to b passes through τ_1 in the TD_{θ_1, θ_2} -Delaunay graph G of the point set $\{a, b, \tau_1, \tau_2, \tau_3\}$. As a, b get closer to Δ , then $d_G(a, b)$ approaches $\frac{|ab|}{\sin(\theta_1/2)}$.

the boundary of Δ . While this point set may not be in general position, the vertices can be perturbed to satisfy the general position constraint. $\square$

The upper bound of $\frac{1}{\sin(\theta_1/2)}$ follows from Lemma 6 of [36] by Lubiw and Mondal.

3.5 Local Routing

Local routing has been studied in many contexts, and in Section 3.5.2, we will show that the known routing algorithms do not give optimal results in TD_{θ_1, θ_2} -Delaunay graphs. In this section, we provide an optimal local routing algorithm. Our approach is to generalize the algorithm from [20], leading to our algorithm (refer to Algorithm 1). The key algorithmic insight lies in the threshold for making decisions in routing. Each decision is carefully made to reduce the total path length. The goal of this section is to prove the following theorem.

Theorem 1. *The routing ratio of the TD_{θ_1, θ_2} -Delaunay graph is at most $C(\theta_1, \theta_2)$. Furthermore, this bound is tight in the worst case.*

We will start with the following proposition:

Proposition 2. *Let k be a positive integer and let $\epsilon > 0$. Every k -local routing algorithm for TD_{θ_1, θ_2} -Delaunay graphs must have a routing ratio at least $C(\theta_1, \theta_2) - \epsilon$.*

Proof. We will construct two vertex sets S_1 and S_2 and refer to their corresponding TD_{θ_1, θ_2} -Delaunay graphs as G_1 and G_2 . Importantly, the k -neighbourhoods of G_1 and G_2 around the start vertices s are identical, however the rest of the graphs will be vastly different. In this way any algorithm that performs well for one graph will not for the other. These are analogous to the constructions of Figure 12 in [20]. Assume $j = 3$ maximizes the expression of $C(\theta_1, \theta_2)$. Let s be on $\tau_1\tau_2$. Place p_1 inside $C_{s,1} \cap C_{\tau_2,2}$ arbitrarily close to τ_2 , then place q_1 in $C_{s,2} \cap C_{p_1,2} \cap C_{\tau_1,1}$ arbitrarily close to τ_1 . Next, place p_2 on segment τ_3p_1 in cone $C_{q_1,1}$, arbitrarily close to p_1 . Next, for $i = 2, \dots, k$, place q_i such that triangle τ_3, p_i, q_i is similar to τ_3, p_1, q_1 , and place p_{i+1} such that triangle τ_3, p_{i+1}, q_i is similar to τ_3, p_2, q_1 . Finally, let $S_1 = \{s, p_1, \dots, p_k, q_1, \dots, q_k, \tau_3\}$

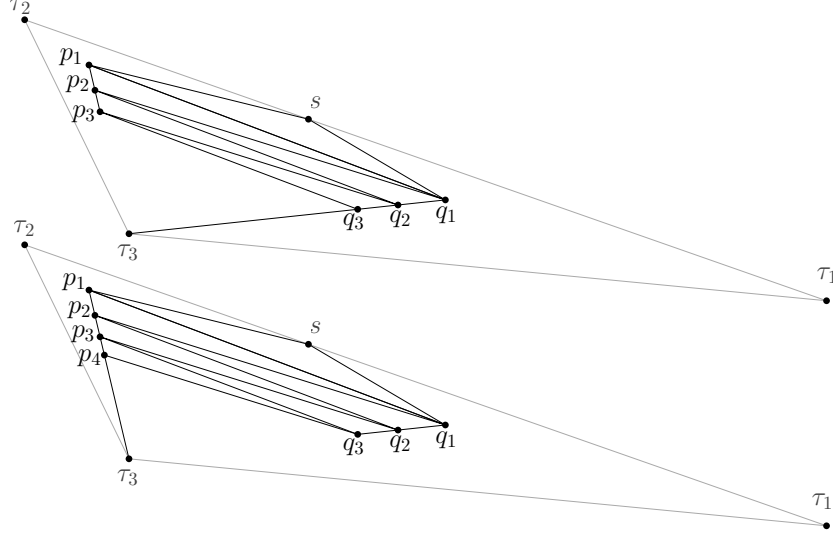


Fig. 3.3: The $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graphs G_1 and G_2 constructed for the lower bound of k -local routing from s to τ_3 . In this example, $k = 3$.

and $S_2 = \{s, p_1, \dots, p_k, p_{k+1}, q_1, \dots, q_k, \tau_3\}$. This construction ensures that G_1 contains the edges $sp_1, sq_1, p_1q_1, p_{i-1}p_i, q_{i-1}q_i, q_{i-1}p_i, p_iq_i, q_k\tau_3$ where $i \in \{2, \dots, k\}$. On the other hand, G_2 contains the edges $sp_1, sq_1, p_1q_1, p_{i-1}p_i, q_{i-1}q_i, q_{i-1}p_i, p_iq_i, p_kp_{k+1}, q_kp_{k+1}, p_{k+1}\tau_3$ where $i \in \{2, \dots, k\}$. Importantly, G_1 does not contain the edge τ_3p_k because q_k is the closest neighbour to τ_3 in the cone $C_{\tau_3, 3}$. Similarly, G_2 does not contain the edge τ_3q_k because p_{k+1} is the closest neighbour to τ_3 in the cone $C_{\tau_3, 3}$. Similarly, the edges $p_k\tau_3$ and $q_k\tau_3$ do not exist in G_1 and G_2 , respectively, since τ_3 is in a negative cone of p_k and q_k .

Since the k -neighbourhood of s in G_1 and G_2 is $\{s, p_1, \dots, p_k, q_1, \dots, q_k\}$, then any algorithm routing from s to τ_3 will choose the same first vertex (p_1 or q_1) in G_1 and G_2 . Moreover, τ_3 only has one neighbour in each graph, so any path from s to τ_3 must pass through q_k in G_1 and through p_{k+1} in G_2 . Then any algorithm that visits p_1 first will output a path from s to τ_3 in G_1 of length at least $|sp_1| + |p_1q_k| + |q_k\tau_3|$. On the other hand, any algorithm that chooses to visit q_1 first will output a path from s to τ_3 in G_2 having length at least $|sq_1| + |q_1p_{k+1}| + |p_{k+1}\tau_3|$. Since p_1 is arbitrarily close to τ_2 , p_2 is arbitrarily close to p_1 , and q_1 is arbitrarily close to τ_1 , then each p_i is arbitrarily close to τ_2 and each q_i is arbitrarily close to τ_1 . Then, for any $\epsilon > 0$, the routing ratio of any algorithm is at least

$$\begin{aligned} & \frac{\min(|s\tau_2| + |\tau_2\tau_1| + |\tau_1\tau_3|, |s\tau_1| + |\tau_1\tau_2| + |\tau_2\tau_3|)}{|s\tau_3|} - \epsilon \\ &= \frac{|s\tau_1|}{|s\tau_3|} + \frac{|s\tau_2|}{|s\tau_3|} + \min\left(\frac{|s\tau_2|}{|s\tau_3|} + \frac{|\tau_1\tau_3|}{|s\tau_3|}, \frac{|s\tau_1|}{|s\tau_3|} + \frac{|\tau_2\tau_3|}{|s\tau_3|}\right) - \epsilon. \end{aligned}$$

Finally, we obtain $C(\theta_1, \theta_2) - \epsilon$ by the law of sines in triangles $s\tau_2\tau_3$ and $\tau_1s\tau_3$, where angle $\alpha := \angle\tau_2\tau_3s$, since

$$\frac{|s\tau_2|}{\sin(\alpha)} = \frac{|s\tau_3|}{\sin(\theta_2)} = \frac{|\tau_2\tau_3|}{\sin(\pi - \alpha - \theta_2)}, \quad \text{and} \quad \frac{|s\tau_1|}{\sin(\theta_3 - \alpha)} = \frac{|\tau_1\tau_3|}{\sin(\alpha + \theta_2)} = \frac{|s\tau_3|}{\sin(\theta_1)}.$$

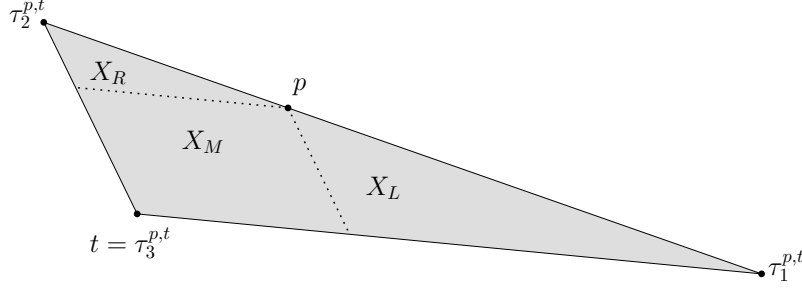


Fig. 3.4: $T^{p,t}$ is the smallest scaled translate of Δ with p and t on its boundary.

□

3.5.1 Local Routing Algorithm

In this section, we present Algorithm 1 which is a 1-local, 0-memory routing algorithm for TD_{θ_1, θ_2} -Delaunay graphs. It is generalized from the routing algorithm by Bose et al. [20]. Let s be the start vertex, t be the target vertex, and p be the current vertex. At each step of Algorithm 1, the next vertex is chosen based on the four cases (i), (ii), (iii), or (iv). To ease notation for cases (ii), (iii), and (iv), we will define the left, middle, and right regions of p : X_L , X_M , and X_R respectively, pictured in Figure 3.4. When t lies in a negative cone $\overline{C_{p,i}}$, then let $X_L := C_{p,i-1} \cap T^{p,t}$, $X_R := C_{p,i+1} \cap T^{p,t}$, and $X_M := \overline{C_{p,i}} \cap T^{p,t}$.

In short, the algorithm prefers to route in the region towards t , however when this is not possible, it stays close to a neighbouring empty region or the side that minimizes a possible detour. Now we will prove the following upper bound:

Proposition 3. *Let s, t be two vertices in a TD_{θ_1, θ_2} -Delaunay graph G . When t is in a negative cone of s , then the path $P_{s,t}$ output by Algorithm 1 from s to t in G has ratio $\frac{|P_{s,t}|}{|st|}$ at most $C(\theta_1, \theta_2)$. When t is in a positive cone of s , then $\frac{|P_{s,t}|}{|st|}$ is at most $\frac{1}{\sin(\theta_1/2)}$.*

Notice that when the angles $\theta_1, \theta_2, \theta_3$ are all equal to $\frac{\pi}{3}$, then $C(\theta_1, \theta_2)$ in Proposition 2 for routing in a negative cone reaches a maximum of $5/\sqrt{3}$ when $\alpha = \frac{\pi}{6}$, matching the bound from [20]. Furthermore, the expression for routing in a positive cone matches the spanning ratio.

Proof. We will bound the path chosen by Algorithm 1 by defining a potential for each vertex along a path and showing that at each step, the potential drops by at least the length of the chosen edge. We define the potential as follows, depicted in Figure 3.5.

- Case (i): $\Phi(p, t) := \max_{j=\pm 1} (|p\tau_{i+j}^{p,t}| + |\tau_{i+j}^{p,t}t|)$.
- Case (ii): $\Phi(p, t) := \min_{j=\pm 1} (|p\tau_{i+j}^{p,t}| + |\tau_{i+j}^{p,t}t|)$.
- Case (iii): $\Phi(p, t) := |p\tau_{i+j}^{p,t}| + |\tau_{i+j}^{p,t}t|$ where the empty region (X_L or X_R) is $C_{p,i+j} \cap T^{p,t}$.

Algorithm 1: Local Routing algorithm in TD_{θ_1, θ_2} -Delaunay graph G

Data: Two points $s, t \in S$

Result: Path in G from s to t

$p \leftarrow s$;

while $p \neq t$ **do**

 Choose the next vertex v based on the following cases, then set $p \leftarrow v$

 (i) Case: t lies in a positive cone $C_{p,i}$.

 Follow the unique edge pv in $C_{p,i}$.

 (ii) Case: t lies in a negative cone $\overline{C_{p,i}}$, and both regions X_L and X_R are empty

 Let $j \in \{1, -1\}$ minimize $|p\tau_{i+j}^{p,t}| + |\tau_{i+j}^{p,t}t|$. Choose the neighbour $v \in X_M$ closest to $C_{p,i+j}$ in cyclic order about p .

 (iii) Case: t lies in a negative cone $\overline{C_{p,i}}$, and only one region of $\{X_L, X_R\}$ is empty.

 If p has neighbours in X_M , choose the neighbour v closest to the empty region in cyclic order about p . Otherwise, choose the unique neighbour v in the non-empty region.

 (iv) Case: t lies in a negative cone $\overline{C_{p,i}}$, and neither X_L nor X_R is empty.

 If p has neighbours in X_M , choose an arbitrary $v \in X_M$. Otherwise, let $j \in \{1, -1\}$ minimize $|p\tau_{i+j}^{p,t}| + |\tau_{i-j}^{p,t}t|$, and choose v in $C_{p,i+j}$.

end

- Case (iv): $\Phi(p, t) := \min_{j=\pm 1} (|p\tau_{i+j}^{p,t}| + |\tau_{i+j}^{p,t}\tau_{i-j}^{p,t}| + |\tau_{i-j}^{p,t}t|)$.

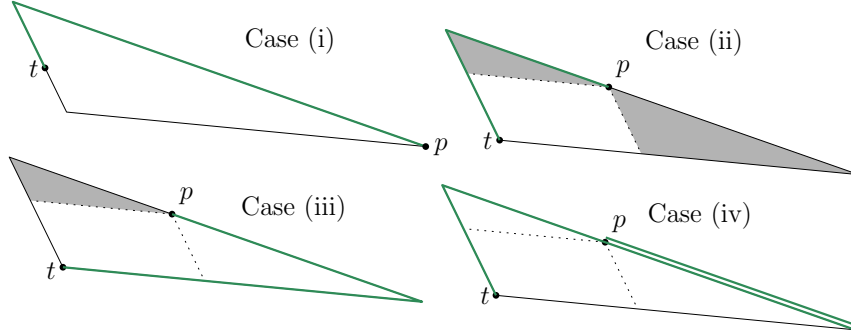


Fig. 3.5: The potential is given by the green path. In this example, i takes values 1, 3, 3, 3 for cases (i), (ii), (iii), (iv) respectively. The grey regions are empty.

Now, we will show that in each case of Algorithm 1, the length of each chosen edge is less than the drop in potential. More precisely, we want to show $|pv| + \Phi(v, t) \leq \Phi(p, t)$ for cases (i), (ii), (iii), and (iv).

Suppose the current vertex is p and the case is (i), as can be seen in Figure 3.6. Then after an edge pv is chosen, the current vertex will proceed to v and the case will be either (i), (ii), or (iii). Case (iv) is not possible when t lies in a negative cone of v because at least one of the regions of v is empty. Then the next potential, $\Phi(v, t)$, passes through some vertex $\tau_{i+k}^{v,t}$ for $k = \pm 1$. We have

$$\begin{aligned}
 |pv| + \Phi(v, t) &\leq (|p\tau_{i+k}^{p,v}| + |\tau_{i+k}^{p,v}v|) + (|v\tau_{i+k}^{v,t}| + |\tau_{i+k}^{v,t}t|) \\
 &= (|p\tau_{i+k}^{p,v}| + |v\tau_{i+k}^{v,t}|) + (|\tau_{i+k}^{p,v}v| + |\tau_{i+k}^{v,t}t|) \\
 &= |p\tau_{i+k}^{p,t}| + |\tau_{i+k}^{p,t}t| \\
 &\leq \max_{j=\pm 1} (|p\tau_{i+j}^{p,t}| + |\tau_{i+j}^{p,t}t|) = \Phi(p, t)
 \end{aligned}$$

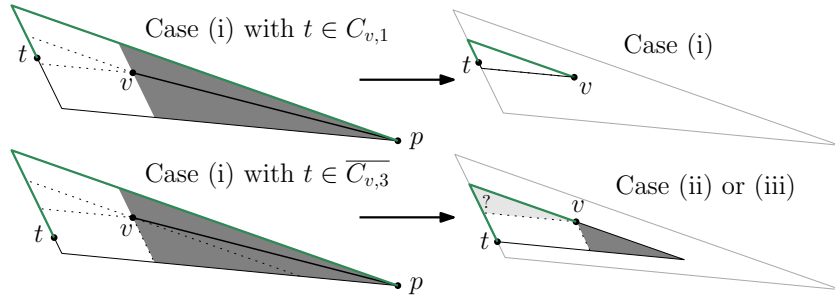


Fig. 3.6: Bounding the potential in case (i) since t lies in $C_{p,1}$. The dark grey regions are empty.

Next, suppose the current vertex is p and the case is (ii), depicted in Figure 3.7. Let j minimize the expression from $\Phi(p, t)$. Notice that since we choose the edge closest in cyclic

order about p to the region $C_{p,i+j} \cap T^{p,t}$, then we can deduce that v has no neighbours in its region $C_{v,i+j} \cap T^{v,t}$. Therefore once the current vertex proceeds to v , then the possible cases are only (ii) or (iii). Then we have

$$\begin{aligned}
|pv| + \Phi(v, t) &\leq (|p\tau_{i+j}^{p,v}| + |\tau_{i+j}^{p,v}v|) + (|v\tau_{i+j}^{v,t}| + |\tau_{i+j}^{v,t}t|) \\
&= (|p\tau_{i+j}^{p,v}| + |v\tau_{i+j}^{v,t}|) + (|\tau_{i+j}^{p,v}v| + |\tau_{i+j}^{v,t}t|) \\
&= |p\tau_{i+j}^{p,t}| + |\tau_{i+j}^{p,t}t| = \Phi(p, t)
\end{aligned}$$

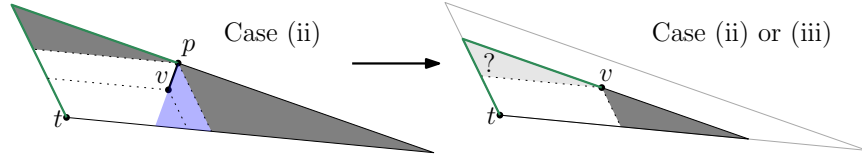


Fig. 3.7: Case (ii) when t lies in $\overline{C_{p,3}}$ and $j = -1$. The dark grey and blue regions are empty.

Now suppose p is the current vertex, the case is (iii), and $C_{p,i+j} \cap T^{p,t}$ is the empty region, shown in Figure 3.8. If X_M is not empty, then the choice of closest neighbour v to $C_{p,i+j} \cap T^{p,t}$ guarantees that the corresponding region $C_{v,i+j} \cap T^{v,t}$ of v is also empty. Likewise, if X_M is empty, then choosing the unique neighbour in $C_{p,i-j} \cap T^{p,t}$ again guarantees that v has an empty region $C_{v,i+j} \cap T^{v,t}$. Either way, once the current vertex continues to v , then the case must be either (ii) or (iii). Then the exact same sequence of inequalities as from case (ii) completes the argument.

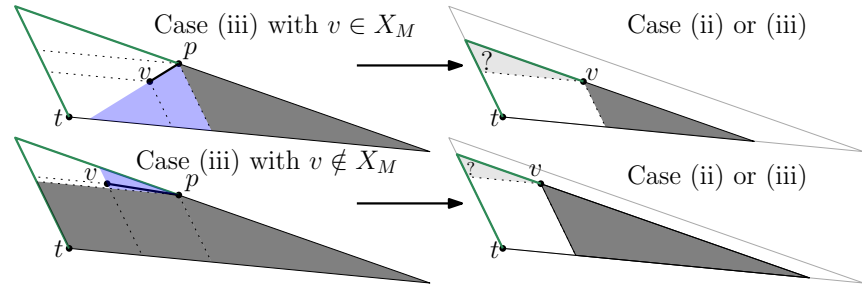


Fig. 3.8: Case (iii) when t lies in $\overline{C_{p,3}}$ and $j = -1$. The dark grey and blue regions are empty.

Finally, we move on to case (iv), where p is the current vertex. After choosing the next vertex v , the next possible cases are (ii), (iii), or (iv). Let $j = \pm 1$ minimize the expression of $\Phi(p, t)$. In any case, we have $\Phi(v, t) \leq (|v\tau_{i+j}^{v,t}| + |\tau_{i+j}^{v,t}\tau_{i-j}^{v,t}| + |\tau_{i-j}^{v,t}t|)$ by the triangle inequality. When $v \in X_M$, then we use the following inequalities to prove the claim, also shown in Figure 3.9.

1. $|pv| \leq |p\tau_{i-j}^{v,p}| + |\tau_{i-j}^{v,p}v|$ by triangle inequality
2. $|\tau_{i-j}^{v,t}t| + |\tau_{i-j}^{v,p}v| = |\tau_{i-j}^{p,t}t|$ by projection

3. $|p\tau_{i-j}^{v,p}| \leq |\tau_{i+j}^{v,p}\tau_{i-j}^{v,p}|$ since p lies on $\tau_{i+j}^{v,p}\tau_{i-j}^{v,p}$
4. $|\tau_{i+j}^{v,t}\tau_{i-j}^{v,t}| + |\tau_{i+j}^{v,p}\tau_{i-j}^{v,p}| = |\tau_{i+j}^{p,t}\tau_{i-j}^{p,t}|$ by translation and projection
5. $|v\tau_{i+j}^{v,t}| \leq |p\tau_{i+j}^{t,p}|$ by projection

When v is not in X_M , then let u be the intersection of $p\tau_i^{p,v}$ and $\tau_{i-j}^{t,v}\tau_{i+j}^{t,v}$. The following inequalities suffice to prove the claim.

1. $|pv| \leq |pu| + |uv|$ by triangle inequality
2. $|uv| + |v\tau_{i+j}^{v,t}| \leq |p\tau_{i+j}^{p,t}|$ by projection
3. $|\tau_{i-j}^{v,t}t| + |pu| = |\tau_{i-j}^{p,t}t|$ by projection
4. $|\tau_{i+j}^{v,t}\tau_{i-j}^{v,t}| \leq |\tau_{i+j}^{p,t}\tau_{i-j}^{p,t}|$ by projection

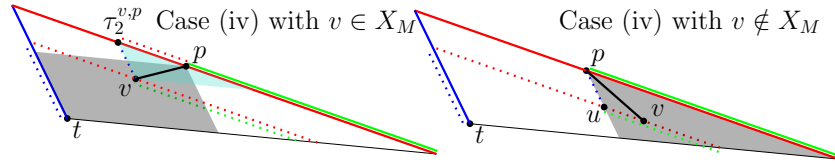


Fig. 3.9: Bounding the potential in case (iv) when t lies in $\overline{C_{p,3}}$, with $i = 3, j = 1$. The dotted paths representing $|pv| + \Phi(v, t)$ are shorter than the corresponding solid paths of $\Phi(p, t)$. The grey region contains v , and the blue triangle is $T^{v,p}$.

Since $\Phi(t, t) = 0$, then the path from s to t output by Algorithm 1 can have length at most $\Phi(s, t)$. When t is in a positive cone of s , then the potential is defined using case (i). The corresponding path $p\tau_{i+j}^{p,t} + \tau_{i+j}^{p,t}t$ is $\pi - \theta_{i+j}$ monotone, then the routing ratio in such a case can be at most $\frac{1}{\sin(\theta_1/2)}$ by Observation 1.

On the other hand, when t is in a negative cone of s , there are three possible cases: (ii), (iii) or (iv). The triangle inequality tells us that $\Phi(s, t)$ is largest in case (iv). Then, similar to the proof of Proposition 2, the routing ratio is bounded by $C(\theta_1, \theta_2)$ using the law of sines. $\square$

Finally, Theorem 1 is a consequence of Propositions 2 and 3 since Algorithm 1 is 1-local.

3.5.2 Comparison to known routing algorithms

In this subsection, we show that currently known local routing algorithms when applied on the TD_{θ_1, θ_2} -Delaunay graph are suboptimal. Firstly, note that by using a stretch factor upper bound from Section 3.4, we can apply the technique of Bose and Morin [22] to obtain a local routing algorithm that finds a path between any two vertices with length at most 9 times the

stretch factor, which is not optimal. Another approach is to route in $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graphs by combining the algorithm of Bose et al. [20] with an affine transformation. When $\triangle$ is the equilateral triangle, then Algorithm 1 simplifies to the standard TD-Delaunay routing algorithm from [20]. In this case, notice that the thresholds in cases (ii) and (iv) simplify so that j refers to the corner $\tau_{i+j}^{p,t}$ nearest p . In other words, the decision threshold is the midpoint of the segment $\tau_{i+1}^{p,t} \tau_{i-1}^{p,t}$. We will analyze this standard TD-Delaunay routing algorithm when it is used on the affine transformation of a general $\text{TD}_{\theta_1, \theta_2}$ -Delaunay graph. Since affine transformations preserve midpoints, then the decision threshold in case (iv) is also the midpoint of the segment $\tau_{i+1}^{p,t} \tau_{i-1}^{p,t}$. It is in this way that applying an affine transformation to the existing algorithm differs from our Algorithm 1. To see the difference in routing ratio of these two approaches, consider the construction of G_1 from Proposition 2. If we enforce $|s\tau_2| < |s\tau_1|$, then the path output by the affine transformation of the standard TD-Delaunay routing algorithm would choose to visit p_1 first. The routing ratio of this algorithm would therefore be at least

$$\frac{\sin(\theta_3 - \alpha)}{\sin(\theta_1)} + \frac{\sin(\alpha)}{\sin(\theta_2)} + \frac{\sin(\alpha)}{\sin(\theta_2)} + \frac{\sin(\alpha + \theta_2)}{\sin(\theta_1)} - \epsilon$$

where $\alpha := \angle \tau_2 \tau_3 s$. For example, when $\theta_1 = \frac{\pi}{6}$, $\theta_2 = \frac{\pi}{5}$, and $\alpha = \frac{\pi}{3}$ then the routing ratio of the standard TD-Delaunay algorithm under an affine transformation is strictly more than 6.55, whereas the optimal routing ratio is less than 6.52 by Proposition 3.

Chapter 4

Improved Routing for Polygon-Delaunay Graphs

Prosenjit Bose, Jean-Lou De Carufel, John Stuart

4.1 Abstract

The standard Delaunay triangulation is a geometric graph whose vertices are points in the plane, and two vertices share an edge if they lie on the boundary of an empty disk. If the disk is replaced with a homothet of a fixed convex shape C , then the resulting graph is called a C -Delaunay graph. We study the problem of local routing in C -Delaunay graphs where C is a regular polygon having five to eight sides. In particular, we generalize the routing algorithm of Chew for square-Delaunay graphs (Chew. *SCG 1986*, 169–177) in order to obtain the following approximate upper bounds of 4.640, 6.429, 8.531 and 4.054 on the spanning and routing ratios for pentagon-, hexagon-, septagon-, and octagon-Delaunay graphs, respectively. The exact expression for the upper bounds of the routing ratio is

$$\Psi(n) := \begin{cases} \sqrt{1 + ((\cos(2\pi/n) + n - 1)/\sin(2\pi/n))^2} & \text{if } n \in \{5, 6, 7\}, \\ \sqrt{1 + ((\cos(\pi/8)\cos(3\pi/8) + 3)/(\cos(\pi/8)\sin(3\pi/8)))^2} & \text{if } n = 8. \end{cases}$$

We show that these bounds are tight for the output of our routing algorithm by providing a point set where these bounds are achieved. We also include lower bounds of 1.708 and 1.995 on the spanning and routing ratios of the pentagon-Delaunay graph.

Our upper bounds yield a significant improvement over the previous routing ratio upper bounds for this problem, which previously sat at around 400 for the pentagon, septagon, and octagon [11, 22] as well as 18 for the hexagon [22, 40]. Our routing ratios also provide significant improvements over the previously best known spanning ratios for pentagon-, septagon- and octagon-Delaunay graphs, which were around 45 [11].

4.2 Introduction

A geometric graph is a weighted graph whose vertices are points in the plane and edges are line segments weighted with the Euclidean distance between their endpoints. Two of the main distance-preserving properties of a graph are the spanning ratio and routing ratio. The spanning ratio of a pair of points is the ratio of the shortest path between them in the graph divided by

their Euclidean distance, and the spanning ratio of a graph is the maximum spanning ratio over all pairs of points [39]. On the other hand, the routing ratio is defined similarly, except that the path is usually computed locally with only information of the current vertex's neighbourhood. Since a routing ratio is based on an algorithm that finds a path and the spanning ratio is based on the existence of a path, the spanning ratio of any graph is a lower bound on the routing ratio.

In this paper, we consider variants of standard Delaunay triangulations, which are geometric graphs with an edge between two points if there exists a disk with the endpoints on its boundary and no vertices in its interior. The spanning ratio of the Delaunay triangulation is known to be between 1.5932 [45] and 1.998 [44], however the exact value still remains unknown. The gap is even larger for the routing ratio, lying somewhere between 1.70 [6] and 3.56 [5]. Many papers study the related graphs that result from replacing the disk with a homothet of a fixed convex shape C , resulting in C -Delaunay graphs. Chew [26] proved that square-Delaunay graphs have a spanning ratio of at most $\sqrt{10}$ by giving a local routing algorithm. Subsequently, Chew [27] adapted his algorithm to equilateral triangle-Delaunay graphs to find a spanning ratio of 2, however the adapted algorithm was no longer a routing algorithm. In fact, Bose et al. [20] showed that the routing ratio of the equilateral triangle-Delaunay graph is exactly $\frac{5}{\sqrt{3}}$, showing the first separation between the spanning ratio and routing ratio. By generalizing Chew's algorithm, Bose et al. [6] were then able to show that the standard Delaunay triangulation has a routing ratio of at most 5.90 which was an improvement on the previously known upper bound of 15.48 [12]. Currently, the best-known bound is 3.56 [5]. In this paper, we show that Chew's algorithm can be further generalized to pentagon-, hexagon-, septagon-, and octagon-Delaunay graphs to obtain routing ratios of 4.640, 6.429, 8.531 and 4.054, respectively.

The hexagon-Delaunay graph is known to have a tight spanning ratio of 2 [40], however less is known about Delaunay graphs based on pentagons, septagons and octagons. With the exception of the hexagon-Delaunay graph, our routing ratio upper bounds yield a significant improvement over the previous best spanning ratio upper bounds. Bose et al. [11] give a spanning ratio upper bound for any C -Delaunay graph, where C is any convex shape. In particular, their bound is based intuitively on the thinness of C , which is essentially measured by the ratio of the perimeter to the width of C . For example, this ratio is π when C is a disk. Furthermore, by the construction of the paths from Bose et al. [11] and Perkovic et al. [40], it is possible to route using the algorithm of Bose and Morin [22] with a constant routing ratio of 9 times the spanning ratio. For each polygon, we compare our contribution to the previous best known upper bound in Table 4.1. We also prove lower bounds of 1.708 and 1.995 on the spanning and routing ratios of the pentagon-Delaunay graph in the appendix.

4.3 Preliminaries

We denote the line segment between points u, v as uv , and the Euclidean length of uv is denoted $|uv|$. For a path $\mathcal{P}$ in the plane, denote $|\mathcal{P}|$ as the length of the path. If paths $\mathcal{P}, \mathcal{Q}$ share an endpoint, then $\mathcal{P} + \mathcal{Q}$ denotes their concatenation. Next, for $a, b, c \in \mathbb{R}^2$, we define $\angle abc$ as the angle from ab to bc clockwise around b . The x and y coordinates of $a \in \mathbb{R}^2$ are denoted $x(a), y(a)$ respectively. For two vertices u, v in a geometric graph G , the length of the

C	Spanning Ratio	Routing Ratio	Our Routing and Spanning Ratio
Triangle	2[27]	$5/\sqrt{3}$ [20]	
Square	$\sqrt{4 + 2\sqrt{2}}$ [9]	$\sqrt{10}$ [26]	
Pentagon	≈ 45 [11]	≈ 405 [22]	≈ 4.640
Hexagon	2[40]	18[22]	≈ 6.429
Septagon	≈ 45 [11]	≈ 405 [22]	≈ 8.531
Octagon	≈ 43 [11]	≈ 387 [22]	≈ 4.054
Circle	≈ 1.998 [44]	≈ 3.56 [5]	

Table 4.1: Comparison to previously best-known upper bounds on the spanning and routing ratio of the C -Delaunay graph.

shortest path from u to v in G is denoted $d_G(u, v)$. Then for a real number $c \geq 1$, G is said to be a c -spanner if for all points u, v in G , we have $d_G(u, v) \leq c|uv|$. The spanning ratio of a class of graphs $\mathcal{G}$ is the infimum of $\{c \in \mathbb{R} \mid \forall G \in \mathcal{G}, G \text{ is a } c\text{-spanner}\}$.

We make the assumption that the graph is embedded on a polynomial-sized grid and therefore specifying the coordinates of a vertex in $V(G)$ requires $O(\log(|V(G)|))$ bits. Formally, a m -memory local routing algorithm is a function that takes as input $(s, N(s), t, M)$, and outputs some memory M' and a vertex $p \in N(s)$ where s is the current vertex, $N(s)$ is the neighbourhood of s , t is the destination, and both M, M' are bit-strings of length m . An algorithm is said to be c -competitive for a family of geometric graphs $\mathcal{G}$ if the path output by the algorithm for any pair of vertices $s, t \in V(G)$ for $G \in \mathcal{G}$ has length at most $c|st|$. The routing ratio of an algorithm is the infimum over all c for which the algorithm is c -competitive for $\mathcal{G}$. Note that the routing ratio is an upper bound on the spanning ratio.

For $n \in \{5, 6, 7, 8\}$, let $\bigcirc_n$ denote a regular n -gon in the plane. Every time we mention an n -gon, it is assumed to be a scaled translate of $\bigcirc_n$. Note that rotations are not permitted. We refer to the boundary of any n -gon C as ∂C , and to the interior as $\text{int}(C)$. We make the *general position* assumptions that no two points are on a line parallel to a side of $\bigcirc_n$, that neither coordinate axis is parallel to a side of $\bigcirc_n$, and that no four points lie on ∂C for some n -gon C . For two points $a, b \in \partial C$, define $\text{Arc}(C, a, b)$ to be the clockwise portion of ∂C from a to b . Let S be a set of points in the plane.

Definition 1. For $a, b \in S$, an edge ab satisfies the *empty- $\bigcirc_n$ property* with respect to S if there exists an n -gon C with $a, b \in \partial C$ and $S \cap \text{int}(C) = \emptyset$.

Definition 2. A *$\bigcirc_n$ -Delaunay graph* of S is a maximal planar graph on S such that every edge satisfies the empty- $\bigcirc_n$ property with respect to S . By maximal, we mean that no more edges satisfying the empty- $\bigcirc_n$ property can be added.

Note that specifying maximality in Definition 2 guarantees that every bounded face is a triangle [11]. Let u, v be two points in the plane that satisfy the general position assumption. Then $\text{Boundary}(u, v)$ denotes the set of n -gons C such that $u, v \in \partial C$. Also for any homothet C , define the point $\text{Center}(C)$ to be the point in C equidistant from all vertices of C . Furthermore,

denote $\text{North}(C)$ to be the vertex of C with the largest y -coordinate. Similarly, define $\text{East}(C)$, $\text{South}(C)$ and $\text{West}(C)$. For a set H of homothets of $\bigcirc_n$, let $\text{Center}(H) := \{\text{Center}(C) \mid C \in H\}$. Similarly, we define $\text{West}(H)$. For any homothet C of the n -gon $\bigcirc_n$, we label the vertices clockwise from $\text{West}(C)$ as $C^1, \dots, C^n$.

4.4 Routing Ratio Upper Bound

Recall that

$$\Psi(n) := \begin{cases} \sqrt{1 + ((\cos(2\pi/n) + n - 1)/\sin(2\pi/n))^2} & \text{if } n \in \{5, 6, 7\}, \\ \sqrt{1 + ((\cos(\pi/8)\cos(3\pi/8) + 3)/(\cos(\pi/8)\sin(3\pi/8)))^2} & \text{if } n = 8. \end{cases}$$

The goal of this section is to prove the following theorem.

Theorem 2. *The routing ratio of the $\bigcirc_5$ -Delaunay graph is at most $\Psi(5) \approx 4.64$.*

We will use n in place of 5 because the arguments hold for any integer $n \geq 5$, except in Claim 2. In particular, Section 4.4.3 discusses the task of generalizing the arguments to other values of n .

4.4.1 Local Routing Algorithm

We present Algorithm 2, which is a $O(\log(|V(G)|))$ -memory local routing algorithm for $\bigcirc_n$ -Delaunay graphs generalizing Chew's local routing algorithm [26]. Without loss of generality, we can rotate and reflect our frame of reference so that the start vertex s and destination t satisfy $y(s) = y(t)$ and $x(s) < x(t)$. The location of s is stored in memory for each step. In addition, we will assume that all edges of the convex hull are present in the $\bigcirc_n$ -Delaunay graph of S . Then in Section 4.4.2, we describe how Algorithm 2 can be modified to handle routing when the convex hull is not present.

The intuition behind Algorithm 2 is that when the current vertex is p_i , the next vertex p_{i+1} is restricted to one of the vertices of the rightmost triangle T_i of the graph containing vertex p_i and intersecting st . Note that the two neighbours of p_i under consideration are on opposite sides of st . Then, consider the empty n -gon C_i corresponding to T_i . We partition the boundary of C_i into two arcs by splitting at its west point w_i and its rightmost intersection with st , denoted t_i . If p_i is in the upper arc, we choose the clockwise neighbour, otherwise we choose the counterclockwise neighbour. A trace of Algorithm 2 is illustrated in Figure 4.1. Note that the triangles $T_0, \dots, T_k$ are ordered from left to right along st , so the algorithm terminates. We assume that there are $k + 1$ edges in the path output by Algorithm 2. Note that the last edge $p_k t$ in Algorithm 2 may appear to be a separate case from case (ii) in Algorithm 2, but we can avoid analyzing it separately by viewing t as both above and below st . The important detail in the analysis is that t is in the same portion of ∂C_k as p_k (either $p_k, t \in \text{Arc}(C_k, w_k, t_k)$ or $p_k, t \in \text{Arc}(C_k, t_k, w_k)$).

When u, v are in general position, the set $\text{Center}(\text{Boundary}(u, v))$ is analogous to the perpendicular bisector of uv when the n -gon $\bigcirc_n$ is replaced with a disk. For this reason, we will refer to $\text{Center}(\text{Boundary}(u, v))$ as $\text{Bisector}(u, v)$. In Lemma 2.2.1.1 of [38], Ma shows that

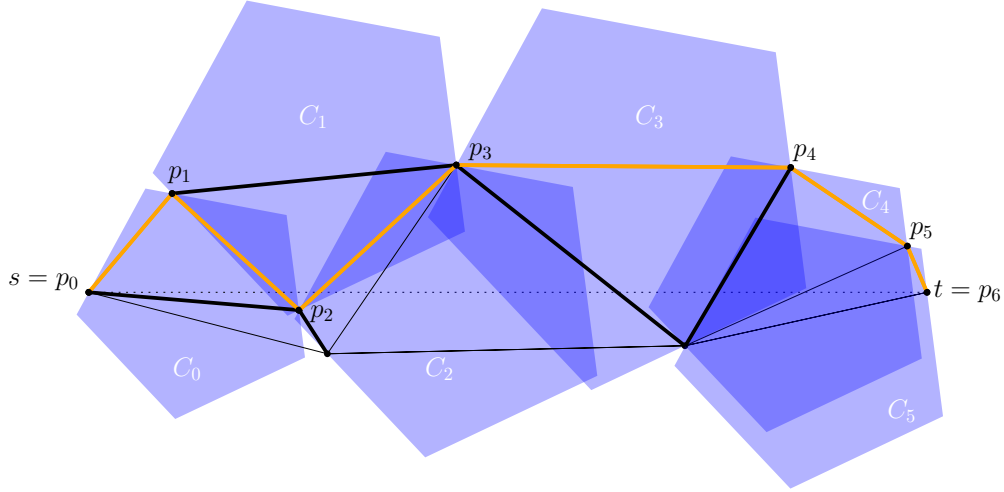


Fig. 4.1: Trace of Algorithm 2. In this case, $k = 5$ and $n = 5$. The orange line is the path chosen by Algorithm 2 and the thick black edges represent the other edge ($p_i a$ or $p_i b$) considered in step (ii).

for any regular n -gon, $\text{Bisector}(u, v)$ is a polygonal chain completed with two rays at the ends. In this way, $\text{Bisector}(u, v)$ partitions $\mathbb{R}^2$ into two half-spaces (see Figure 4.2).

When $y(v) > y(u)$, then there is a natural ordering of the points in $\text{Bisector}(u, v)$ from left to right. By convention, for any $u, v \in \mathbb{R}^2$ in general position, we say that the point $\text{Bisector}(u, v) \cap \text{Arc}(C, u, v)$ is to the left of the point $\text{Bisector}(u, v) \cap \text{Arc}(C, v, u)$. This is extended to an ordering on all the points of $\text{Bisector}(u, v)$. Note that with this convention, $\text{Bisector}(u, v)$ does not have the same ordering as $\text{Bisector}(v, u)$. For example, this convention tells us that in Figure 4.2, $\text{Center}(C_{i-1})$ is to the left of $\text{Center}(C_i)$ on $\text{Bisector}(q, p_i)$, whereas $\text{Center}(C_{i-1})$ is to the right of $\text{Center}(C_i)$ on $\text{Bisector}(p_i, q)$. Then the following remark is based on Ma's plane sweep algorithm [38] which produces the vertices of $\text{Bisector}(u, v)$.

Remark 1 ([38]). *Let $u, v \in \mathbb{R}^2$ be in general position with $y(v) > y(u)$. Then for any $C, C' \in \text{Boundary}(u, v)$ we have that $\angle \text{South}(C) \text{Center}(C) v \geq \angle \text{South}(C') \text{Center}(C') v$ provided that $\text{Center}(C)$ is to the left of $\text{Center}(C')$ on $\text{Bisector}(u, v)$.*

In the following lemma, we describe the structure of the path output by Algorithm 2.

Lemma 1. *Let $i \in \{1, \dots, k\}$. If edges $p_{i-1}p_i$ and $p_i p_{i+1}$ both use case (a) in Algorithm 2, then $\angle w_{i-1} \text{Center}(C_{i-1}) p_i \geq \angle w_i \text{Center}(C_i) p_i$. If instead both edges use case (b) in Algorithm 2, then $\angle p_i \text{Center}(C_{i-1}) w_{i-1} \geq \angle p_i \text{Center}(C_i) w_i$.*

Proof. Assume without loss of generality that edges $p_{i-1}p_i$ and $p_i p_{i+1}$ were chosen using case (a) in Algorithm 2. Then p_i is above st . We will first consider the case where T_{i-1} and T_i share an edge, denoted $p_i q$. Since p_i is above st , then q is below st . Refer to Figure 4.2. Both $\text{Center}(C_{i-1})$ and $\text{Center}(C_i)$ lie on $\text{Bisector}(q, p_i)$, however it remains to establish their relative position. Then by Remark 1, the result follows if we show that $\text{Center}(C_{i-1})$

Algorithm 2: Local Routing algorithm in $\odot_n$ -Delaunay triangulation

Data: Two points $s, t \in S$ (w.l.o.g. $y(s) = y(t)$ and $x(s) < x(t)$)

Result: Vertices $s = p_0, \dots, t = p_{k+1}$ forming a path in $\odot_n$ -Delaunay graph of S

Set $i \leftarrow 0$ and $p_i \leftarrow s$;

while $p_i \neq t$ **do**

(i) If t is a neighbour of p_i , set $p_{i+1} \leftarrow t$.

(ii) Otherwise, let T_i be the rightmost triangle in the graph intersecting st with vertex p_i . Let a, b be the other vertices of T_i above and below st , respectively. Let C_i be the empty n -gon with $p_i, a, b \in \partial C_i$. Let $w_i := \text{West}(C_i)$ and t_i be the intersection of C_i with st closest to t .

(a) If $p_i \in \text{Arc}(C_i, w_i, t_i)$, then set $p_{i+1} \leftarrow a$ and $i \leftarrow i + 1$.

(b) Else, set $p_{i+1} \leftarrow b$ and $i \leftarrow i + 1$.

end

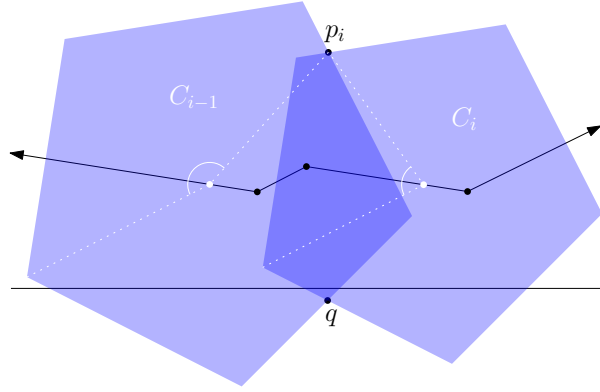


Fig. 4.2: The black polygonal chain is $\text{Bisector}(q, p_i)$. The white points are the centers of the 5-gons C_{i-1} and C_i .

is to the left of $\text{Center}(C_i)$. Define $L := \{\text{Center}(C) \mid C \in \text{Boundary}(p_i, q) \text{ and } p_{i-1} \in \text{int}(C)\}$ and $R := \{\text{Center}(C) \mid C \in \text{Boundary}(p_i, q) \text{ and } p_{i+1} \in \text{int}(C)\}$. Since $p_{i-1} \in \text{Arc}(C_{i-1}, q, p_i)$, then L propagates from $\text{Center}(C_{i-1})$ to the left. Similarly, R propagates from $\text{Center}(C_i)$ to the right since $p_{i+1} \in \text{Arc}(C_i, p_i, q)$. If $\text{Center}(C_{i-1})$ is to the right of $\text{Center}(C_i)$, then $L \cup R = \text{Bisector}(q, p_i)$. However, $\text{Center}(C_{i-1})$ and $\text{Center}(C_i)$ are both examples of points in $\text{Bisector}(q, p_i)$ but not in $L \cup R$. Therefore $\text{Center}(C_{i-1})$ is to the left of $\text{Center}(C_i)$, hence $\angle \text{South}(C_{i-1})\text{Center}(C_{i-1})p_i \geq \angle \text{South}(C_i)\text{Center}(C_i)p_i$. Then we obtain the desired inequality by remarking that p_i is not on $\text{Arc}(C_r, \text{South}(C_r), w_r)$ for $r \in \{i-1, i\}$ and also the angle $\angle \text{South}(C)\text{Center}(C)\text{West}(C)$ is constant for all homothets C of $\odot_n$.

If T_i and T_{i-1} do not share an edge, then we use this argument on all the triangles between T_{i-1} and T_i . The result follows since inequality is transitive. $\square$

Next, we define the worst-case n -gons, shown in Figure 4.3.

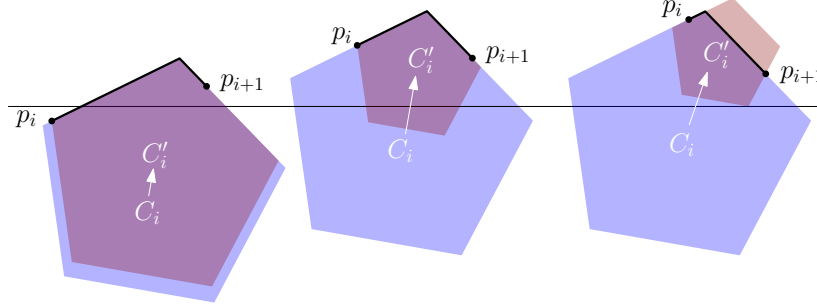


Fig. 4.3: The original 5-gons, C_i , are blue, and the worst-case 5-gons, C'_i from Definition 3, are purple. In all examples, p_{i+1} is above st . Left: $\text{West}(C'_i) = p_i$. Middle: $\text{West}(C'_i) = p_i$. Right: $\text{South}(C'_i), s, t$ are collinear.

Definition 3 (Worst-Case n -gons). Let $i \in \{0, \dots, k\}$ and suppose p_{i+1} is above st . Start with an n -gon $C = C_i$, then move $\text{Center}(C)$ left along $\text{Bisector}(p_i, p_{i+1})$ while keeping $C \in \text{Boundary}(p_i, p_{i+1})$ until the points $\text{South}(C), s, t$ are collinear, or $p_i = \text{West}(C)$. The resulting n -gon is denoted by C'_i . If instead p_{i+1} is below st , then move $\text{Center}(C)$ right along $\text{Bisector}(p_i, p_{i+1})$ while keeping $C \in \text{Boundary}(p_i, p_{i+1})$ until the points $\text{North}(C), s, t$ are collinear, or $p_i = \text{West}(C)$. The resulting n -gon is again denoted by C'_i .

To shorten notation, denote $w'_i := \text{West}(C'_i)$ for $i \in \{0, \dots, k\}$. A similar statement to that of Lemma 1 can be made about the worst-case n -gons:

Lemma 2. Let $i \in \{1, \dots, k\}$. If edges $p_{i-1}p_i$ and $p_i p_{i+1}$ both use case (a) in Algorithm 2, then $\angle w'_{i-1} \text{Center}(C'_{i-1}) p_i \geq \angle w'_i \text{Center}(C'_i) p_i$. If instead both edges use case (b) in Algorithm 2, then $\angle p_i \text{Center}(C'_{i-1}) w'_{i-1} \geq \angle p_i \text{Center}(C'_i) w'_i$.

Proof. Let $i \in \{1, \dots, k\}$ and assume edges $p_{i-1}p_i$ and $p_i p_{i+1}$ both use case (a) in Algorithm 2. By Lemma 1, we have $\angle w_{i-1} \text{Center}(C_{i-1}) p_i \geq \angle w_i \text{Center}(C_i) p_i$. Since p_i is above st , then the construction of Definition 3 guarantees that $\angle w_{i-1} \text{Center}(C_{i-1}) p_i \leq \angle w'_{i-1} \text{Center}(C'_{i-1}) p_i$. Similarly, p_{i+1} is above st , meaning that $\angle w'_i \text{Center}(C'_i) p_i \leq \angle w_i \text{Center}(C_i) p_i$. Combining inequalities yields the result. Proving the case when both edges use case (b) in Algorithm 2 is similar. $\square$

Using the same point set as in Figure 4.1, the trace of Algorithm 2 is shown in Figure 4.4 with worst-case n -gons.

Now we define the wedge W_p to be the area swept by stretching $\bigcirc_n$ with its west point on p . More precisely, $W_p := \{v \in \mathbb{R}^2 \mid \exists \text{ homothet } C \text{ of } \bigcirc_n \text{ such that } p = \text{West}(C) \text{ and } v \in C\}$.

Lemma 3. For $i \in \{1, \dots, k\}$, we have $w'_i \in W_{w'_{i-1}}$.

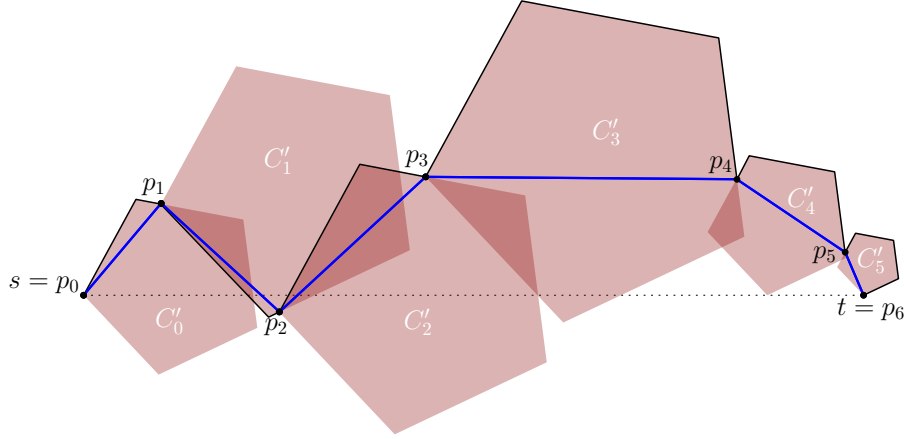


Fig. 4.4: Trace of Algorithm 2 with worst-case 5-gons.

Proof. Suppose p_i is above st . If $p_i = w'_i$, then w'_i is on the boundary of C'_{i-1} , hence $w'_i \in C_{w'_{i-1}}$. If on the other hand $p_i \neq w'_i$, by Definition 3 we must have $p_i \in \text{Arc}(C'_i, w'_i, \text{South}(C'_i))$ and $\text{South}(C'_i)$ is on st . Suppose for now that $\text{South}(C'_{i-1})$ is also on st . Define the set of homothets

$$L := \{C \mid \text{South}(C), s, t \text{ are collinear and } p_i \in \text{Arc}(C, \text{West}(C), \text{South}(C))\}.$$

Then we will prove the following claim, illustrated in Figure 4.5.

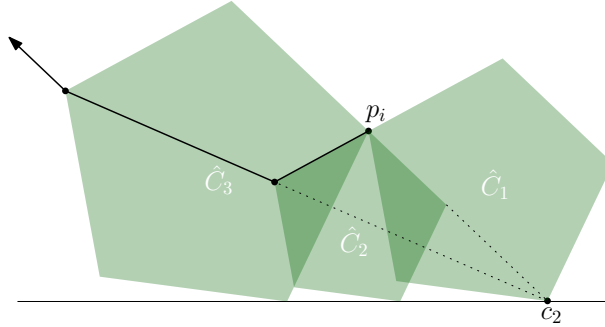


Fig. 4.5: Illustration of Claim 1. $\text{West}(L)$ is the black line, and c_2 is the homothety center relating $\hat{C}_2$ and $\hat{C}_3$. Segments $\text{West}(\hat{C}_2)\text{West}(\hat{C}_3)$ and $c_2\text{West}(\hat{C}_2)$ lie on the same line. In this example, $\sigma = 4$ and $m = 2$.

Claim 1 ($\text{West}(L)$ is a polygonal chain connected to a ray). *Firstly, suppose $1 < \sigma < n$ such that $C^\sigma = \text{South}(C)$. Then for $j \in \{1, \dots, \sigma - 1\}$, define $\hat{C}_j$ to be the unique homothet of $\odot_n$ where $p_i = \hat{C}_j^j$ and $\hat{C}_j^\sigma$ is collinear with st . By definition, for $j \in \{1, \dots, \sigma - 1\}$, we have $\text{West}(\hat{C}_j) \in \text{West}(L)$.*

Next, for $j \in \{1, \dots, \sigma - 2\}$, the n -gons $\hat{C}_j$ and $\hat{C}_{j+1}$ are related by a homothety whose center c_j lies on the intersection of lines given by extending the segments st and $\hat{C}_j^j \hat{C}_j^{j+1}$. Furthermore, for any $C \in L$ with p_i on $C^j C^{j+1}$, the homothety relating $\hat{C}_j$ and C has the same center, c_j . Therefore

the point $\text{West}(C)$ lies on the segment $\text{West}(\hat{C}_j)\text{West}(\hat{C}_{j+1})$. On the other hand, for $C \in L$ with p_i on $C^{\sigma-1}C^\sigma$, the n -gons C and $\hat{C}_{\sigma-1}$ are related by a homothety whose center is $\text{South}(\hat{C}_{\sigma-1})$. Therefore the point $\text{West}(C)$ must lie on the ray r starting at $\text{West}(\hat{C}_{\sigma-1})$ extending directly opposite $\text{South}(\hat{C}_{\sigma-1})$. We have shown that $\text{West}(L)$ is the polygonal chain $(\text{West}(\hat{C}_1), \dots, \text{West}(\hat{C}_{\sigma-1}))$ along with the ray r , so the proof of the claim is completed.

To establish a direction on $\text{West}(L)$, which is homeomorphic to a ray by the claim, we choose the convention that p_i is the rightmost point. Then since the vertices C^j are labelled in the clockwise orientation in the claim, we have that for $C, C' \in L$, if $\text{West}(C)$ is to the left of $\text{West}(C')$ on $\text{West}(L)$ then $\angle \text{West}(C)\text{Center}(C)p_i \geq \angle \text{West}(C')\text{Center}(C')p_i$. Therefore by Lemma 1, w'_i must be to the right of w'_{i-1} on $\text{West}(L)$.

Finally, we will show that the slope of $\text{West}(L)$ remains between the slope of $\bigcirc_n^1 \bigcirc_n^n$ and $\bigcirc_n^1 \bigcirc_n^2$. Recall the notation $\bigcirc_n^j$ denotes the j -th vertex clockwise around $\bigcirc_n$ where $\bigcirc_n^1 := \text{West}(\bigcirc_n)$. Let $1 < m < n$ such that $\bigcirc_n^m = \text{North}(\bigcirc_n)$. We analyze $\text{West}(L)$ in three portions.

Segment $\text{West}(\hat{C}_j)\text{West}(\hat{C}_{j+1})$ for $j \in \{1, \dots, m-1\}$ Let $j \in \{1, \dots, m-1\}$. Then by the homothety relating $\hat{C}_j$ and $\hat{C}_{j+1}$, the segment $\text{West}(\hat{C}_j)\text{West}(\hat{C}_{j+1})$ has slope equal to the slope of segment $c_j\text{West}(\hat{C}_j)$. Since c_j lies to the left of the line by extending $\text{West}(\hat{C}_j)\hat{C}_j^2$, then the slope of $c_j\text{West}(\hat{C}_j)$ is positive and less than the slope of $\bigcirc_n^1 \bigcirc_n^2$.

Segment $\text{West}(\hat{C}_j)\text{West}(\hat{C}_{j+1})$ for $j \in \{m, \dots, \sigma-2\}$ Suppose $j \in \{m, \dots, \sigma-2\}$. Then by the homothety relating $\hat{C}_j$ and $\hat{C}_{j+1}$, the segment $\text{West}(\hat{C}_j)\text{West}(\hat{C}_{j+1})$ has slope equal to the slope of segment $\text{West}(\hat{C}_j)c_j$. Since c_j lies to the right of $\text{South}(\hat{C}_j)$, then the slope of $\text{West}(\hat{C}_j)c_j$ is negative and greater than or equal to the slope of $\bigcirc_n^1 \bigcirc_n^\sigma$, which is always at least the slope of $\bigcirc_n^1 \bigcirc_n^n$.

Ray r The slope of ray r is $\text{West}(\hat{C}_{\sigma-1})c_{\sigma-1}$. Since $c_{\sigma-1}$ is $\text{South}(\hat{C}_{\sigma-1})$, then the slope of the ray is equal to the slope of $\bigcirc_n^1 \bigcirc_n^\sigma$, which is again in the desired range.

Therefore the slope of each segment of $\text{West}(L)$ is within the range given by the cone $W_{w'_{i-1}}$. Since w'_i is to the right on $\text{West}(L)$, then we must have $w'_i \in W_{w'_{i-1}}$.

If now $\text{South}(C'_{i-1})$ is instead below st , then we define the homothet C' such that $p_i \in \partial C'$, the points $s, \text{South}(C'), t$ are collinear, and $\text{Center}(C'_{i-1}), \text{Center}(C'), p_i$ are also collinear. Since C' is fully contained in C'_{i-1} , then $\text{West}(C') \in W_{w'_{i-1}}$. Also, $\angle \text{West}(C')\text{Center}(C')p_i = \angle \text{West}(C'_{i-1})\text{Center}(C'_{i-1})p_i$, therefore the argument from above can show that $w'_i \in W_{\text{West}(C')}$, hence $w'_i \in W_{w'_{i-1}}$. $\square$

Next we define a projection that depends on whether the point is above or below st .

Definition 4 (West Side Projection). For a point p above st , let $\bar{p}$ be the intersection of st with the line passing through p with the same slope as $\bigcirc_n^1 \bigcirc_n^2$. Similarly, when p is below st , let $\bar{p}$ be the intersection of st with the line passing through p with the same slope as $\bigcirc_n^1 \bigcirc_n^n$.

Using the projections, we define the two paths that we will call *snail paths* and that are used to bound the length of the path output by Algorithm 2.

Definition 5 (Snail Paths). Let $a, b \in \mathbb{R}^2$ satisfy $y(a) = y(b)$. When $x(a) < x(b)$, define the n -gon C such that $b = \text{South}(C)$ and a, C^1, C^2 are collinear. Then $\nearrow_{a,b} := aC^1 + \text{Arc}(C, C^1, b)$. Similarly, let C' be the n -gon such that $b = \text{North}(C')$ and a, C'^1, C'^n are collinear, and then $\searrow_{a,b} := aC'^1 + \text{Arc}(C', b, C'^1)$. When $x(a) \geq x(b)$, we define $\nearrow_{a,b}$ and $\searrow_{a,b}$ to be empty paths.

The shape of the snail path is very important as it will directly lead to the routing ratio in the proof of Theorem 2. Notice how the path in Figure 4.9 is arbitrarily close to $\nearrow_{s,t}$.

Remark 2. After fixing the orientation of the $\bigcirc_n$, there exists a real number $c > 1$ such that for any $a, b \in \mathbb{R}^2$ with $y(a) = y(b)$ and $x(a) < x(b)$, we have $|\nearrow_{a,b}| = c|\overline{ab}|$. When $x(a) \geq x(b)$, then $|\nearrow_{a,b}| = 0$. The same is true for $|\searrow_{a,b}|$.

When $n = 5$, we prove $\Psi(5)$ is an upper bound on the constant c from Remark 2.

Lemma 4. Let $n = 5$ and $a, b \in \mathbb{R}^2$ with $y(a) = y(b)$ and $x(a) < x(b)$. Then $\max(|\nearrow_{a,b}|, |\searrow_{a,b}|) \leq \Psi(5)|ab|$.

Proof. Let C be the 5-gon corresponding to $\nearrow_{a,b}$, and let $\theta := \angle \text{West}(C)ab$. For now, assume $C^5 = \text{South}(C)$, meaning $\pi/5 \leq \theta$, and also $\theta \leq \pi/2$. Then the side length $|\text{West}(C)\text{South}(C)| = \frac{|ab|\sin(\theta)}{\sin(2\pi/5)}$ by the law of sines. Similarly, $|a\text{West}(C)| = \frac{|ab|\sin(2\pi/5+\theta)}{\sin(2\pi/5)}$. Since $|\nearrow_{a,b}| = |a\text{West}(C)| + 4|\text{West}(C)\text{South}(C)|$, then we have

$$|\nearrow_{a,b}| = \frac{|ab|\sin(2\pi/5 + \theta)}{\sin(2\pi/5)} + 4\frac{|ab|\sin(\theta)}{\sin(2\pi/5)} \leq \Psi(5)|ab|,$$

where the last inequality follows from the analysis in the appendix. See Lemma 7. It is straightforward to verify that the claim still holds when $C^5 \neq \text{South}(C)$. The analysis for $\searrow_{a,b}$ is symmetric. $\square$

One useful tool that we will often use is a convex path bound from [6].

Observation 2 (Convex Path Bound). Suppose two convex paths $\mathcal{P}_1, \mathcal{P}_2$ have the same endpoints a and b , and $\mathcal{P}_1$ is contained in the region formed by the simple polygon $\mathcal{P}_2 + ab$. Then $|\mathcal{P}_1| \leq |\mathcal{P}_2|$.

Next, for $i \in \{0, \dots, k\}$, we define the path $P_i := \overline{w'_{i-1}w'_i} + \text{Arc}(C'_i, w'_i, p_i)$. Paths of this form will be used in the following lemma for pentagons ($n = 5$).

Lemma 5. Let $n = 5$. For $1 \leq i \leq k$, we have

$$|P_{i-1}| + |p_{i-1}p_i| \leq \Psi(5)|\overline{w'_{i-1}w'_i}| + |P_i|. \quad (4.1)$$

Furthermore, we have $|P_k| + |p_k t| \leq \Psi(5)|\overline{w'_k t}|$.

Proof. For $i \in \{1, \dots, k\}$, we will show that (4.1) holds using a case analysis. Without loss of generality, we will assume that p_i is above st for all cases. This is equivalent to assuming that the routing decision is case (a) in Algorithm 2, meaning that $p_{i-1} \in \text{Arc}(C'_{i-1}, w'_{i-1}, p_i)$. The arguments for when p_i is below st are symmetric. We will use the following shorthand: $s'_i := \text{South}(C'_i)$.

Case 1: p_{i-1} is above st and both s'_{i-1}, s'_i lie on st See Figure 4.6. We split the snail path of $\overline{C'_{i-1}}$ up into several parts:

$$|\nearrow_{\overline{w'_{i-1}, s'_{i-1}}}^{\overline{C'_{i-1}}}| = |P_{i-1}| + |\text{Arc}(C'_{i-1}, p_{i-1}, p_i)| + |\text{Arc}(C'_{i-1}, p_i, s'_{i-1})|. \quad (4.2)$$

If $\overline{w'_i}$ is west of s'_{i-1} , then by a convex path bound, we get

$$|\nearrow_{\overline{w'_i, s'_{i-1}}}^{\overline{C'_{i-1}}}| \leq |P_i| + |\text{Arc}(C'_{i-1}, p_i, s'_{i-1})|. \quad (4.3)$$

On the other hand, inequality 4.3 still holds when $\overline{w'_i}$ is east of s'_{i-1} since $|\nearrow_{\overline{w'_i, s'_{i-1}}}^{\overline{C'_{i-1}}}| = 0$. Finally,

$$\begin{aligned} |P_{i-1}| + |\text{Arc}(C'_{i-1}, p_{i-1}, p_i)| &= |\nearrow_{\overline{w'_{i-1}, s'_{i-1}}}^{\overline{C'_{i-1}}}| - |\text{Arc}(C'_{i-1}, p_i, s'_{i-1})| - |P_i| + |P_i| && \text{by 4.2} \\ &\leq |\nearrow_{\overline{w'_{i-1}, s'_{i-1}}}^{\overline{C'_{i-1}}}| - |\nearrow_{\overline{w'_i, s'_{i-1}}}^{\overline{C'_{i-1}}}| + |P_i| && \text{by 4.3} \\ &\leq |\nearrow_{\overline{w'_{i-1}, w'_i}}^{\overline{C'_{i-1}}}| + |P_i| && \text{by Remark 2} \end{aligned}$$

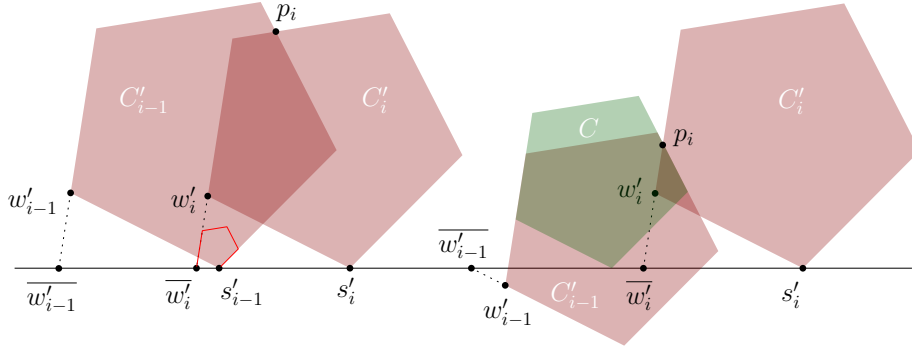


Fig. 4.6: Left: Case 1 when $x(\overline{w'_i}) < x(s'_{i-1})$. The path $\nearrow_{\overline{w'_i, s'_{i-1}}}^{\overline{C'_{i-1}}}$ is red. Right: Case 2 reduces to case 1 by defining n -gon C in green.

Case 2: p_{i-1} is above st and only s'_i lies on st We will reduce this case to Case 1. See Figure 4.6 First, define a new n -gon C with $w := \text{West}(C)$ such that $\overline{w} = \overline{w'_{i-1}}$, and $\text{South}(C), s, t$ are collinear, and $p_i \in \partial C$. By a convex path bound, we have

$$|P_{i-1}| + |\text{Arc}(C'_{i-1}, p_{i-1}, p_i)| \leq |\overline{w'_{i-1} w}| + |\text{Arc}(C, w, p_i)| \quad (4.4)$$

Lastly, we proceed with the argument in Case 1, replacing C'_{i-1} with C in order to obtain

$$|\overline{w'_{i-1} w}| + |\text{Arc}(C, w, p_i)| \leq |P_i| + |\nearrow_{\overline{w'_{i-1}, w'_i}}^{\overline{C'_{i-1}}}| \quad (4.5)$$

Combining (4.4) with (4.5) yields (4.1).

Case 3: p_{i-1} is above st and s'_i lies below st In this case, $p_i = w'_i$, meaning $P_i = \overline{w'_i p_i}$. Let point p be collinear with $\overline{w'_{i-1}}, w'_{i-1}$ such that $y(p) = y(p_i)$. Then by a convex path bound, we have

$$|P_{i-1}| + |\text{Arc}(C'_{i-1}, p_{i-1}, p_i)| \leq |\overline{w'_{i-1} p}| + |\bigwedge_{p, p_i}|$$

Finally, equation (4.1) follows since the paths $\overline{w'_{i-1} p}$ and $\bigwedge_{p, p_i}$ are translates of P_i and $\bigwedge_{\overline{w'_{i-1}}, \overline{w'_i}}$ respectively.

Case 4: p_{i-1} is below st In this case, $w'_{i-1} = p_{i-1}$, hence $P_{i-1} = \overline{w'_{i-1} w'_i}$. Define point p

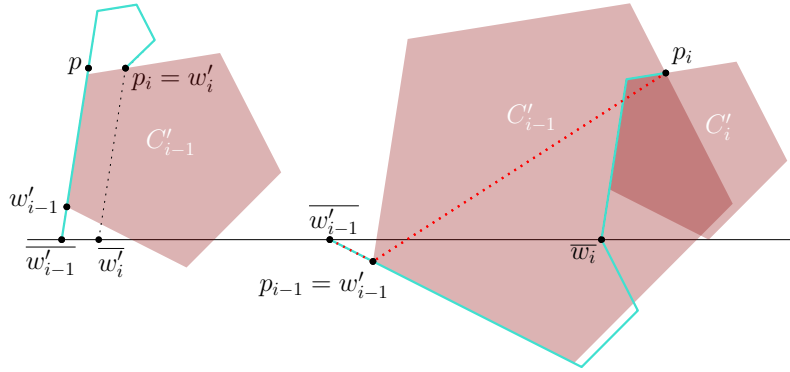


Fig. 4.7: Left: In Case 3, the blue path $\overline{w'_{i-1} X} + \bigwedge_{X, p_i}$ is longer than the dotted red path $P_{i-1} + \text{Arc}(C'_{i-1}, p_{i-1}, p_i)$. Right: In Case 4: The blue path $P_{i-1} + p_{i-1} p_i$ is longer than the dotted red path $\bigwedge_{\overline{w'_{i-1}}, \overline{w'_i}} + P_i$.

be the intersection of C'_{i-1} closest to s . We will prove the claim, illustrated in Figure 4.8.

Claim 2 (P_{i-1} is a sub-path of $\bigwedge_{\overline{w'_{i-1}}, p}$). Fix m such that $\bigcirc_n^m := \text{North}(\bigcirc_n)$. Then for $0 \leq j \leq m-2$ let $\hat{C}_j$ be the n -gon such that $\hat{C}_j^{m-j} = p$ and $\text{West}(\hat{C}_j) = \overline{w'_{i-1}}$. Note that $\hat{C}_0$ is exactly the n -gon corresponding to $\bigwedge_{\overline{w'_{i-1}}, p}$. In addition, p was defined to be collinear with $C_{i-1}^1 C_{i-1}^2$, therefore $w'_{i-1} = \text{West}(\hat{C}_{m-2})$. Then, for $1 \leq j \leq m-2$, the n -gons $\hat{C}_{j-1}, \hat{C}_j$ are related by a homothety with center c_j lying on the intersection of extended segments $\hat{C}_j^1 \hat{C}_j^n$ and $\hat{C}_j^{m-j} \hat{C}_j^{m-j+1}$. By construction, we have $\hat{C}_j^{m-j} = \hat{C}_{j-1}^{m-j+1}$, therefore by symmetry (reflection about the extended line $c_j \text{Center}(\hat{C}_j)$), we also have $\hat{C}_j^1 = \hat{C}_{j-1}^n$. Then $\text{West}(\hat{C}_j)$ is on the snail path $\bigwedge_{\overline{w'_{i-1}}, p}$ if and only if $j < 2$. In particular, the claim holds when $m-2 < 2$, which is the case for $n \in \{5, 6, 7, 8\}$ since in general $\lfloor \frac{n}{4} \rfloor \leq m-1 \leq \lceil \frac{n}{4} \rceil$.

Then by Lemma 3, $x(p) \leq x(\overline{w'_i})$, therefore P_{i-1} is also sub-path of $\bigwedge_{\overline{w'_{i-1}}, \overline{w'_i}}$. Inequality (4.1) follows since the paths $P_{i-1} + p_{i-1} p_i$ and $\bigwedge_{\overline{w'_{i-1}}, \overline{w'_i}} + P_i$ have the same endpoints, concluding this case.

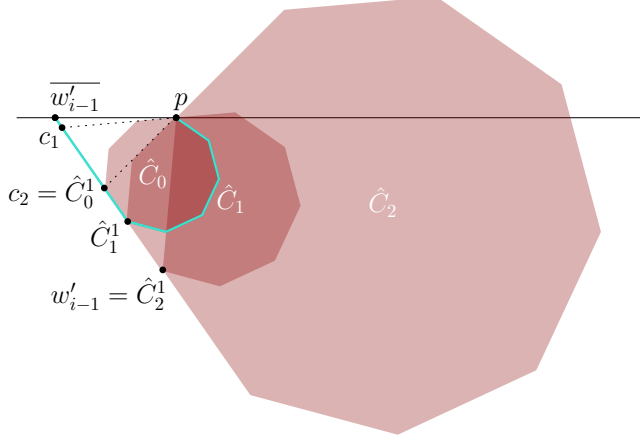


Fig. 4.8: Case 4, Claim 2. Here, $n = 9$, $m = 4$, and therefore $C'_{i-1} = \hat{C}_{m-2}^1$ is not on the blue path $\searrow_{\overline{w'_{i-1}, p}}$. If instead $n \in \{5, 6, 7, 8\}$, then $m \in \{2, 3\}$.

Finally, we have $|P_k| + |p_k t| \leq \max(|\searrow_{\overline{w'_k, t}}|, |\searrow_{\overline{w'_k, t}}|)$ by a convex path bound. Note that for $i = 1$, the edge sp_1 classifies as Case 4. $\square$

Putting this all together, we can now prove Theorem 2.

Proof. Consider the path $s = p_0, \dots, p_{k+1} = t$ from Algorithm 2 and apply Lemma 5:

$$\begin{aligned} \sum_{i=1}^{k+1} |p_{i-1} p_i| &\leq \left(\sum_{i=1}^k |P_i| + \Psi(5) |\overline{w'_{i-1} w'_i}| - |P_{i-1}| \right) + (\Psi(5) |\overline{w'_k t}| - |P_k|) \\ &= \Psi(5) |\overline{w'_0 t}| - |P_0| = \Psi(5) |st|. \end{aligned}$$

Since Algorithm 2 is a $O(\log(|V(G)|))$ -memory local routing algorithm, then $\Psi(5)$ is an upper bound on the routing ratio of $\bigcirc_5$ -Delaunay graphs. $\square$

Now we present a lemma showing that our analysis is optimal for Algorithm 2.

Lemma 6. *Let $n \in \{5, 6, 7, 8\}$. For any $\epsilon > 0$, there is a $\bigcirc_n$ -Delaunay graph for which Algorithm 2 has a routing ratio of at least $\Psi(n) - \epsilon$.*

Proof. For simplicity, we present S not in general position, however it is possible to perturb the position of each vertex to force the $\bigcirc_n$ -Delaunay graph of S to have the desired edges while being in general position. See Figure 4.9. Let C be an n -gon, and define σ such that $C^\sigma = \text{South}(C)$. Then let L be a horizontal line above and arbitrarily close to C^σ . Define $t := L \cap C^{\sigma-1} C^\sigma$ and $v_j := C^j$ for $j \in \{2, \dots, \sigma-1\}$. Let v_0 be on $C^\sigma C^{\sigma+1}$ below L . Let v_1 be on $C^1 C^2$ arbitrarily close to C^1 . Then let $\hat{C}$ be an n -gon such that $\text{East}(\hat{C}) = v_0$ and $p_1 \in \partial \hat{C}$. Finally, s is the leftmost intersection of L and $\partial \hat{C}$. Let $S := \{s, t, v_0, v_1, \dots, v_{\sigma-1}\}$. While there are several valid $\bigcirc_n$ -Delaunay graphs of S , we will choose to route on the graph with edges $sv_0, sv_1, v_{j-1}v_j, v_1v_0, v_jv_0, v_0t, v_{\sigma-1}t$ for $j \in \{2, \dots, \sigma-1\}$. When Algorithm 2 routes from s to t

in the $\odot_n$ -Delaunay graph of S , then each step of case (ii) is sub-case (a), therefore the path is $s, v_1, v_2, \dots, v_{\sigma-1}, t$. By construction, this path is arbitrarily close to the snail path $\bigwedge_{s,t}$. Since $\Psi(n)|st|$ represents the maximum length of the snail path $\bigwedge_{s,t}$ over all orientations of $\odot_n$, then the routing ratio of Algorithm 2 for this graph is at least $\Psi(n) - \epsilon$. $\square$

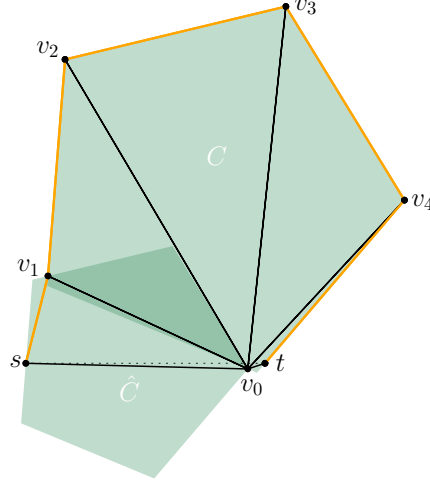


Fig. 4.9: Worst-case point-set construction causes Algorithm 2 to choose the orange path which is arbitrarily close to $|\bigwedge_{s,t}|$ since each step of case (ii) is sub-case (a). Here $n = \sigma = 5$. For more details, see proof of Lemma 6.

4.4.2 Extending Algorithm 2 to graphs where the convex hull is not present

In order to extend Algorithm 2 to graphs where the convex hull is not present, we add dummy vertices to the graph G to ensure the entire set S is triangulated. In particular, define the set D of dummy vertices as follows:

Let C be the scaled translate of $\odot_n$ with $\text{Center}(C) = s$ and $|\text{West}(C)\text{Center}(C)| = 10|st|$. Then let $\text{Rot}(C)$ denote the rotated n -gon obtained by rotating C about $\text{Center}(C)$ by π radians (or by $\pi - \epsilon$ radians for $\epsilon > 0$ to satisfy the general position assumption). Lastly, let D be the corners of $\text{Rot}(C)$ that are in the unbounded region of the $\odot_n$ -Delaunay graph of S .

If Algorithm 2 is used in the $\odot_n$ -Delaunay graph of $S \cup D$, then the path from s to t will have length at most $c|st|$ where c is given in Lemma 7. In all cases, $c < 10$, so for all $i \in \{0, \dots, k+1\}$, we have that $p_i \notin D$. The choice of D ensures that all p_i are not incident to the unbounded region.

When routing in the $\odot_n$ -Delaunay graph of S , Algorithm 2 can be modified to simulate the edges of the $\odot_n$ -Delaunay graph of $S \cup D$. Firstly, it can be verified locally whether the current vertex p_i is incident to the unbounded region. Indeed, p_i is in the unbounded region if and only if there exists a $j \in \{1, \dots, n\}$ and an n -gon C such that $C^j = p_i$ and no edges incident to p_i intersect C . Note that the size of C is irrelevant. Then if p_i is incident with the unbounded region, it can be verified whether p_i has neighbours in D in the $\odot_n$ -Delaunay graph of $S \cup D$ since the algorithm stores the location of s and can calculate the distance $|st|$.

4.4.3 Extending our result to hexagons, septagons and octagons

So far, we have shown that Algorithm 2 has a routing ratio of at most $\Psi(5) \approx 4.640$ for any $\bigcirc_5$ -Delaunay graph. It turns out that a nearly identical argument can be used for $n \in \{6, 7, 8\}$, however a counter-intuitive behaviour happens at $n = 9$; it turns out that the length of the snail path for $n = 9$ is not an upper bound on the routing ratio. This interesting behaviour appears for the circle in [6]. See Figure 4.8 for an example of how Claim 2 does not hold for $n \geq 9$.

To extend our result to n -gons with $6 \leq n \leq 8$, then Lemma 4 needs to be generalized to Lemma 7. Additionally, Lemma 5 is trivially generalized by replacing each occurrence of $\Psi(5)$ with $\Psi(n)$ for the corresponding n -gon.

Lemma 7. *Let $a, b \in \mathbb{R}^2$ with $y(a) = y(b)$ and $x(a) < x(b)$. Then $\max(|\nearrow_{a,b}|, |\searrow_{a,b}|) \leq \Psi(n)|ab|$ for $n \in \{5, 6, 7, 8\}$.*

Proof. Let $n \in \{5, 6, 7\}$ and let C be the n -gon corresponding to $\nearrow_{a,b}$. Let $\theta := \angle \text{West}(C)ab$, and for now assume that $\text{South}(C) = C^n$. This means that $\pi/5 \leq \theta \leq \pi/2$. Then by the law of sines, $|\text{West}(C)\text{South}(C)| = \frac{|ab|\sin(\theta)}{\sin(2\pi/n)}$ and $|a\text{West}(C)| = \frac{|ab|\sin(2\pi/n+\theta)}{\sin(2\pi/n)}$. Since $|\nearrow_{a,b}| = |a\text{West}(C)| + (n-1)|\text{West}(C)\text{South}(C)|$, then we have

$$\frac{|\nearrow_{a,b}|}{|ab|} = \frac{\sin(2\pi/n + \theta)}{\sin(2\pi/n)} + (n-1) \frac{\sin(\theta)}{\sin(2\pi/n)}. \quad (4.6)$$

Focusing on the numerators, we have

$$\begin{aligned} \frac{d}{d\theta} \left(\sin(2\pi/n + \theta) + (n-1)\sin(\theta) \right) &= \cos(2\pi/n + \theta) + (n-1)\cos(\theta) \\ &= (\cos(2\pi/n)\cos(\theta) - \sin(2\pi/n)\sin(\theta)) + (n-1)\cos(\theta) \\ &= (\cos(2\pi/n) + n-1)\cos(\theta) - \sin(2\pi/n)\sin(\theta). \end{aligned}$$

From there, we get that the critical value of θ is

$$\theta^* = \arctan \left(\frac{\cos(2\pi/n) + n-1}{\sin(2\pi/n)} \right).$$

Therefore the maximum value is

$$\frac{\sin(2\pi/n + \theta^*) + (n-1)\sin(\theta^*)}{\sin(2\pi/n)}.$$

When n is 5, 6 or 7, the maximum value of (4.6) is approximately 4.640, 6.429, or 8.531 respectively. It is straightforward to verify that the claim still holds when $\text{South}(C) \neq C^n$.

When $n = 8$, the analysis is slightly different since we can not have $\text{South}(C) = C^n$. As before, let C be the 8-gon corresponding to $\nearrow_{a,b}$, and let $\theta := \angle \text{West}(C)ab$. Then necessarily, we have $\text{South}(C) = C^7$, meaning that $\angle bC^1a = \frac{3\pi}{8}$. By the law of sines, $|C^1C^7| = \frac{|ab|\sin(\theta)}{\sin(3\pi/8)}$

and $|aC^1| = \frac{|ab|\sin(3\pi/8+\theta)}{\sin(3\pi/8)}$. Also, $|\text{Arc}(C, C^7, C^1)|\cos(\pi/8) = |C^1C^7|$. Since $|\nearrow_{a,b}| = |aC^1| + 3|\text{Arc}(C, C^7, C^1)|$, then we have

$$\frac{|\nearrow_{a,b}|}{|ab|} = \frac{\sin(3\pi/8+\theta)}{\sin(3\pi/8)} + 3\frac{\sin(\theta)}{\sin(3\pi/8)\cos(\pi/8)} \text{ where } \pi/4 \leq \theta \leq \pi/2,$$

which reaches a maximum of around 4.054 by a similar analysis to above. $\square$

4.5 Routing Ratio Lower Bound for $\bigcirc_5$ -Delaunay graphs

Lemma 8. *Every local routing algorithm for $\bigcirc_5$ -Delaunay graphs must have a routing ratio of at least $1.995 - \epsilon$ for any $\epsilon > 0$.*

Proof. Consider the construction shown in Figure 4.10. Let $\delta > 0$ and let C be a pentagon

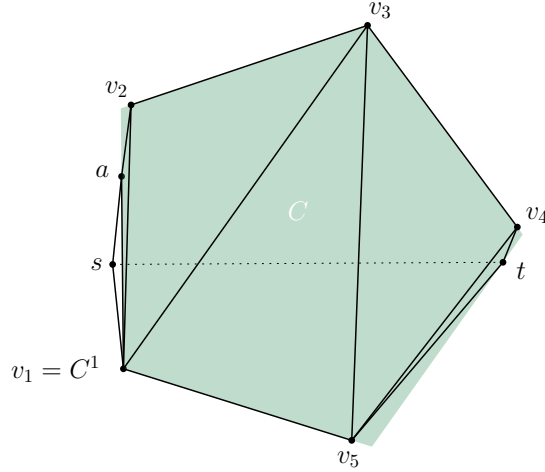


Fig. 4.10: Point set construction obtaining a routing ratio lower bound of approximately 1.995.

with $|C^1C^2| = 1 + \delta$. Define $v_1 := C^1$, $v_2 \in C^2C^3$ arbitrarily close to C^2 , $v_3 := C^3$, $v_4 \in C^3C^4$ arbitrarily close to C^4 , and $v_5 \in C^5C^1$ arbitrarily close to C^5 . Finally place point a on C^1C^2 such that $|C^1a| = 1$. Place s outside C equidistant from v_1, a , arbitrarily close to C^1C^2 . Define the point $t \in C^4C^5$ on the line perpendicular to C^1C^2 through s . The point set is $S := \{s, t, a, v_1, v_2, v_3, v_4, v_5\}$. While S admits many different triangulations, we choose the triangulation from Figure 4.10. The edges are $sa, sv_1, av_1, av_2, v_1v_2, v_1v_3, v_1v_5, v_2v_3, v_3v_4, v_3v_5, v_4v_5, v_4t, v_5t$. In particular, the neighbourhood of s is $\{a, v_1\}$, however the shortest path from a to t is approximately along the boundary ∂C . We will analyze the routing ratio of any algorithm that chooses to go to a . Then, any algorithm that chooses v_1 first will perform poorly on a symmetric graph reflected about the line through st .

We consider the clockwise and counterclockwise arcs from point a to point t . The

counterclockwise arc has length

$$\begin{aligned} |\text{Arc}(C_1, t, a)| &= |av_1| + |v_1C^5| + |C^5t| \\ &= 1 + (1 + \delta) + \frac{\frac{1}{2} + (1 + \delta)\sin 18}{\sin 54} \end{aligned}$$

On the other hand, the clockwise arc from a to t has length

$$\begin{aligned} |\text{Arc}(C_1, a, t)| &= |aC^2| + |C^2C^3| + |C^3C^4| + |C^4t| \\ &= \delta + (1 + \delta) + (1 + \delta) + (1 + \delta - \frac{\frac{1}{2} + (1 + \delta)\sin 18}{\sin 54}) \end{aligned}$$

Finally, we have $|st| = |v_1C^5|\cos 18 + |C^5t|\cos 54 = (1 + \delta)\cos 18 + \frac{\frac{1}{2} + (1 + \delta)\sin 18}{\tan 54}$. This means that the routing ratio is at least

$$\frac{|sa| + |\text{ShortestPath}(a, t)|}{|st|} = \frac{\frac{1}{2} + \min(|\text{Arc}(C_1, a, t)|, |\text{Arc}(C_1, t, a)|)}{|st|}$$

which reaches approximately 1.995 when $\delta = 0.447$. $\square$

4.6 Spanning Ratio Lower Bound for $\bigcirc_5$ -Delaunay graphs

Lemma 9. *For any $\epsilon > 0$, there exists a $\bigcirc_5$ -Delaunay graph with spanning ratio of at least $\frac{5}{2+3\sin(\frac{\pi}{10})} - \epsilon$.*

Proof. Let $\epsilon > 0$. We will construct a point set for the $\bigcirc_5$ -Delaunay graph shown in Figure 4.11. In particular, let the pentagon C have side length 1, then place a on C^1C^5 with $|C^1a| = \frac{1}{4}$ and b on C^3C^4 with $|C^3b| = \frac{1}{4}$. Next, place $v_1 \in C^1C^2$ arbitrarily close to C^1 , $v_2 := C^2$, $v_3 \in C^2C^3$ arbitrarily close to C^3 , $v_4 \in C^4C^5$ arbitrarily close to C^4 , and $v_5 \in C^4C^5$ arbitrarily close to C^5 . We define $S := \{a, b, v_1, v_2, v_3, v_4, v_5\}$ and consider the triangulation with edges $av_1, av_5, v_1v_2, v_1v_5, v_2v_3, v_2v_4, v_2v_5, v_3v_4, v_3b, v_4b, v_4v_5$. The shortest path from a to b is approximately the boundary ∂C . Therefore the length of the shortest path from a to b is approximately $\frac{5}{2}$, whereas $|ab| = 1 + \frac{3}{2}\sin(\frac{\pi}{10})$. This means that the spanning ratio of is approximately 1.708. $\square$

4.7 Conclusions

We have dramatically improved the upper bounds on the spanning and routing ratios for $\bigcirc_n$ -Delaunay graphs for $n \in \{5, 6, 7, 8\}$. We also provided matching lower bounds for the paths output by our routing algorithm, showing that our analysis is tight. Furthermore, we prove that no routing algorithm for pentagon-Delaunay graphs can have a routing ratio less than 1.995. Finally, we show that the worst-case spanning ratio is at least 1.708 for the pentagon-Delaunay graph. We conclude with the following three open questions: (1) Can we improve the lower bound on the routing ratio for these graphs? (2) Can we provide a routing algorithm that attains this lower bound? (3) Can our approach be generalized for regular polygons with more than 8 sides?

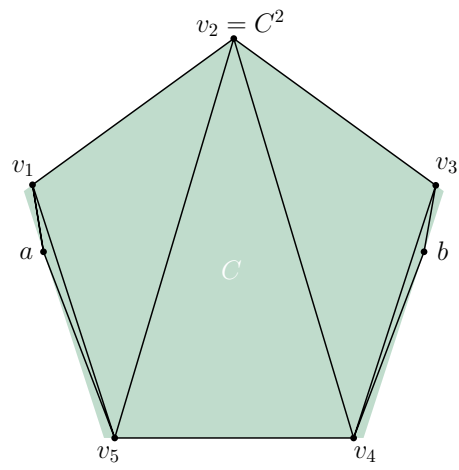


Fig. 4.11: Point set construction obtaining a spanning ratio lower bound of approximately 1.708. The shortest path between a and b is arbitrarily close to the perimeter of the pentagon.

Chapter 5

Routing in Sparse Directed Planar Graphs: The $\vec{\text{Yao}}_4^\infty$ -Graph

Prosenjit Bose, Jean-Lou De Carufel, John Stuart

5.1 Abstract

Local routing is the process of finding short paths in a graph without global knowledge. Fewer edges makes the task of local routing even more difficult. Previously, all memoryless local routing algorithms for directed plane graphs required there to be roughly $6|V|$ directed edges in the graph, where V is the vertex set. We introduce a memoryless local routing algorithm for a graph, referred to as the $\vec{\text{Yao}}_4^\infty$ -graph, which contains only $4|V|$ directed edges. Given a finite vertex set consisting of points in the plane, a $\vec{\text{Yao}}_4^\infty$ -graph can be built as follows. Consider a horizontal and a vertical line through each vertex v , defining four quadrants around v . Then v has an outgoing edge to the *closest* vertex in each of its four quadrants. When the distance is measured using the Euclidean norm, the resulting graph is the $\vec{\text{Yao}}_4$ -graph, whereas with the L_∞ -norm, we obtain the $\vec{\text{Yao}}_4^\infty$ -graph, which is a sub-graph of the well-known L_∞ -Delaunay graph. In general, we show that there are vertex sets which require at least $2|V| - 2$ directed edges in order to construct a spanner. Since the spanning ratio is a lower bound on the routing ratio, this begs the question as to whether memoryless local routing algorithms can be developed for even sparser plane graphs with at most $\alpha|V| + O(1)$ directed edges for some $2 \leq \alpha < 4$.

Our local routing algorithm, which only requires memory to store one distance in the graph, has a routing ratio of at most 85.17 for $\vec{\text{Yao}}_4^\infty$ -graphs. We also give an alternate local routing algorithm which is memoryless, and has routing ratio at most 222.20. Furthermore, we show that no local routing algorithm for $\vec{\text{Yao}}_4^\infty$ -graphs can have a routing ratio lower than $7 + \sqrt{2} \approx 8.41$. Moreover, we prove that the spanning ratio is at least $5 + \sqrt{2} \approx 6.41$ in the worst case. Prior to this work, there was no known upper bound on the spanning ratio of $\vec{\text{Yao}}_4^\infty$ -graphs. The techniques we develop in this paper also allow us to prove lower bounds of $7 - \sqrt{3} + \sqrt{5 - 2\sqrt{3}} \approx 6.51$ and $7 + \sqrt{2}$ for the spanning and routing ratios of $\vec{\text{Yao}}_4$, respectively.

5.2 Introduction

A fundamental problem in geometric routing is to construct a directed plane graph on a given set of points that supports local routing with constant routing ratio, using the fewest number of edges. The routing ratio of a routing algorithm is an upper bound on the ratio of the length of the path that it finds, divided by the length of the shortest path. Importantly, a local algorithm does not have access to the full graph. This problem has applications in wireless sensor networks, robotic navigation, and distributed computing. Routing is inherently directional, and in many real-world settings, signals are sent from a transmitter to an antenna. This directional nature is particularly relevant in wireless communication, where signal transmission can be costly. Wide-angle broadcasting consumes excessive power, and long-distance transmission is expensive, with energy consumption proportional to the distance squared. As a result, nodes should send signals in restricted directions, ensuring that each transmitted signal has a unique receiver. To reduce the effects of interference, we desire planarity, meaning that signals do not cross. However, we allow bidirectional edges, which count as two directed edges, since otherwise, not all point sets admit strongly-connected directed plane graphs. Disallowing bidirectional edges presents different challenges and has led to the study of oriented spanners [25]. Note that every undirected graph can be converted into a directed graph by replacing every undirected edge with two directed edges. This implies that every routing result on undirected graphs can also be viewed as a routing result on directed graphs with twice as many edges. Thus, the challenge is to construct directed plane graphs that support local routing while minimizing the number of directed edges.

Routing on graphs in an online fashion, without relying on precomputed routing tables is a compelling motivation for studying routing on geometric embeddings of graphs. The use of routing tables can be circumvented by exploiting geometric properties of the graph, such as empty proximity regions, to guide path-finding. Routing tables are often inefficient to compute and difficult to maintain in a dynamic setting, compared to maintaining a geometric embedding of a graph. Embeddings like Yao and Theta graphs provide a framework for leveraging these empty proximity regions to facilitate efficient routing. Our focus on the Yao_4^∞ -graph stems from its unique combination of properties: it is planar, has a bounded out-degree of 4, and supports a novel local routing algorithm with constant routing ratio.

For decades, researchers have been studying the distance-preserving properties of geometric graphs, such as the Delaunay triangulation [39]. The vertex set of a geometric graph is a finite set of points in the plane and the edges are weighted by the L_2 -norm. The Delaunay triangulation has an edge between two vertices exactly when they lie on the boundary of a disk which contains no other vertex in its interior [30]. When the disk is replaced with a square, then we obtain an L_∞ -Delaunay graph. Also a geometric graph of interest is the Yao graph. For any $k > 2$, a Yao_k -graph is constructed by considering k equal-sized cones around each vertex and connecting each vertex to the *nearest* point in each cone [46]. In this way, a graph on n points has at most nk edges. Normally, *nearest* is defined using the L_2 -norm, however in the case of $k = 4$, replacing L_2 with L_∞ is a natural choice because the resulting graph, denoted Yao_4^∞ , is a subgraph of the well-known L_∞ -Delaunay graph. In other words, the Yao_4^∞ -graph has an edge between two vertices when there exists an empty square with one endpoint on a corner and

the other endpoint on an opposing side. If the graph is defined using the L_1 -norm, then we obtain the well-studied Θ_4 -graph.

Given a geometric graph G , the spanning ratio of a subgraph H is a measure of how well distances of G are preserved. A subgraph H is called a c -spanner of G if for every edge $uv \in G$, the shortest path in H from u to v is at most c times the length of the edge uv in G [39]. The smallest such c is referred to as the spanning ratio of H . Throughout this paper, we consider G to be the complete graph with edges weighted by Euclidean distance. The undirected Yao₄-graph was first proved to be a spanner by Bose et al. [19] with a stretch factor of approximately 662. As part of the proof, the authors established that the undirected Yao₄[∞]-graph is an 8-spanner. Then, Bonichon et al. [10] proved that the Yao₄[∞]-graph has spanning ratio at most 6.31. Later, Damian and Nelavalli [29] improved both results by showing that the Yao₄ and Yao₄[∞]-graphs have spanning ratios of at most 54.62 and 4.93, respectively. We will denote the directed versions of these graphs using an arrow: for example, the directed version of Yao₄[∞] is denoted $\vec{\text{Yao}}_4^\infty$. Despite being a natural construction, the $\vec{\text{Yao}}_4^\infty$ -graph previously had no known upper bound on the spanning ratio. The spanning ratio of the L_∞ -Delaunay graph was originally shown to be at most $\sqrt{10} \approx 3.16$ by Chew [26]. Then, Bonichon et al. [9] improved the spanning ratio of the L_∞ -Delaunay triangulation to $\sqrt{4 + 2\sqrt{2}} \approx 2.61$, which is tight. Their lower bound construction also shows that Yao₄[∞] has a spanning ratio of at least 2.61.

While spanning ratios deal with the existence of short paths, routing ratios are based on finding short directed paths with only local information. Roughly speaking, a path-finding algorithm has routing ratio c if the paths produced have length at most c times the straight-line distance from source to destination. In the context of plane graphs, Chew gave a local routing algorithm for L_∞ -Delaunay graphs in 1986 [26]. An L_∞ -Delaunay graph can contain nearly $6|V|$ directed edges in the worst case. Since then, Bonichon et al. [8] defined a number of bounded-degree plane spanners, including G_9 and G_{12} . See Table 5.1. In [20], local routing algorithms were provided for these graphs. A parametrized construction, denoted $\mathcal{MBDG}$, was used in [2] to attain several lightness properties. In [34], the authors give a memoryless local routing algorithm for a graph construction with constant degree [35]. In all of these plane graphs, the number of directed edges required is still at least $6|V| - C$ for some constant C . In this paper, we define a local routing algorithm for the $\vec{\text{Yao}}_4^\infty$ -graph, which contains at most $4|V|$ directed edges, which is the fewest number of edges among all known plane directed graphs that admit a local routing algorithm. Towards answering whether it is possible to route in even sparser plane graphs with at most $\alpha|V| + O(1)$ directed edges, we show that any directed spanner must have at least $2|V| - 2$ directed edges, yielding a lower bound of 2 for α .

In any bounded out-degree graph, each vertex only needs to store a constant amount of information to fully describe its outgoing neighbourhood. As a result, bounded out-degree directed graphs are a natural setting for local routing. For example, when k is larger than 6, a reasonable local routing algorithm for Yao_k-graphs is simply *cone routing*, where each decision is to move to the neighbour in the same cone as the destination. When k is at most 6, the Yao_k-graph is less dense and cone routing can fail to produce short paths. However, [32] provides a local routing algorithm for the Yao₆-graph with routing ratio at most 22.94. The smallest possible k for which the Yao_k-graph is strongly connected is $k = 4$, and Bose et al.

[21] recently provided a local routing algorithm achieving a routing ratio of at most 23.36 for $\vec{\text{Yao}}_4$ -graphs. Furthermore, they show how to modify the path output by their local routing algorithm to upper bound the spanning ratio by 16.54 for undirected Yao_4 -graphs. Interestingly, their algorithm is based on the Greedy/Sweep algorithm from [13] which has a routing ratio of 17 in the directed $\vec{\Theta}_4$ -graph.

Our local routing algorithm extends the Greedy/Sweep approach from $\vec{\Theta}_4$ -graphs to $\vec{\text{Yao}}_4^\infty$ -graphs. The original Greedy/Sweep algorithm is quite simple. If the current vertex is u and the target vertex is t , then we consider the right triangle T bounded by vertical and horizontal lines through u and a *diagonal* line through t with slope -1 . If no vertices are in T , we say that T is *clean* and we take a *greedy* step in the cone containing t . Otherwise, if T is not *clean*, then we take a *sweeping* step in the cone of T . The authors of [21] adapted the Greedy/Sweep approach to $\vec{\text{Yao}}_4$ -graphs by replacing the right triangle with a wedge. However, in the case of $\vec{\text{Yao}}_4^\infty$ -graphs, a direct application of the Greedy/Sweep approach fails miserably, resulting in long paths. The main challenge is that *sweeping* steps can cross the diagonal while making negligible progress towards the destination. To overcome this, we modify the definition of *clean* to penalize such diagonal crossings and force provable progress toward the target. Given our redefinition of cleanliness, we refer to our local routing algorithm as the DIRTY Algorithm (**D**irected **I**nfinity **R**outing **T**hrough **Y**ao). Our analysis yields an upper bound of 85.17 on the routing ratio of our local routing algorithm. We also provide a memoryless version of our algorithm with a routing ratio of at most 222.20.

Notice that our routing ratio of 85.17 is also the best known upper bound on the spanning ratio for $\vec{\text{Yao}}_4^\infty$ -graphs. Next, we give a lower bound of $5 + \sqrt{2} \approx 6.41$ on the spanning ratio of $\vec{\text{Yao}}_4^\infty$ -graphs in the worst case, highlighting a clear gap compared to the upper bound of 4.93 for Yao_4^∞ -graphs. Finally, we prove that no local routing algorithm for $\vec{\text{Yao}}_4^\infty$ -graphs can achieve a routing ratio below $7 + \sqrt{2} \approx 8.41$. Additionally, we use a similar technique to provide lower bounds of $7 - \sqrt{3} + \sqrt{5 - 2\sqrt{3}} \approx 6.51$ and $7 + \sqrt{2}$ for the spanning and routing ratios of $\vec{\text{Yao}}_4$, respectively.

5.3 Preliminaries

For any point $u \in \mathbb{R}^2$, we let $x(u)$ and $y(u)$ denote the x - and y -coordinates of u , respectively. We denote the line segment between points u, v as uv , and define $\|uv\|_x := |x(u) - x(v)|$ and $\|uv\|_y := |y(u) - y(v)|$. For $p \in [1, \infty)$, the L_p -length of uv is denoted $\|uv\|_p := (\|uv\|_x^p + \|uv\|_y^p)^{\frac{1}{p}}$. We also define $\|uv\|_\infty := \max(\|uv\|_x, \|uv\|_y)$. A geometric graph is a graph whose vertex set contains points in the plane and whose edge weights are the L_2 (Euclidean) lengths of the corresponding line segments. In a geometric graph, each vertex is identified with its coordinates. Throughout the paper, we make the *general position assumption* that no two vertices have the same x - or y -coordinates and all distances between pairs of vertices are unique in the L_∞ -norm. On such a point set, the edges of the $\vec{\text{Yao}}_4^\infty$ -graph are well-defined. Since this paper studies directed graphs, all paths in directed graphs are directed unless explicitly stated otherwise. For two vertices u, v in a directed geometric graph G , the length of the shortest path from u to v

Reference	Graph	Planar guarantee	Max out-degree	Routing ratio	Extra vertex memory	Memoryless algorithm
[21]	$\vec{\text{Yao}}_4$	No	4	23.43	No	Yes
[32]	$\vec{\text{Yao}}_6$	No	6	22.94	No	Yes
[4]	$\vec{\text{Yao}}_k$ ($k \geq 7$)	No	k	$\leq \frac{1}{1-2\sin(\pi/k)}$	No	Yes
[13]	$\vec{\Theta}_4$	No	4	17.03	No	Yes
[1]	$\vec{\Theta}_6$	No	6	8	No	Yes
[14]	$\vec{\Theta}_k$ ($k \geq 7$)	No	k	$\leq \frac{1}{1-2\sin(\pi/k)}$	No	Yes
[34]	(Thm 3.4)	No	7	40.79	Yes	Yes
[2]	$\mathcal{MBDG}$	Yes	≥ 20	≥ 9.27	Yes	No
[20]	G_{12}	Yes	12	54.85	No	No
[20]	G_9	Yes	9	8.66	Yes	No
This paper	$\vec{\text{Yao}}_4^\infty$	Yes	4	85.17	No	No
This paper	$\vec{\text{Yao}}_4^\infty$	Yes	4	222.20	No	Yes

Table 5.1: Comparison of upper bounds on the routing ratio of geometric graphs with bounded out-degree. When the information stored at a vertex v contains more than just the list of neighbouring vertex coordinates, then we say that the vertex v uses extra memory.

in G is denoted $d_G(u, v)$. Then for a real number $c \geq 1$, G is said to be a c -spanner if for all pairs (u, v) of vertices u, v in G , we have $d_G(u, v) \leq c\|uv\|_2$. The spanning ratio of a class of graphs $\mathcal{G}$ is the infimum of $\{c \in \mathbb{R} \mid \forall G \in \mathcal{G}, G \text{ is a } c\text{-spanner}\}$.

When routing in a geometric graph, each vertex is identified with its coordinates. Here, one unit of memory is $\log_2(|V(G)|)$ bits in order to store points on a polynomially-sized grid. Formally, a local routing algorithm is a function that takes as input $(s, N(s), t, M)$, and outputs some memory register M' and a vertex $p \in N(s)$ where s is the current vertex, $N(s)$ is the outgoing neighbourhood of s , t is the destination, and both M, M' are memory registers that can hold m bits each for some $m \in \mathbb{Z}$. If $m = 0$, then the local routing algorithm is referred to as *memoryless*. The routing ratio of a routing algorithm defined for a family of directed geometric graphs $\mathcal{G}$ is the infimum over all c such that the path output by the algorithm for any pair of vertices $s, t \in V(G)$ for $G \in \mathcal{G}$ has length at most $c\|st\|_2$. Note that the routing ratio is an upper bound on the spanning ratio.

5.4 A Local Routing Algorithm for $\vec{\text{Yao}}_4^\infty$ -Graphs

We will now describe our local routing algorithm for $\vec{\text{Yao}}_4^\infty$ -graphs. Without loss of generality, assume the coordinates of t are $(0, 0)$ and $\|st\|_\infty = 1$. Let $0 < \delta < 0.5$, and define the *green square*  to be the set of points $p \in \mathbb{R}^2$ such that $\|pt\|_\infty \leq 2\delta$. In addition, we define the diagonal lines ℓ_t^-, ℓ_t^+ with slope ± 1 passing through t . Define the north, east, south and western quadrants as the cones delimited by ℓ_t^+ and ℓ_t^- . Refer to the four cones as , , , .

respectively.

Next, for a point $p \in \mathbb{R}^2$, we define its *height*, $h(p)$, as the L_1 distance from p to the nearest diagonal through t . Notice that $h(p) = ||x(p)| - |y(p)||$. The *bands*, denoted by $\bowtie$, are the regions close to the diagonals, excluding the green square. More precisely, let $\bowtie$ refer to the set of points $p \in \mathbb{R}^2$ such that $p \notin \blacksquare$ and $h(p) \leq \delta$. Refer to Figure 5.1. Note that $\bowtie \subset \mathbb{R}^2$ and is composed of four regions, which we label $\bowtie_{NE}, \bowtie_{SE}, \bowtie_{SW}, \bowtie_{NW}$ clockwise from the north-eastern region. Then define the truncated wedges $\boxtimes$ as those points $p \in \mathbb{R}^2$ such that $p \notin \blacksquare \cup \bowtie$. We have $\boxtimes \subset \mathbb{R}^2$ and since $\boxtimes$ is composed of four regions, we label them $\boxtimes_{NE}, \boxtimes_{SE}, \boxtimes_{SW}, \boxtimes_{NW}$ clockwise from the northern region.

Next we describe the terminology used in our local routing algorithm, which we will refer to as DIRTY Algorithm 3 (Directed Infinity Routing Through Yao). For any vertex $u \in \bowtie$, a *band step* from u follows the edge in the cone of u containing t . For any vertex $u \in \boxtimes$, u is *clean* if the quadrant of u facing the closest diagonal $\ell_t \in \{\ell_t^-, \ell_t^+\}$ does not contain any vertex v such that $||uv||_\infty + \delta \leq h(u)$. Intuitively, u is clean if there are no points between u and the nearest diagonal. For example, see Figure 5.1. A *sweep step* from u follows the edge in the cone towards the closest diagonal to u . We denote the vertex resulting from a sweep step from u as $\mathbf{Sweep}(u, t, N(u), \delta)$. A *greedy step* from u follows the edge in the cone of u containing t . The vertex resulting from a greedy step from u is denoted $\mathbf{Greedy}(u, t, N(u))$. Note that a band step is technically a greedy step, however we will distinguish band steps from greedy steps in the analysis. Now we will present the local routing algorithm. In short, if there is a neighbour in the green square, then we choose it and rescale δ . On the other hand, we take a greedy edge if the current vertex is in $\bowtie$ or is clean. When the current vertex is not clean, we take a sweep edge. A partial trace of Algorithm 3 is shown in Figure 5.1. The constant $0 < c < 1$ in step 3 of Algorithm 3 will be set to 0.08, however the analysis holds for a general c . Furthermore, δ should be set to $c||st||_\infty$ for the first call of Algorithm 3.

Algorithm 3: DIRTY (Directed Infinity Routing Through Yao)

Local routing decision in $\vec{\text{Yao}}_4^\infty$ -graph.

Data:

- Current vertex $u \in S$
- Destination vertex $t \in S$
- Neighbourhood $N(u)$ of u
- Parameter $\delta \in \mathbb{R}_{\geq 0}$

Result: A neighbour $v \in N(u)$ and a parameter $\delta' \in \mathbb{R}_{\geq 0}$

1. If $N(u) \cap \blacksquare \neq \emptyset$, select an arbitrary $v \in N(u) \cap \blacksquare$ and set $\delta' := c||vt||_\infty$. END
 2. Set $\delta' := \delta$. If $u \in \bowtie$ or u is clean, then $v := \mathbf{Greedy}(u, t, N(u))$. Else, $v := \mathbf{Sweep}(u, t, N(u), \delta)$.
-

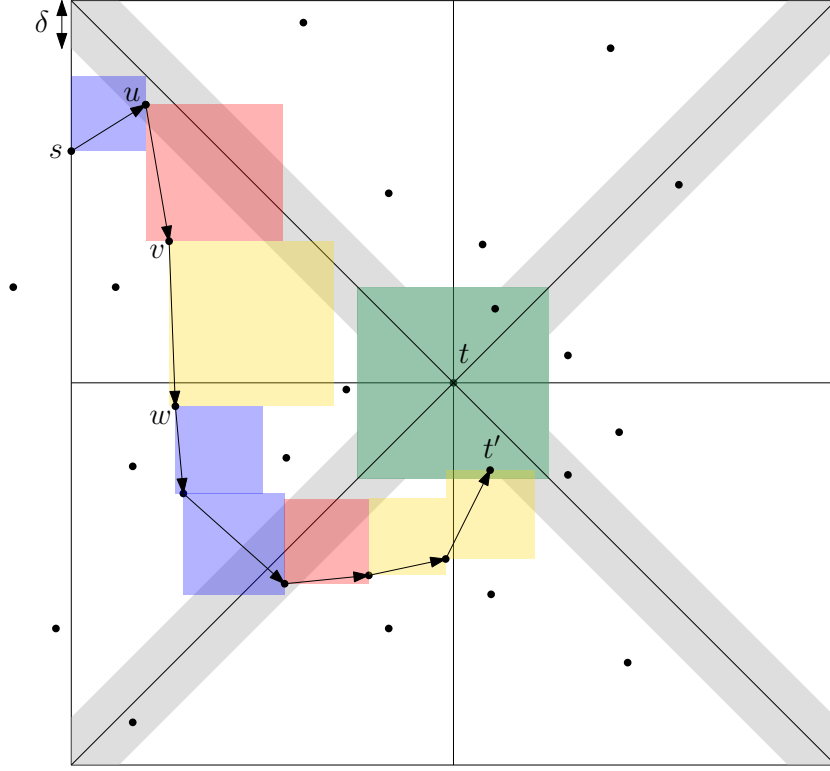


Fig. 5.1: We show the partial trace of DIRTY Algorithm 3 until we reach a vertex in the green square, . The grey regions are the *bands*, denoted as $\bowtie$, and the white regions are the truncated wedges, denoted as $\boxtimes$. The sweep edges are in blue squares, the band edges are in red squares, and the greedy edges are in gold squares. For example, su is a sweep step since s is not clean. Next, uv is a band step as $u \in \bowtie$. Then v is clean, hence vw is a greedy edge.

5.4.1 Analysis of DIRTY Algorithm 3

If edge uv is chosen by Algorithm 3, then notice that $\|ut\|_\infty \geq \|vt\|_\infty$ regardless of whether uv is a sweep, a greedy, or a band edge. In fact, by the general position assumption that all vertices have distinct x - and y -coordinates, we have $\|ut\|_\infty > \|vt\|_\infty$.

Remark 3. If uv is an edge chosen by Algorithm 3, then $\|ut\|_\infty > \|vt\|_\infty$. That is, the vertex v is inside the square centered at t with u on the boundary.

The fact that the above remark holds even for sweep edges is a result of the definition of a vertex being clean, limiting the length of any sweep edge. The vertex set being finite means that choosing each successive edge according to Algorithm 3 will output a path from s to t , which we will denote $\mathcal{P}$. For any two vertices u, v on path $\mathcal{P}$ with v occurring after u , we let $\mathcal{P}_{u,v}$ denote the sub-path of $\mathcal{P}$ from u to v .

We define t' to be the first vertex of $\mathcal{P}$ chosen by step 3 of Algorithm 3. In other words, t' is the first vertex of $\mathcal{P}$ such that $\|t't\|_\infty \leq 2c\|st\|_\infty$ because δ is initially set to $c\|st\|_\infty$ and $t' \in \text{green square}$. We will focus on upper bounding the length of $\mathcal{P}_{s,t'}$, which is a path such that each

edge is chosen according to step 3 except the last edge. As such, we will only refer to edges of $\mathcal{P}_{s,t'}$. In particular, define the set $\mathcal{S}$ as the sweep edges of $\mathcal{P}_{s,t'}$ and define the set $\mathcal{G}$ as the greedy edges uv of $\mathcal{P}_{s,t'}$, where $u \in \blacktriangleleft$. Next we define the set of band edges, $\mathcal{B}$, as the greedy edges uv of $\mathcal{P}_{s,t'}$, where $u \in \blacktriangleright$. These definitions allow us to partition the edges: $\{uv \mid uv \text{ is an edge on } \mathcal{P}_{s,t'}\} = \mathcal{S} \cup \mathcal{G} \cup \mathcal{B}$.

Lemma 10. *If $uv \in \mathcal{S}$ and uv does not cross a diagonal, then $\|uv\|_2 \leq h(u) - h(v)$.*

Proof. Since u, v both lie in the same cone among $\blacktriangleleft, \blacktriangle, \blacktriangleright, \blacktriangleright$, then $\|uv\|_2 \leq \|uv\|_1 = h(u) - h(v)$. $\square$

Lemma 11. *If $uv \in \mathcal{S}$ and uv crosses a diagonal of t , then $h(u) - h(v) \geq 2\delta$, and $\max(\|uv\|_\infty, \|vt\|_\infty) \leq \|ut\|_\infty - \delta$.*

Proof. Assume without loss of generality that $u \in \blacktriangleleft$ and uv crosses ℓ_t^+ . This means that $v \in \blacktriangleright$, as in Figure 5.2. Since u is not clean, we have $\|uv\|_\infty + \delta \leq h(u) \leq \|ut\|_\infty$. Also since v is above ℓ_t^+ we have $h(v) + \delta \leq \|uv\|_\infty$. Combining these two inequalities yields $h(u) - h(v) \geq 2\delta$. Next, $u \in \blacktriangleleft$ and $v \in \blacktriangleright$ together imply $h(u) = x(u) - y(u)$, so we have $\|ut\|_\infty - \|vt\|_\infty = x(u) - y(v) = h(u) + y(u) - y(v) \geq \|uv\|_\infty + \delta + y(u) - y(v) \geq \delta$. $\square$

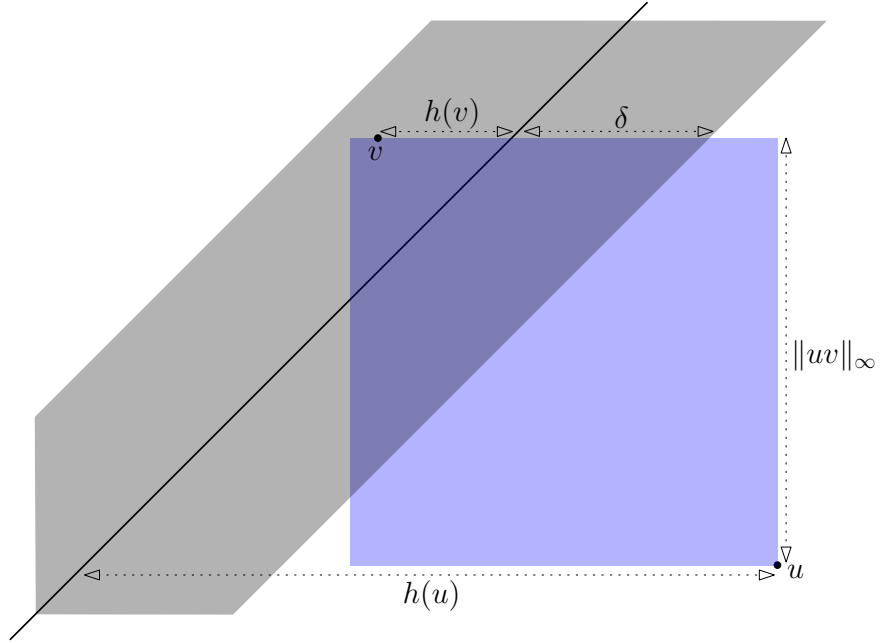


Fig. 5.2: Lemma 11: An example of a sweep edge uv that crosses the diagonal ℓ_t^+ of t .

Lemma 12. *If $uv \in \mathcal{G}$, then $h(v) - h(u) \leq \|uv\|_\infty$.*

Proof. Suppose without loss of generality that $u \in \blacktriangleleft$ with $y(u) > 0$. Then since uv is greedy, we have $x(v) < x(u) \leq x(v) + \|uv\|_\infty$ and $y(v) < y(u) \leq y(v) + \|uv\|_\infty$. Combining these

inequalities yields

$$x(v) - y(v) \leq x(u) - y(u) + \|uv\|_\infty \text{ and } y(v) - x(v) \leq y(u) - x(u) + \|uv\|_\infty.$$

Since $0 \leq h(u) = |x(u) - y(u)|$, then we have shown $|x(v) - y(v)| \leq \|uv\|_\infty + h(u)$. The result follows from the reverse triangle inequality $h(v) = \|x(v) - y(v)\| \leq |x(v) - y(v)|$. $\square$

Next we will characterize the behaviour of band edges.

Lemma 13. *If $uv \in \mathcal{B}$, then $\max_{\psi \in \{x, y\}} (\|uv\|_\psi + \|vt\|_\psi - \|ut\|_\psi) \leq 2\delta$.*

Proof. Assume without loss of generality that $u \in \nwarrow \nearrow$. Notice that $\|uv\|_y + \|vt\|_y - \|ut\|_y = (y(u) - y(v)) + |y(v) - y(u)| = |y(v) - y(u)|$, which is at most 2δ when $y(v) \geq -\delta$. Assume towards a contradiction that $y(v) < -\delta$. When u is above ℓ_t^+ , then $\|uv\|_\infty \geq y(u) - y(v) > y(u) + \delta > y(u) = \|ut\|_\infty$. Otherwise when u is below the diagonal ℓ_t^+ , we have $\|uv\|_\infty \geq y(u) - y(v) > y(u) + \delta \geq x(u) = \|ut\|_\infty$. Either way is a contradiction since t and v are in the same cone of u , so $\|uv\|_\infty \leq \|ut\|_\infty$. A symmetric argument completes the proof when $\psi = x$. $\square$

Lemma 13 tells us that when uv is a band edge with $u \in \nwarrow \nearrow$, then $v \in \nwarrow \nearrow \cup \nwarrow \swarrow \cup \searrow \swarrow \cup \blacksquare$. Indeed, the proof describes why $y(v) \geq -\delta$ and $x(v) \geq -\delta$, and the region $\{p \in \mathbb{R}^2 \mid \min(x(p), y(p)) \geq -\delta\}$ is a subset of $\nwarrow \nearrow \cup \nwarrow \swarrow \cup \searrow \swarrow \cup \blacksquare$. In other words, a sequence of consecutive band edges remains in the same region of $\nwarrow \nearrow$ until it ends in a neighbouring truncated wedge, or in the green square. To formalize this, we will define a set $\mathcal{B}'$ containing pairs of vertices that correspond to maximal sub-paths of band edges.

Definition 6 (Maximal sub-paths of band edges). Define the set $\mathcal{B}'$ as follows. Let u, v be vertices of $\mathcal{P}_{s,t'}$ such that $\mathcal{P}_{u,v}$ is a sequence of band edges. We let the pair $(u, v) \in \mathcal{B}'$ if and only if $u \in \nwarrow \nearrow$, $v \in \nwarrow \swarrow$, and u is not the tip of a band edge.

Notice that if $(u, v) \in \mathcal{B}'$, then $\mathcal{P}_{u,v}$ must be x - and y -monotone since all edges between u and v are band edges in the same cone.

Lemma 14. *For all $(u, v) \in \mathcal{B}'$, we have $h(v) - h(u) \leq \|uv\|_\infty$.*

Proof. Suppose without loss of generality that $u \in \nwarrow \nearrow$ and $v \in \nwarrow \swarrow$. Then we have $x(v) < x(u) \leq x(v) + \|uv\|_\infty$ and $y(v) < y(u) \leq y(v) + \|uv\|_\infty$. The rest of the proof is identical to that of Lemma 12. $\square$

To better understand which band edges are accounted for by paths in $\mathcal{B}'$, we provide the following definition.

Definition 7 (Vertices v' and u'). Let v' be the last vertex of $\mathcal{P}_{s,t'}$ in $\nwarrow \swarrow$. If no such v' exists, $v' := s$. Then, let u' be the vertex directly after v' on $\mathcal{P}_{s,t'}$.

Notice that all band edges of $\mathcal{P}_{s,v'}$ are contained in a sub-path $\mathcal{P}_{u,v}$ for some $(u,v) \in \mathcal{B}'$. In addition, the path $\mathcal{P}_{u',t'}$ consists entirely of band edges.

Lemma 15. *We have $\sum_{uv \in \mathcal{G}} (\|ut\|_\infty - \|vt\|_\infty) + \sum_{(u,v) \in \mathcal{B}'} (\|ut\|_\infty - \|vt\|_\infty) \leq 1$.*

Proof. Each difference $\|ut\|_\infty - \|vt\|_\infty$ in the following telescoping sum corresponds to a sub-path of $\mathcal{P}_{s,u'}$ and is therefore positive by Remark 3.

$$\sum_{uv \in \mathcal{G} \cup \mathcal{S}} (\|ut\|_\infty - \|vt\|_\infty) + \sum_{(u,v) \in \mathcal{B}'} (\|ut\|_\infty - \|vt\|_\infty) = \|st\|_\infty - \|u't\|_\infty \leq 1.$$

Dropping the terms corresponding to sweep edges yields the desired inequality. $\square$

Now we can establish a bound on the sweep edges that depends on the greedy edges and $\mathcal{B}'$ paths.

Lemma 16. *Let $k := \lfloor \frac{1-2\delta}{\delta} \rfloor + 1$. We have*

$$\sum_{uv \in \mathcal{S}} \|uv\|_2 \leq 1 + \sqrt{2}(k - \delta \frac{k(k+1)}{2}) - 2\delta k + \sum_{uv \in \mathcal{G}} \|uv\|_\infty + \sum_{(u,v) \in \mathcal{B}'} \|uv\|_\infty.$$

Proof. First, we split the sum based on whether the sweeping edge crosses a diagonal. Let $\mathcal{X} \subseteq \mathcal{S}$ denote the set of sweep edges that cross a diagonal ℓ_t^+ or ℓ_t^- .

$$\begin{aligned} \sum_{uv \in \mathcal{S}} \|uv\|_2 &= \sum_{uv \in \mathcal{S} \setminus \mathcal{X}} \|uv\|_2 + \sum_{uv \in \mathcal{X}} \|uv\|_2 \\ &\leq \sum_{uv \in \mathcal{S} \setminus \mathcal{X}} (h(u) - h(v)) + \sum_{uv \in \mathcal{X}} \|uv\|_2 && \text{By Lemma 10} \\ &= \sum_{uv \in \mathcal{S}} (h(u) - h(v)) + \sum_{uv \in \mathcal{X}} (\|uv\|_2 - h(u) + h(v)) \end{aligned}$$

Next, by Definition 7, the path $\mathcal{P}_{s,u'}$ is precisely all the greedy edges, sweep edges, and sub-paths of band edges $\{\mathcal{P}_{u,v} \mid (u,v) \in \mathcal{B}'\}$. This means that

$$\sum_{uv \in \mathcal{S} \cup \mathcal{G}} (h(u) - h(v)) + \sum_{(u,v) \in \mathcal{B}'} (h(u) - h(v)) = h(s) - h(u') \quad (5.1)$$

Next we rearrange (5.1) and use Lemmas 12 and 14:

$$\begin{aligned} \sum_{uv \in \mathcal{S}} (h(u) - h(v)) &= h(s) - h(u') + \sum_{uv \in \mathcal{G}} (h(v) - h(u)) + \sum_{(u,v) \in \mathcal{B}'} (h(v) - h(u)) \\ &\leq 1 + \sum_{uv \in \mathcal{G}} \|uv\|_\infty + \sum_{(u,v) \in \mathcal{B}'} \|uv\|_\infty \end{aligned}$$

Now, let $m := |\mathcal{X}| - 1$, and label the edges of $\mathcal{X}$ as $u_0v_0, \dots, u_mv_m$ in the order they appear on the path $\mathcal{P}_{s,t'}$. By Lemma 11, we have $\|u_it\|_\infty - \|v_it\|_\infty \geq \delta$ for all $i \in \{0, \dots, m\}$. Additionally, u_{i+1} appears after v_i on the path $\mathcal{P}_{s,t'}$, then by Remark 3 we have $\|u_{i+1}t\|_\infty \leq \|v_it\|_\infty$ for $i \in \{0, \dots, m-1\}$. Since $\|u_0t\|_\infty \leq \|st\|_\infty = 1$, the above inequalities imply that $\|u_it\|_\infty \leq 1 - i\delta$ for $i \in \{0, \dots, m\}$. Next, vertex t' was defined to be the first vertex on the path $\mathcal{P}$ such that $\|t't\|_\infty \leq 2\delta$, therefore we must have $\|u_mt\|_\infty > 2\delta$ since u_mv_m is an edge in $\mathcal{P}_{s,t'}$. Combining inequalities yields $1 - m\delta > 2\delta$, and isolating m yields $m \leq \lfloor \frac{1-2\delta}{\delta} \rfloor = k-1$. Also, Lemma 11 gives us $\|u_iv_i\|_\infty + \delta \leq \|u_it\|_\infty \leq 1 - i\delta$ and $h(u_i) - h(v_i) \geq 2\delta$. The result follows:

$$\begin{aligned} \sum_{uv \in \mathcal{X}} (\|uv\|_2 - h(u) + h(v)) &\leq \sum_{i=0}^m (\sqrt{2}\|u_iv_i\|_\infty - h(u_i) + h(v_i)) \quad \text{By } \|uv\|_2 \leq \sqrt{2}\|uv\|_\infty \\ &\leq \sum_{i=0}^{k-1} (\sqrt{2}(1 - i\delta - \delta) - 2\delta) \\ &= \sqrt{2}(k - \delta \frac{k(k+1)}{2}) - 2\delta k. \end{aligned}$$

□

From Lemma 16, we can see that the total L_2 distance of sweep edges can be upper bounded if we upper bound the total L_∞ distance of greedy edges and pairs in $\mathcal{B}'$.

Lemma 17. *We have $\sum_{uv \in \mathcal{G}} \|uv\|_\infty \leq 16 - 8\delta$.*

Proof. Consider the set $\mathcal{G}' \subseteq \mathcal{G}$ of greedy edges uv where $u \in \blacktriangleleft$ and $y(u) > 0$. We will show that $\sum_{uv \in \mathcal{G}'} \|uv\|_\infty \leq 2 - \delta$ using a disjoint projection argument illustrated in Figure 5.3.

Firstly, label the edges of $\mathcal{G}'$ as $u_1v_1, \dots, u_mv_m$. Note that the edges are labelled in the order they appear, and not necessarily consecutive. For $i \in \{1, \dots, m\}$, define the point $p_i := u_i - (0, \|u_iv_i\|_\infty)$. Note that $\|u_iv_i\|_\infty = \|u_ip_i\|_y$. Next, define the projection $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ as $F(x, y) := (0, x + y)$. Notice that for a point $p \in \mathbb{R}^2$, the point $F(p)$ lies on the vertical line through t with the segment $pF(p)$ having slope -1 . We will show that the segments $\{F(u_i)F(p_i) \mid 1 \leq i \leq m\}$ are disjoint.

Let $i \in \{1, \dots, m-1\}$. Firstly, since u_iv_i is greedy, we have $y(v_i) < y(u_i)$, hence $y(F(p_i)) < y(F(u_i))$. Next, we have $x(u_{i+1}) < x(u_i)$ by Remark 3 and $y(u_{i+1}) < y(u_i)$ since u_i is clean. Combining these coordinate restrictions with the fact that p_i lies on the southeast corner of the empty square corresponding to u_iv_i , we obtain $y(F(u_{i+1})) < y(F(p_i)) < y(F(u_i))$. This means that the projections are disjoint and their y -coordinates range from $y(F(p_m))$ to $y(F(u_1))$. Since $p_m \in \blacktriangleleft$, then $y(F(p_m)) \geq 0$. Also, we have $y(F(u_1)) \leq 2 - \delta$ given that $u_1 \in \blacktriangleleft$.

The proof is completed after multiplication by 8 to account for the greedy edges of $\mathcal{G}$ originating in the seven other regions symmetric to $\mathcal{G}'$. □

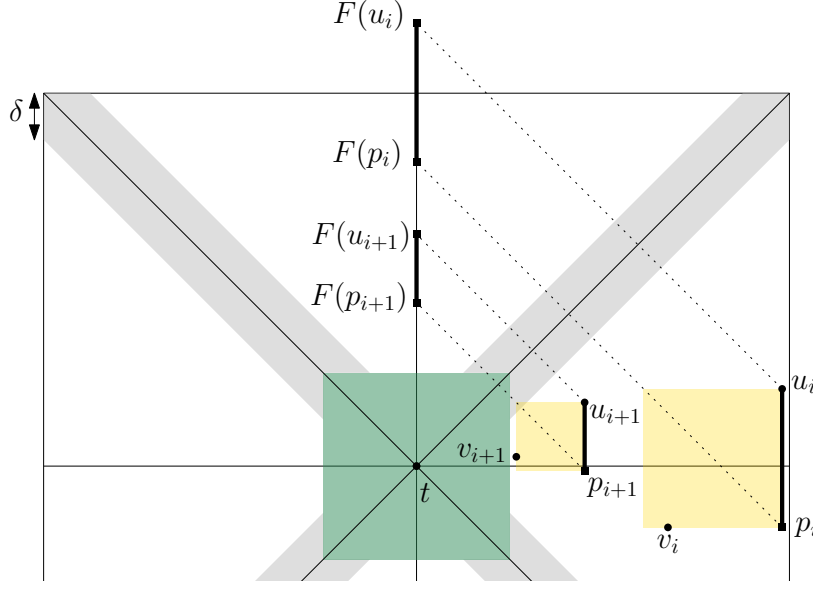


Fig. 5.3: Lemma 17: The vertical components of the edges $u_i v_i$ and $u_{i+1} v_{i+1}$ can be projected onto the y -axis using the projection F . Furthermore, the projections are disjoint.

At this stage, we could multiply the bound from Lemma 17 by $\sqrt{2}$ in order to bound the total L_2 length of the greedy edges. However, we can do better with a more careful analysis described in Lemma 18.

Lemma 18. We have $\sum_{uv \in \mathcal{G}} \|uv\|_2 \leq 12 + 4\sqrt{2} - 8\delta + \sum_{uv \in \mathcal{G}} (\|ut\|_\infty - \|vt\|_\infty)$.

Proof. We start by partitioning $\mathcal{G}$ into three parts that correspond to the following three types of greedy edges, illustrated in Figure 5.4. Suppose without loss of generality that $uv \in \mathcal{G}$ and $u \in \blacktriangleleft$ with $y(u) > 0$. Then the edge uv is Type A if $v \in \blacktriangleleft$, Type B if $v \in \blacktriangledown$, and Type C if $v \in \blacktriangleright$. Now we extend the definition of Types A, B, and C to all edges in $\mathcal{G}$ by symmetry. For example, an edge $uv \in \mathcal{G}$ with $u \in \blacktriangleright$, $0 < x(u)$ and $v \in \blacktriangleleft$ would be Type C. Define the sets $\mathbf{A}, \mathbf{B}, \mathbf{C}$ of Type A, B, and C greedy edges, respectively. This means that we have partitioned the greedy edges as follows: $\mathcal{G} = \mathbf{A} \cup \mathbf{B} \cup \mathbf{C}$. We proceed with the following claim for Type A and B edges.

Claim 3 (If $uv \in \mathbf{A} \cup \mathbf{B}$ then $\|uv\|_1 \leq \|uv\|_\infty + \|ut\|_\infty - \|vt\|_\infty$). Suppose without loss of generality that $u \in \blacktriangleleft$ and $y(u) > 0$. If $uv \in \mathbf{A}$ then $v \in \blacktriangleleft$, so $\|uv\|_x \leq \|ut\|_\infty - \|vt\|_\infty$. Combining this with $\|uv\|_y \leq \|uv\|_\infty$ yields the result. If instead $uv \in \mathbf{B}$, then $v \in \blacktriangledown$, giving us $\|uv\|_y \leq \|ut\|_\infty - \|vt\|_\infty$. Adding this to $\|uv\|_x \leq \|uv\|_\infty$ completes the claim.

Next, we prove that for Type C edges, $\sum_{uv \in \mathbf{C}} \|uv\|_\infty \leq 4$. Consider the greedy edges $u_1 v_1, \dots, u_m v_m \in \mathbf{C}$ with $u_i \in \blacktriangleleft$ labelled in the order they appear in $\mathcal{P}_{s,t'}$. Then by the two empty squares (one from u_i being clean and one from $u_i v_i$), we have $\|u_{i+1} t\|_\infty \leq \|u_i t\|_\infty - \|u_i v_i\|_\infty$ for $i \in \{1, \dots, m-1\}$ and $\|u_m v_m\|_\infty \leq \|u_m t\|_\infty$. Then we have $\sum_{i=1}^m \|u_i v_i\|_\infty \leq$

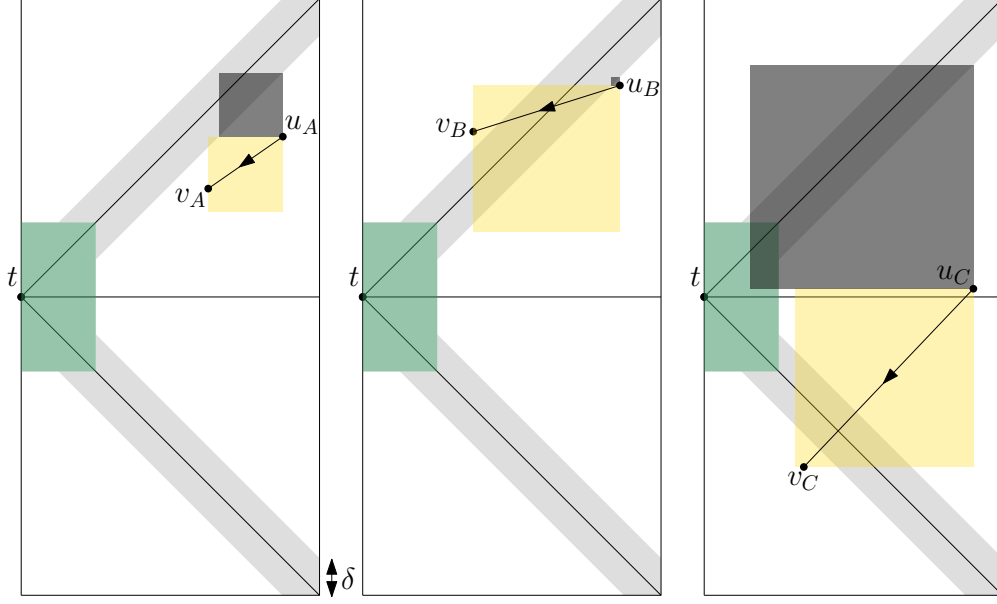


Fig. 5.4: Lemma 18: Edges $u_A v_A$, $u_B v_B$, $u_C v_C$ are examples of Type A, B and C greedy edges, respectively. Notice that u_A, u_B, u_C are all clean with respect to ℓ_t^+ , meaning that the dark grey regions are empty.

$\|u_m t\|_\infty + \sum_{i=1}^{m-1} \|u_i t\|_\infty - \|u_{i+1} t\|_\infty = \|u_1 t\|_\infty \leq 1$. The final step is multiplication by 4 to account for the four connected components of $\boxtimes$.

Now we combine the above claim with Lemma 17 to complete the proof.

$$\begin{aligned}
 \sum_{uv \in \mathcal{G}} \|uv\|_2 &\leq \sum_{uv \in A \cup B} \|uv\|_1 + \sum_{uv \in C} \|uv\|_2 \\
 &\leq \sum_{uv \in A \cup B} (\|uv\|_\infty + \|ut\|_\infty - \|vt\|_\infty) + \sum_{uv \in C} (\|uv\|_\infty + (\sqrt{2} - 1)\|uv\|_\infty) \\
 &\leq \sum_{uv \in \mathcal{G}} (\|uv\|_\infty + \|ut\|_\infty - \|vt\|_\infty) + \sum_{uv \in C} (\sqrt{2} - 1)\|uv\|_\infty \\
 &\leq \sum_{uv \in \mathcal{G}} (\|ut\|_\infty - \|vt\|_\infty) + 16 - 8\delta + 4(\sqrt{2} - 1)
 \end{aligned}$$

□

Next, a similar projection argument can be used to bound the total the length of band edges. First, we provide the following definition.

Definition 8. For $2\delta \leq r \leq 1$, define the eastern hexagon $\blacksquare_r$ to be the set of points $p \in \blacksquare$ such that $2\delta < x(p) < r$ and $|y(p)| < (r - \delta)$. Define the north, west and south hexagons of radius r similarly.

Next, we present a lemma that formalizes the idea that when the path $\mathcal{P}$ exits $\blacksquare_r$, either $\mathcal{P}$ becomes significantly closer to t , or $\mathcal{P}$ never revisits the region $\boxtimes$.

Lemma 19. *Suppose vertex $w \in \mathbb{X}$ and let v be the next vertex of $\mathcal{P}$ outside of $\mathbb{Y}_{\|wt\|_\infty}$. Then at least one of the following must be true:*

1. $v \in \text{red}$,
2. $\|vt\|_\infty \leq \|wt\|_\infty - \delta$,
3. $\mathcal{P}_{v,t'} \cap \text{red} = \emptyset$

Proof. Throughout the proof, let u denote the vertex before v in $\mathcal{P}$, meaning $u \in \blacksquare_{\|wt\|_\infty}$. Since $u \in \blacksquare_{\|wt\|_\infty} \subset \blacktriangleleft$, then we must have $v \in \blacktriangledown \cup \blacksquare \cup \blacktriangle$. Without loss of generality, assume $v \in \blacktriangledown \setminus \blacksquare$, implying that $y(v) = \|vt\|_\infty$. Since we have assumed Item 1 to be false, we will show that in each of the following three cases, Item 2 or Item 3 is true. The cases are shown in Figure 5.5.

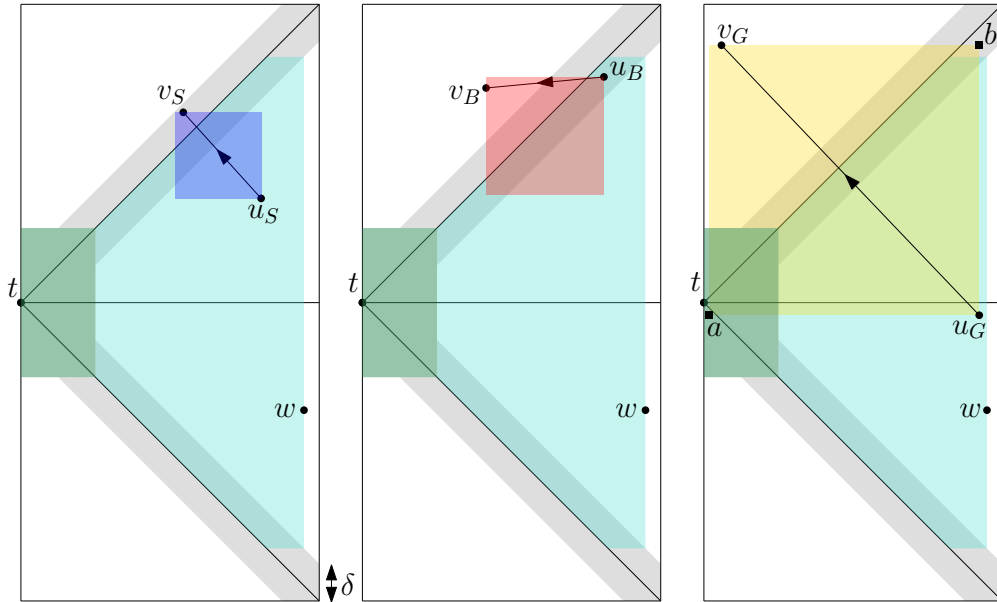


Fig. 5.5: Lemma 19: Edges $u_S v_S$, $u_B v_B$, $u_G v_G$ are examples of sweep, band, and greedy edges, respectively. The turquoise region is $\square_{\|wt\|_\infty}$. The labels a and b refer to the southwest and northeast corners of the gold square corresponding to edge $u_G v_G$.

Case: uv is a sweep edge Then we must have $u \in \boxtimes$ and $y(u) > 0$, therefore $y(v) - y(u) \leq \|uv\|_\infty \leq h(u) - \delta = x(u) - y(u) - \delta$. Item (2) follows from $x(u) = \|ut\|_\infty \leq \|wt\|_\infty$.

Case: uv is a band edge Since $v \in \blacksquare \setminus \color{blue}{\blacksquare}$, we must have $u \in \color{gray}{\blacktriangledown}$, meaning $y(v) < y(u)$. By definition of $u \in \color{gray}{\blacktriangledown}_{\|wt\|_\infty}$, we have $y(u) < \|wt\|_\infty - \delta$, which again satisfies Item (2).

Case: uv is a greedy edge Assume $\|vt\|_\infty > \|wt\|_\infty - \delta$. We will show that the region $\{p \in \square \mid \|pt\|_\infty < \|ut\|_\infty\}$ contains no vertices, implying Item (3). Since $v \in \blacktriangleleft$, then the closest diagonal to u must be ℓ_t^- . Let points a, b be the southwest and northeast corners of the empty square corresponding to uv . Then $v(a) = v(u) < 0$ and $0 < x(a)$ since the square bounding a, b

is empty. Furthermore, since $x(u) - \delta \leq \|wt\|_\infty - \delta \leq y(v) \leq y(b)$, we must have $b \in \blacktriangleright\blacktriangleleft$. This leads to $d_1(a, \ell_t^+) < \delta$, so $a \in \blacksquare$. Then since u is clean with respect to ℓ_t^- , and also $a \in \blacksquare$ and $b \in \blacktriangleright\blacktriangleleft$ then the region $\{p \in \blacktriangleleft\blacktriangleright \mid \|pt\|_\infty < \|ut\|_\infty\}$ contains no vertices. Item (3) follows since $\|vt\|_\infty < \|ut\|_\infty$ by Remark 3. $\square$

We will also make use of the following remark:

Remark 4. If $(u, v) \in \mathcal{B}'$, then v is clean with respect to the closest diagonal to u . Indeed, v is the tip of a band edge from the connected component of $\blacktriangleright\blacktriangleleft$ containing u by Lemma 13.

Now we have the tools to prove Lemma 20.

Lemma 20. We have $\sum_{(u,v) \in \mathcal{B}'} \|uv\|_\infty \leq 16 - 20\delta$.

Proof. We will first focus on paths $\mathcal{P}_{u,v}$ where $(u, v) \in \mathcal{B}'$ and $v \in \blacktriangleleft\blacktriangleright$. Let $(u_1, v_1), \dots, (u_m, v_m) \in \mathcal{B}'$ be labelled according to their order on $\mathcal{P}$, where $v_i \in \blacktriangleleft\blacktriangleright$. Recall that all edges between u_i and v_i are band edges, so by Lemma 13, we have $u_i \in \blacktriangleright\blacktriangleleft \cup \blacktriangleleft\blacktriangleright$. Then $\|u_i v_i\|_\infty = \|u_i v_i\|_y$. Consider the point $p_i := (x(u_i), y(v_i))$ and the resulting segment $u_i p_i$ with length $\|u_i v_i\|_y$. We will show how to project $u_i p_i$ onto the line $x = 2\delta$ such that when $i \neq j$, the projections of $u_i p_i$ and $u_j p_j$ are disjoint, illustrated in Figure 5.6. Furthermore, we will show that all projections lie on a sub-segment of length $4 - 5\delta$.

Define the projections $F, G : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ as $F(x, y) := (2\delta, x + y - 2\delta)$ and $G(x, y) := (2\delta, y - x + 2\delta)$. Note that for $p \in \mathbb{R}^2$, both F and G project p onto the vertical line $x = 2\delta$, and the slope of the segments $pF(p)$ and $pG(p)$ are -1 and 1 , respectively. For $1 \leq i \leq m$, we define the projection

$$\text{proj}(i) := \begin{cases} F(u_i)F(p_i) & \text{if } u_i \in \blacktriangleright\blacktriangleleft \\ G(u_i)G(p_i) & \text{if } u_i \in \blacktriangleleft\blacktriangleright \end{cases}$$

Let $1 \leq i < j \leq m$. Now we show that $\text{proj}(i)$ and $\text{proj}(j)$ are disjoint by cases. Assume without loss of generality that $u_i \in \blacktriangleright\blacktriangleleft$ so by Remark 4, v_i is clean with respect to ℓ_t^+ . We will show that $\text{proj}(j)$ is entirely below $\text{proj}(i)$ in both cases.

Case: $u_j \in \blacktriangleright\blacktriangleleft$ By Lemma 19, we have $y(u_j) < x(v_i) - \delta$. Indeed, $v_i \in \blacktriangleleft\blacktriangleright$ therefore either $\mathcal{P}_{v_i, u_j} \subset \blacksquare_{\|v_i t\|_\infty}$, or $\|u_j t\|_\infty \leq \|v_i t\|_\infty - \delta$ since $u_j \notin \blacksquare$ and $v_j \in \blacktriangleleft\blacktriangleright$. Furthermore, $x(u_j) \leq \|u_j t\|_\infty < \|v_i t\|_\infty = x(v_i)$. Combining $x(u_j) < x(v_i)$ with $y(u_j) < x(v_i) - \delta$ and the fact that v_i is clean with respect to ℓ_t^+ , we have that $y(F(u_j)) < y(F(v_i))$. Lastly, since $\mathcal{P}_{u_i, v_i}$ is x -monotone, then $x(u_i) > x(v_i)$, so $y(F(v_i)) < y(F(p_i))$. By transitivity, we have $y(F(u_j)) < y(F(p_i))$ completing this case.

Case: $u_j \in \blacktriangleleft\blacktriangleright$ Assume towards a contradiction that $y(F(p_i)) \leq y(G(p_j))$. Then since $2\delta < x(p_j) < x(p_i)$ we must have $y(p_i) < y(p_j)$. However, this implies that v_i is not clean with respect to ℓ_t^+ since $v_i \in \blacktriangleleft\blacktriangleright$ with $y(v_i) = y(p_i) < y(p_j) = y(v_j)$ and $x(v_j) < x(v_i)$.

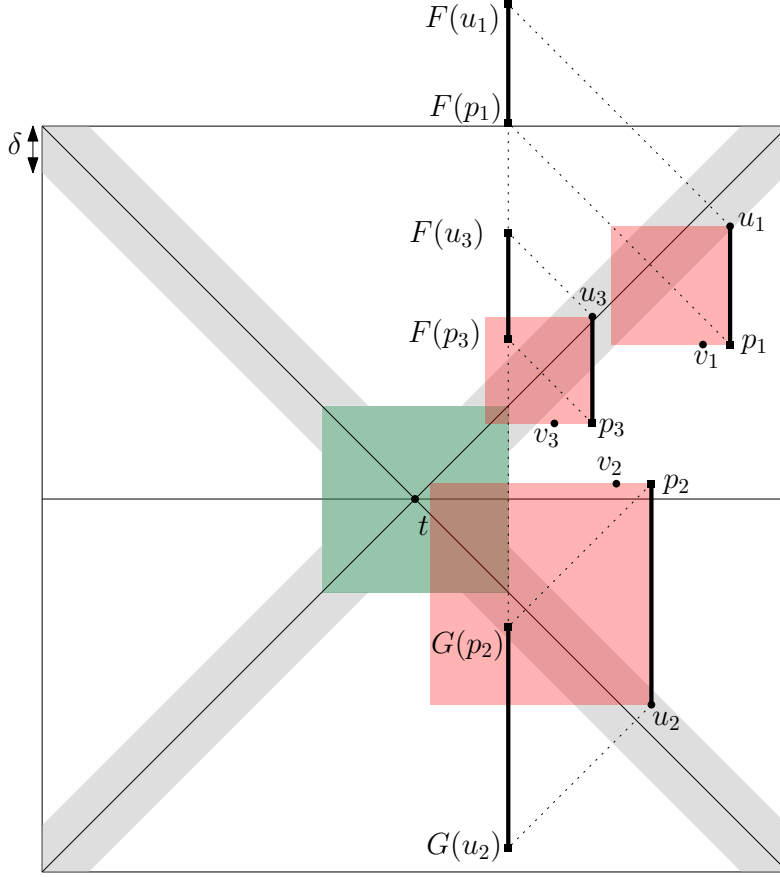


Fig. 5.6: Lemma 20: We project the vertical components of banded sequences onto the dotted line, and prove that the projections are disjoint.

Now that we have proven that the projections $\text{proj}(i)$ are disjoint on the vertical line $x = 2\delta$, we can proceed to showing that the projections lie on a sub-segment of the line with length at most $4 - 5\delta$. Assume without loss of generality that $u_1 \in \mathcal{B}'$. Then all projections lie below $y(F(u_1)) \leq y(F(1, 1)) = 2 - 2\delta$. Furthermore, we will show that all projections lie above $y(G(1, \delta - 1)) = 3\delta - 2$. Indeed, if $u_i \in \mathcal{B}'$ for all $1 \leq i \leq m$, then the projections are all above $y(F(p_m)) \geq y(F(1, \delta - 1)) = -\delta$. Otherwise, let j be the least index in $\{2, \dots, m\}$ such that $u_j \in \mathcal{B}'$. Then all projections are above $y(G(u_j))$, and by Lemma 19 we have $|y(u_j)| \leq x(v_1) - \delta \leq 1 - \delta$. This means that $y(G(u_j)) \geq y(G(1, \delta - 1)) = 3\delta - 2$. Therefore all projections $\text{proj}(i)$ have y -coordinates between $3\delta - 2$ and $2 - 2\delta$. Summarizing, we have shown

$$\sum_{\substack{(u,v) \in \mathcal{B}' \\ v \in \mathcal{B}}} \|uv\|_\infty = \sum_{\substack{(u,v) \in \mathcal{B}' \\ v \in \mathcal{B}}} \|uv\|_y \leq (2 - 2\delta) - (3\delta - 2) = 4 - 5\delta$$

Multiplying by 4 for the four connected components of $\mathcal{B}$ concludes the proof. $\square$

Now we can use Lemma 20 to bound the length of all band edges.

Lemma 21. We have $\sum_{uv \in \mathcal{B}} \|uv\|_2 \leq 18 - 19\delta + \sum_{(u,v) \in \mathcal{B}'} (\|ut\|_\infty - \|vt\|_\infty)$.

Proof. By Definition 7, we can rewrite $\sum_{uv \in \mathcal{B}} \|uv\|_2 = \|\mathcal{P}_{u',t'}\|_2 + \sum_{(u,v) \in \mathcal{B}'} \|\mathcal{P}_{u,v}\|_2$. For now we focus on $\|\mathcal{P}_{u',t'}\|_2$. If $u' = t'$, then $\|\mathcal{P}_{u',t'}\|_2 = 0$. Otherwise, assume without loss of generality that $u' \in \nwarrow$ and $t' \in \blacksquare \cap \blacktriangleleft$. Then $0 \leq x(t') < x(u')$ and by Lemma 13, $-\delta < y(t')$ since t' is the tip of a band edge. Since $\mathcal{P}_{u',t'}$ is x - and y -monotone, we have

$$\|\mathcal{P}_{u',t'}\|_2 \leq \|u't'\|_x + \|u't'\|_y \leq 1 + (1 + \delta)$$

Next, for $(u,v) \in \mathcal{B}'$, we have $\|\mathcal{P}_{u,v}\|_2 \leq \|uv\|_x + \|uv\|_y$ since $\mathcal{P}_{u,v}$ is x - and y -monotone. Also, $\min(\|uv\|_x, \|uv\|_y) = \|ut\|_\infty - \|vt\|_\infty$ since uv crosses a diagonal boundary of $\nwarrow$ by Lemma 13, yielding

$$\begin{aligned} \sum_{(u,v) \in \mathcal{B}'} \|\mathcal{P}_{u,v}\|_2 &\leq \sum_{(u,v) \in \mathcal{B}'} (\min(\|uv\|_x, \|uv\|_y) + \max(\|uv\|_x, \|uv\|_y)) \\ &\leq \sum_{(u,v) \in \mathcal{B}'} (\|ut\|_\infty - \|vt\|_\infty + \|uv\|_\infty) \\ &\leq 16 - 20\delta + \sum_{(u,v) \in \mathcal{B}'} (\|ut\|_\infty - \|vt\|_\infty) \quad \text{by Lemma 20.} \end{aligned}$$

□

The bounds we have proved for sweep, greedy and band edges can now be combined to prove Theorem 3.

Theorem 3. *DIRTY Algorithm 3 is a local routing algorithm with routing ratio of at most 85.17 for $\vec{\text{Yao}}_4^\infty$ -graphs. This algorithm requires storing a distance between two vertices using $O(\log(|V(G)|))$ bits*

Proof. We bound the total L_2 distance of sweep edges by combining Lemmas 16, 17 and 20:

$$\begin{aligned} \sum_{uv \in \mathcal{S}} \|uv\|_2 &\leq 1 + \sqrt{2}(k - \delta \frac{k(k+1)}{2}) - 2\delta k + \sum_{uv \in \mathcal{G}} \|uv\|_\infty + \sum_{(u,v) \in \mathcal{B}'} \|uv\|_\infty \quad \text{by Lemma 16,} \\ &\leq 1 + \sqrt{2}(k - \delta \frac{k(k+1)}{2}) - 2\delta k + (16 - 8\delta) + (16 - 20\delta) \quad \text{by Lemmas 17 and 20,} \\ &= 33 - 28\delta + \sqrt{2}(k - \delta \frac{k(k+1)}{2}) - 2\delta k. \quad (5.2) \end{aligned}$$

Finally, we can bound the length of the path from s to t' .

$$\begin{aligned}
\|\mathcal{P}_{s,t'}\|_2 &= \sum_{uv \in \mathcal{S}} \|uv\|_2 + \sum_{uv \in \mathcal{G}} \|uv\|_2 + \sum_{uv \in \mathcal{B}} \|uv\|_2 \\
&\leq (33 - 28\delta + \sqrt{2}(k - \delta \frac{k(k+1)}{2}) - 2\delta k) && \text{by (5.2),} \\
&\quad + (12 + 4\sqrt{2} - 8\delta + \sum_{uv \in \mathcal{G}} (\|ut\|_\infty - \|vt\|_\infty)) && \text{by Lemma 18,} \\
&\quad + (18 - 19\delta + \sum_{(u,v) \in \mathcal{B}'} (\|ut\|_\infty - \|vt\|_\infty)) && \text{by Lemma 21,} \\
&\leq 64 - 55\delta + \sqrt{2}(k - \delta \frac{k(k+1)}{2} + 4) - 2\delta k && \text{by Lemma 15.}
\end{aligned}$$

Let us denote the last line of this inequality by $f(\delta)$. Notice that by the definition of step 3 of Algorithm 3, the path $\mathcal{P}_{t',t}$ is identical to the path that Algorithm 3 would output if the first vertex s was chosen to be t' . This means that the entire path from s to t has length at most $f(c) + 2c \cdot f(c) + (2c)^2 \cdot f(c) + \dots = \sum_{j=0}^{\infty} (2c)^j \cdot f(c) = \frac{f(c)}{1-2c}$ because δ is initially set to $c\|st\|_\infty$ until an edge is chosen in step 3. The ratio evaluates to 85.17 when $c = 0.07692307$. Furthermore, $\|st\|_2 \geq \|st\|_\infty = 1$ by assumption. The only memory required to construct the path locally is to store δ for each successive iteration. Hence we have given a local routing algorithm with routing ratio at most 85.17 for Yao_4^∞ -graphs. $\square$

As stated in Theorem 3, the DIRTY Algorithm 3 must store a distance δ . Roughly speaking, the purpose of δ is to manage the size of the green square at each step. However, δ is only rescaled upon reaching a vertex in the green square. In the following corollary, we describe how to avoid storing δ . This results in a memoryless algorithm, albeit at the expense of a larger routing ratio.

Corollary 1. *There exists a memoryless local routing algorithm for Yao_4^∞ -graphs with routing ratio at most 222.20.*

Proof. Consider the memoryless algorithm defined as follows. Let u be the current vertex, and consider $i \in \mathbb{Z}$ such that $(2c)^{i+1} < \|ut\|_\infty \leq (2c)^i$. Set $\delta := c^{i+1}$ and then choose the next vertex according to the Dirty Algorithm 3. Notice this is memoryless and local since only the coordinates of u and t are required to compute δ , as the value of c will be hard-coded into the algorithm.

Now we prove a constant upper bound on the routing ratio of the memoryless routing algorithm. Suppose $\mathcal{P}$ is the path produced from s to t . Suppose κ is the integer such that $(2c)^{\kappa+1} < \|st\|_\infty \leq (2c)^\kappa$. For integers $i \geq \kappa$, let t_i be the first vertex of $\mathcal{P}$ that satisfies $\|t_i t\|_\infty \leq (2c)^i$. Then $t_\kappa = s$ and eventually $t_i = t$ for all sufficiently large i . We notice that for each edge of $\mathcal{P}_{t_i, t_{i+1}}$ chosen according to step 3 of algorithm 3, δ will stay constant. The routing ratio is thus at most

$$\frac{\|\mathcal{P}\|_2}{\|st\|_\infty} = \frac{\sum_{i=\kappa}^{\infty} \|\mathcal{P}_{t_i, t_{i+1}}\|_2}{\|st\|_\infty} \leq \frac{\sum_{i=\kappa}^{\infty} (2c)^i \cdot f(c)}{(2c)^{\kappa+1}} = \frac{f(c)}{2c(1-2c)} \approx 222.20 \text{ when } c := 0.2856$$

□

5.5 Lower Bounds

In the previous section, we designed a local routing algorithm that achieves a constant routing ratio in a sparse directed plane graph containing at most $4|V|$ directed edges. All other known local routing algorithms for directed plane graphs use at least $6|V| - C$ directed edges for some constant C . A main drive in routing is to discover local routing algorithms for even sparser plane graphs requiring fewer than $\alpha|V|$ directed edges for some constant $\alpha < 4$. The following lemma gives a lower bound of 2 on such an α .

Lemma 22. *For any $t \geq 1$ and $n > 1$, there exists a set of n points $V \subset \mathbb{R}^2$ such that any directed t -spanner on V requires at least $2n - 2$ directed edges.*

Proof. Fix a real number $t \geq 1$ and a positive integer n . Choose a real number $c > t$ and define the vertex set $V := \{c^i \mid 0 \leq i < n\}$. Suppose E is a directed edge set such that (V, E) is a t -spanner. We first argue that for all $i \in \{1, \dots, n-1\}$, there exists a $0 \leq j < i$ such that (c^i, c^j) is an edge. Fix $i \in \{1, \dots, n-1\}$. Then there must exist an edge (c^i, c^j) with $j < i$. If no such edge existed, then the shortest path from c^i to c^{i-1} must pass through a larger vertex, meaning that

$$d_G(c^i, c^{i-1}) \geq |c^i - c^{i+1}| + |c^{i+1} - c^{i-1}| = c^{i-1}(2c^2 - c - 1) \geq c^{i-1}(c^2 - c) = c|c^i - c^{i-1}|.$$

This contradicts the assumption that (V, E) is a t -spanner. Therefore c^i must be the source of some edge (c^i, c^j) with $j < i$. By the same argument, c^i must also be the target of some edge (c^k, c^i) with $k < i$. This argument identifies two directed edges in E for each $i \in \{1, \dots, n-1\}$, hence $|E|$ must have size at least $2(n-1)$. Notice that we can slightly perturb the locations of the vertices to yield a point set in general position. □

Lemma 23. *Let $\epsilon > 0$. There exists a $\vec{\text{Yao}}_4^\infty$ -graph with spanning ratio at least $5 + \sqrt{2} - \epsilon$.*

Proof. Let $\epsilon > 0$ be arbitrarily small. We will create a graph with spanning ratio at least $5 + \sqrt{2} - \epsilon$, shown in Figure 5.7. We let $\delta > 0$ and choose its exact value later in the proof. Let $s := (0, -1) + \delta(3, 1)$, $a := (-1, -1) + \delta(6, 3)$, $b := (-1, 0) + \delta(7, -1)$, $c := (-1, 1) + \delta(8, -10)$, $d := (0, 1) + \delta(-1, -11)$, $e := (1, 1) + \delta(-13, -12)$, $f := (2, 1) + \delta(-13, -13)$, $g := (2, -1) + \delta(-12, +2)$, and $t = (0, 0)$. Let G be the $\vec{\text{Yao}}_4^\infty$ -graph with vertices $\{s, a, b, c, d, e, f, g, t\}$. Then excluding the outgoing edges from t , G contains the following edges: $sa, sg, as, ab, ba, bs, bc, cb, cd, db, dc, de, ed, et, ef, fe, fs, fg, gf, gs$. Since the only incoming edge to t is et , then the shortest

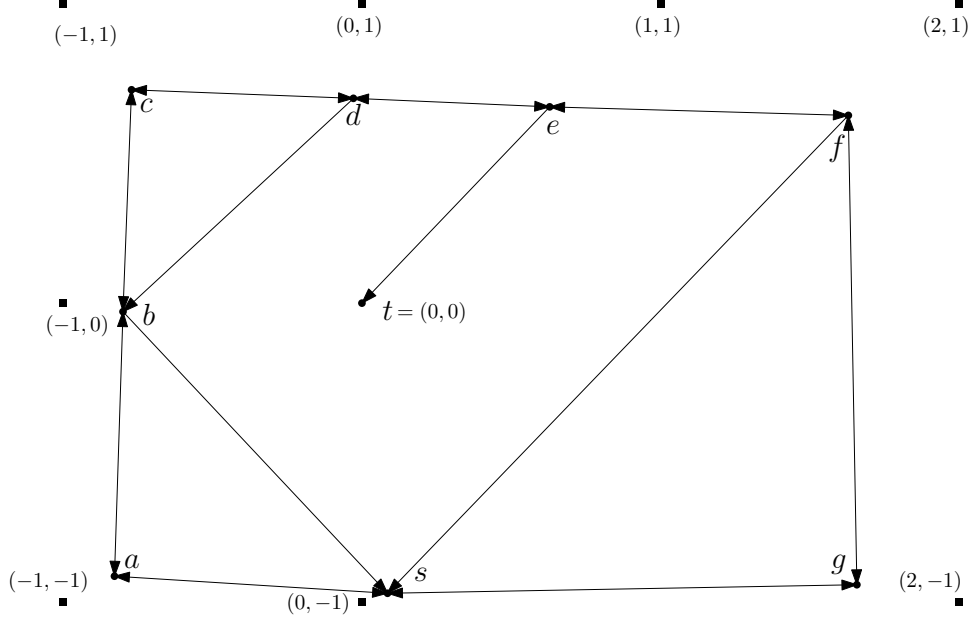


Fig. 5.7: Pictured is the $\vec{\text{Yao}}_4^\infty$ -graph with vertices $\{s, a, b, c, d, e, f, g, t\}$. The shortest path from s to t is (s, a, b, c, d, e, t) and its length can be made arbitrarily close to $5 + \sqrt{2}$. The outgoing edges from t are omitted.

path from s to t must pass by e . The shortest path from s to e is (s, a, b, c, d, e) since

$$\begin{aligned}
& \|sg\|_2 + \|gf\|_2 + \|fe\|_2 \\
& \geq \|sg\|_\infty + \|gf\|_\infty + \|fe\|_\infty \\
& = (2 - 15\delta) + (2 - 15\delta) + 1 \\
& \geq (1 - 1\delta) + (1 - 3\delta) + (1 - 8\delta) + (1 - 8\delta) + (1 - 11\delta) \\
& = \|sa\|_1 + \|ab\|_1 + \|bc\|_1 + \|cd\|_1 + \|de\|_1 \\
& \geq \|sa\|_2 + \|ab\|_2 + \|bc\|_2 + \|cd\|_2 + \|de\|_2 \\
& \geq \|sa\|_\infty + \|ab\|_\infty + \|bc\|_\infty + \|cd\|_\infty + \|de\|_\infty \\
& = (1 - 3\delta) + (1 - 4\delta) + (1 - 9\delta) + (1 - 9\delta) + (1 - 12\delta) = 5 - 37\delta
\end{aligned}$$

Next, we have $\|et\|_2 \geq \sqrt{2} \min(\|et\|_x, \|et\|_y) = \sqrt{2}(1 - 13\delta)$, therefore the shortest path from s to t in G has length at least $5 + \sqrt{2} - \delta(37 + 13\sqrt{2})$. Finally, $\|st\|_2 \leq \sqrt{(1 - \delta)^2 + (3\delta)^2} \leq 1$ when $\delta \leq \frac{1}{5}$. Therefore choosing $\delta < \frac{\epsilon}{37 + 13\sqrt{2}}$ means that the spanning ratio of G is at least $5 + \sqrt{2} - \epsilon$. $\square$

Lemma 24. *Let $\epsilon > 0$. Then any local routing algorithm for $\vec{\text{Yao}}_4^\infty$ -graphs has routing ratio at least $7 + \sqrt{2} - \epsilon$ in the worst case.*

Proof. Let $\epsilon > 0$ be arbitrarily small and let A be a local routing algorithm. We will construct three graphs G_W, G_N, G_E such that A has a routing ratio of at least $7 + \sqrt{2} - \epsilon$ in at least one

of $\{G_W, G_N, G_E\}$ when routing from s to $t := (0, 0)$. The three graphs are shown in Figure 5.9. Let $\delta > 0$ be arbitrarily small.

We will start by defining the *zip-zag paths*, illustrated in Figure 5.8, which will act as sub-graphs. Define the two segments $s_1 := (15\delta - 1, \delta)(0, \delta - \delta^2)$ and $s_2 := (15\delta - 1, -\delta)(0, \delta^2 - \delta)$. Let $p_1 := (15\delta - 1, \delta)$, and $p_2 \in s_1$ such that $x(p_2) = 13\delta - 1$. Let $q_1 \in s_2$ such that p_2q_1 has slope 1. Then for $i \geq 3$ define $p_i \in s_1$ and $q_{i-1} \in s_2$ such that triangle $p_{i-1}p_iq_{i-1}$ is similar to triangle $p_{i-2}p_{i-1}q_{i-2}$. Let k be the greatest index such that $x(p_k) < 0$. Then define $\text{WestZag} := \{p_1, p_2, \dots, p_k, q_1, q_2, \dots, q_{k-1}, t\}$.

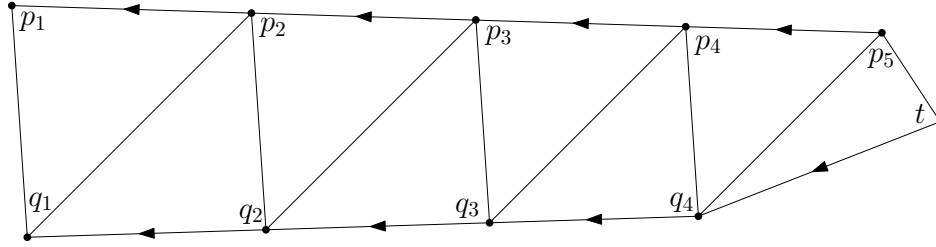


Fig. 5.8: The shortest path from p_1 to t in the $\vec{\text{Yao}}_4^\infty$ -graph of WestZag has length arbitrarily close to $1 + \sqrt{2}$. Note, all $p_1, p_2, \dots, p_k$ lie on segment s_1 , and all $q_1, q_2, \dots, q_{k-1}$ lie on segment s_2 .

The shortest path from p_1 to t in the $\vec{\text{Yao}}_4^\infty$ -graph of WestZag is $p_1, q_1, p_2, q_2, \dots, q_{k-1}, p_k, t$, and it is x -monotone. We will now express the length of this path in terms of δ . Firstly, we have $\|p_1q_1\|_2 \geq \|p_1q_1\|_y \geq 2(\delta - \delta^2)$ and $\|q_1p_2\|_2 = \sqrt{2}\|q_1p_2\|_y \geq 2\sqrt{2}(\delta - \delta^2)$. By similar triangles, for all $1 < i < k$ we have

$$\frac{\|p_1q_1\|_2 + \|q_1p_2\|_2}{x(p_2) - x(p_1)} = \frac{\|p_iq_i\|_2 + \|q_ip_{i+1}\|_2}{x(p_{i+1}) - x(p_i)}.$$

Combining this with the observation that $-2\delta < x(p_k)$, then the shortest path from p_1 to t in WestZag has length at least

$$\begin{aligned} \sum_{i=1}^{k-1} (\|p_iq_i\|_2 + \|q_ip_{i+1}\|_2) &= \frac{\|p_1q_1\|_2 + \|q_1p_2\|_2}{x(p_2) - x(p_1)} \sum_{i=1}^{k-1} (x(p_{i+1}) - x(p_i)) \\ &\geq \frac{2(\delta - \delta^2)(1 + \sqrt{2})}{2\delta} (x(p_k) - x(p_1)) \\ &\geq \frac{2(\delta - \delta^2)(1 + \sqrt{2})}{2\delta} (-2\delta - (15\delta - 1)) \\ &= (1 + \sqrt{2})(1 - \delta)(1 - 17\delta) \\ &\geq 1 + \sqrt{2} - 44\delta \end{aligned}$$

Additionally, the shortest path from q_1 to t in WestZag has length at least $1 + \sqrt{2} - 47\delta$ since $\|p_1q_1\|_2 \leq \sqrt{2}\|p_1q_1\|_\infty \leq 3\delta$. Analogously, we define NorthZag to be WestZag reflected about

the line $y = -x$. Similarly, define EastZag to be WestZag reflected about the vertical line through t .

Now, define the points $s := (0, -1) + \delta(1, 1)$, $a := (-1, -1) + \delta(5, 2)$, $b := (-1, 0) + \delta(6, -2)$, $c := (-1, 1) + \delta(7, -10)$, $d := (0, 1) + \delta(-1, -11)$, $e := (1, 1) + \delta(-4, -6)$, $f := (1, 0) + \delta(-3, -1)$, $g := (1, -1) + \delta(-2, 3)$, $c' := (-1, 1) + \delta(10, -8)$, $d' := (0, 1) + \delta(1, -7)$, $e' := (1, 1) + \delta(-14, -12)$. Note that $\|st\|_2 = \sqrt{(1-\delta)^2 + \delta^2} \leq 1$ since $0 < \delta < 1$. We define vertex sets $V(G_W) := \text{WestZag} \cup \{s, a, b, c', d', e, f, g\}$, $V(G_N) := \text{NorthZag} \cup \{s, a, b, c, d', e, f, g\}$, $V(G_E) := \text{EastZag} \cup \{s, a, b, c, d, e', f, g\}$.

This construction guarantees that the shortest path from any vertex not in a *zip-zag path* to t must pass through a *zip-zag path*. The decision tree for which graph to provide to A is

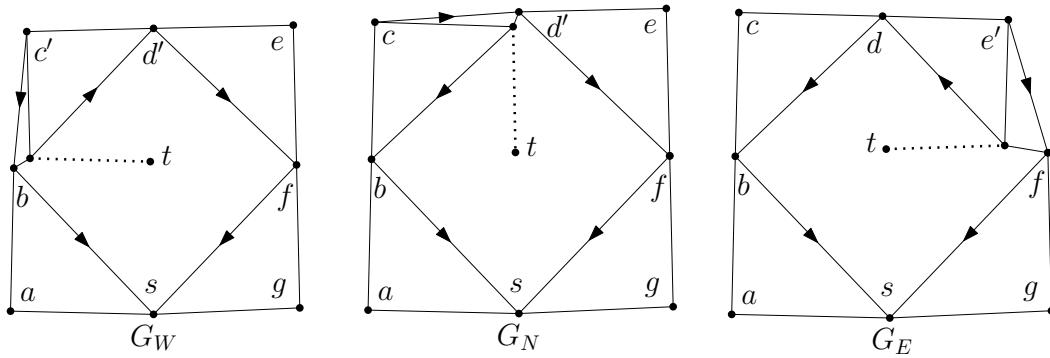


Fig. 5.9: The graphs are G_W, G_N, G_E from left to right. When routing from s to t , the routing ratio of A is at least $7 + \sqrt{2} - \epsilon$ in one of the graphs. The dotted lines represent the zig-zag paths to t . All outgoing edges from the zig-zag paths are omitted.

illustrated in Figure 5.10. One can verify that the decision tree represents a valid adversarial strategy since the collections of graphs at each node contain identical neighbourhoods at all previously visited vertices. Finally, if we exclude the zig-zag paths, then each vertex is within a

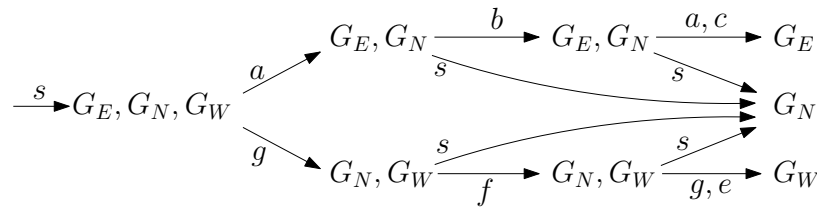


Fig. 5.10: The decision tree describing the adversarial strategy for which graph to provide to A . For example, suppose A first visits the vertices s, a, b . Notice that the neighbourhoods around vertices s, a, b are all identical between graphs G_E, G_N . Then if A chooses c for the next vertex, then A would have a routing ratio of at least $7 + \sqrt{2} - \epsilon$ in G_E .

distance of 20δ from its nearest integer coordinate. Since the shortest path to t in each case uses at least six edges with length at least $1 - 2(20\delta)$ before passing through q_1 , then the total path length in any case is at least $6(1 - 40\delta) + 1 + \sqrt{2} - 47\delta = 7 + \sqrt{2} - 287\delta$. Setting $\delta < \frac{\epsilon}{287}$ completes the proof. $\square$

Lemma 25. *Let $\epsilon > 0$. There exists a $\vec{Yao}_4$ -graph with spanning ratio at least $7 - \sqrt{3} + \sqrt{5 - 2\sqrt{3}} - \epsilon$.*

Proof. Let $\epsilon > 0$ be arbitrarily small. We will create a graph with spanning ratio at least $7 - \sqrt{3} + \sqrt{5 - 2\sqrt{3}} - \epsilon$, shown in Figure 5.11.

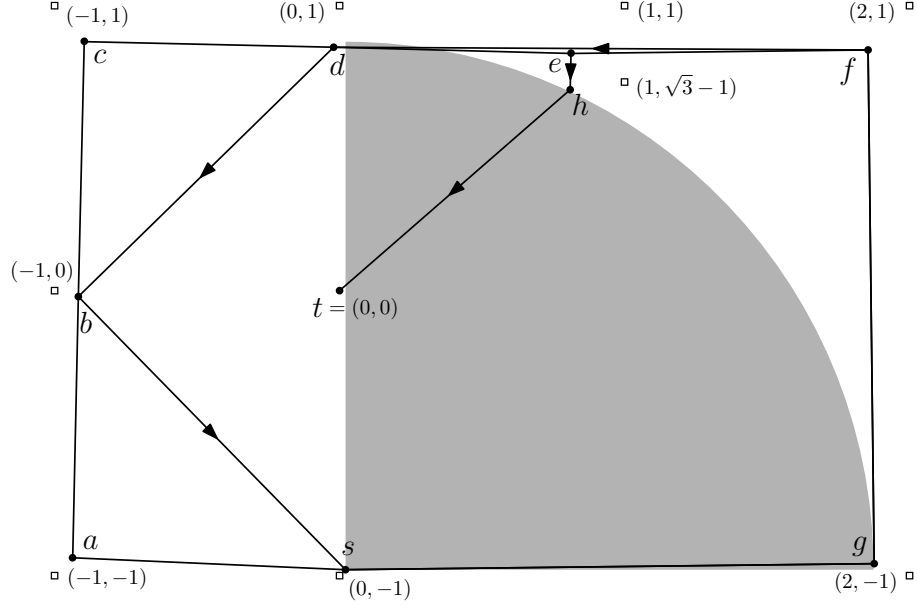


Fig. 5.11: Pictured is the $\vec{Yao}_4$ -graph with vertices $\{s, a, b, c, d, e, f, g, h, t\}$. The shortest path from s to t must pass by h , and its length can be made arbitrarily close to $7 - \sqrt{3} + \sqrt{5 - 2\sqrt{3}} \approx 6.51$. Several edges are omitted for clarity.

Let $\delta > 0$ be arbitrarily small. Define $s := (0, -1) + \delta(1, 1)$, $a := (-1, -1) + \delta(3, 3)$, $b := (-1, 0) + \delta(4, -1)$, $c := (-1, 1) + \delta(5, -6)$, $d := (0, 1) + \delta(-1, -7)$, $e := (1, 1) + \delta(-9, -8)$, $f := (2, 1) + \delta(-7, -7.5)$, $g := (2, -1) + \delta(-6, 2)$, and $t = (0, 0)$.

Next, we define a point in the region below and to the left of e . More precisely, start by defining p_1 to be the north-most point satisfying $x(p_1) = x(e)$ and $\|sp_1\|_2 = \|sg\|_2$. Next, let p_2 be the east-most point such that $\|dp_2\|_2 = \|de\|_2$ and p_1p_2 is orthogonal to sp_1 . Finally let $p_3 := (x(p_1), y(p_2))$. Define the convex combination $h := (1 - 2\delta)p_1 + \delta p_2 + \delta p_3$. Note that this construction guarantees that h is in the triangle $p_1p_2p_3$. Let G be the $\vec{Yao}_4$ -graph with vertices $\{s, a, b, c, d, e, f, g, h, t\}$. By construction, ignoring outgoing edges from t , the edges of G are $sa, sg, as, ab, bs, ba, bc, cb, cd, db, dc, de, ed, eh, ef, eg, fe, fd, fg, gs, gf, ht, hd, he, hg$. Note that the only incoming edge to t is ht , and that the only incoming edge to h is eh . Also, for any distinct $u, v \in \{s, a, b, c, d, e, f, g\}$, the coordinates of u and v differ by at most 9δ from their nearest integer grid points, so the edge uv has length at least $1 - 18\delta$. Furthermore, sg and gf

both have length at least $2 - 18\delta$. It remains to bound the lengths $\|eh\|_2$ and $\|ht\|_2$. We have

$$\begin{aligned} \|sp_1\|_2 = \|sg\|_2 &\implies (1 - 10\delta)^2 + (y(p_1) - \delta + 1)^2 = (2 - 7\delta)^2 + \delta^2 \\ &\implies y(p_1) = \sqrt{3 + 6\delta - 50\delta^2} - 1 + \delta \\ &\implies y(p_1) \leq \sqrt{3} - 1 + 4\sqrt{\delta} \text{ and } y(p_1) \geq \sqrt{3} - 1 - 8\delta \\ &\implies |y(p_1) - (\sqrt{3} - 1)| \leq 8\sqrt{\delta} \end{aligned}$$

Next, the x - and y - coordinates of p_1, p_2 and p_3 are in $[0, 1]$, so by definition, we have

$$\|h - p_1\|_\infty = \|-2\delta y(p_1) + \delta y(p_2) + \delta y(p_3)\|_\infty \leq 2\delta$$

This allows us to bound the distance from h to $(1, \sqrt{3} - 1)$:

$$\|h - (1, \sqrt{3} - 1)\|_\infty \leq \|h - p_1\|_\infty + \|p_1 - (1, \sqrt{3} - 1)\|_\infty \leq 2\delta + \max(9\delta, 8\sqrt{\delta}) \leq 11\sqrt{\delta}$$

Now we can lower bound the edge length $\|ht\|_2$ and $\|eh\|_\infty$ as follows:

$$\begin{aligned} \|ht\|_2 &\geq \|(1, \sqrt{3} - 1)t\|_2 - \|h(1, \sqrt{3} - 1)\|_2 \geq \sqrt{5 - 2\sqrt{3}} - 11\sqrt{2\delta} \\ \|eh\|_\infty &\geq \|(1, 1) - (1, \sqrt{3} - 1)\|_\infty - \|e - (1, 1)\|_\infty - \|h - (1, \sqrt{3} - 1)\|_\infty \geq 2 - \sqrt{3} - 9\delta - 11\sqrt{\delta} \end{aligned}$$

The shortest path from s to t in G will follow either path $P_1 := (s, a, b, c, d, e)$ or $P_2 := (s, g, f, e)$, followed by the edges eh and ht . We have

$$\begin{aligned} d_G(s, t) &\geq \min(\|P_1\|_2, \|P_2\|_2) + \|eh\|_2 + \|ht\|_2 \\ &\geq 5(1 - 18\delta) + (2 - \sqrt{3} - 9\delta - 11\sqrt{\delta}) + (\sqrt{5 - 2\sqrt{3}} - 11\sqrt{2\delta}) \\ &\geq 7 - \sqrt{3} + \sqrt{5 - 2\sqrt{3}} - 126\sqrt{\delta}. \end{aligned}$$

Setting $\delta < (\epsilon/126)^2$ completes the proof since $\|st\|_2 = \sqrt{(1 - \delta)^2 + \delta^2} \leq 1$. $\square$

Lemma 26. *Let $\epsilon > 0$. Then any local routing algorithm for $\vec{\text{Yao}}_4$ -graphs has routing ratio at least $7 + \sqrt{2} - \epsilon$ in the worst case.*

Proof. Let $\epsilon > 0$ be arbitrarily small. We will construct five $\vec{\text{Yao}}_4$ -graphs G_1, G_2, G_3, G_4, G_5 such that any local routing algorithm has a routing ratio of at least $7 + \sqrt{2} - \epsilon$ in at least one of the five graphs. The five graphs are shown in Figure 5.12. Let $\delta < 0$ be arbitrarily small. Define the points $t := (0, 0)$, $s := (0, -1) + \delta(1, 1)$, $a := (-1, -1) + \delta(3, 2)$, $b := (-1, 0) + \delta(4, -1)$, $c := (-1, 1) + \delta(5, -6)$, $d := (0, 1) + \delta(-1, -7)$, $e := (1, 1) + \delta(-9, -8)$, $f := (1, 0) + \delta(-10, 1)$, $g := (1 - 11\delta, \sqrt{22\delta + 2\delta - 1})$, $a' := (8\delta - 1, \sqrt{10\delta + 2\delta - 1})$, $b' := (-1, 0) + \delta(7, 1)$, $c' := (-1, 1) + \delta(6, -5)$, $d' := (0, 1) + \delta(1, -4)$, $e' := (1, 1) + \delta(-2, -3)$, $f' := (1, 0) + \delta(-1, -1)$, $g' := (1, -1) + \delta(0, 3)$. We have $\|st\|_2 = \sqrt{(1 - \delta)^2 + \delta^2} \leq 1$ since $0 < \delta < 1$. Then define the graphs by their point sets: $V(G_1) := \{s, t, a, b, c, d, e, f, g, g'\}$, $V(G_2) := \{s, t, a, b, c, d, e, f, f', g'\}$, $V(G_3) := \{s, t, a, b, c, d, d', e', f', g'\}$, $V(G_4) := \{s, t, a, b, b', c', d', e', f', g'\}$, $V(G_5) := \{s, t, a, a', b', c', d', e', f', g'\}$.

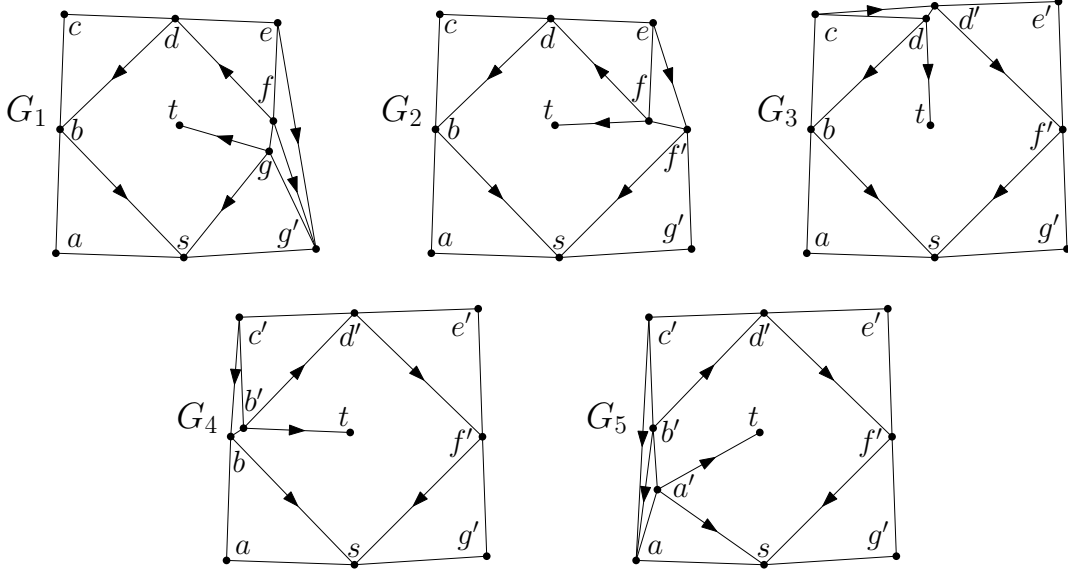


Fig. 5.12: Any local routing algorithm for $\vec{\text{Yao}}_4$ has a routing ratio of at least $7 + \sqrt{2} - \epsilon$ in one of the five graphs. The outgoing edges from t are omitted.

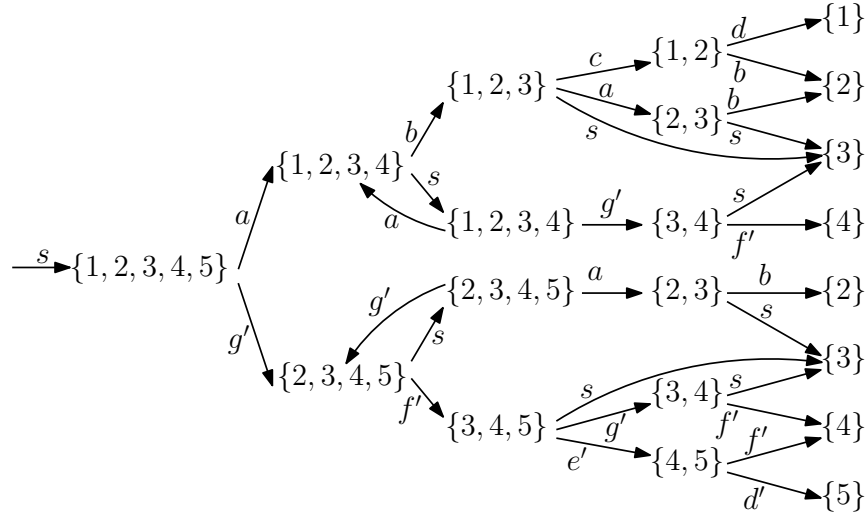


Fig. 5.13: The adversarial decision tree for which graph to provide to any algorithm.

Notice that the only incoming edges to t in the five graphs are $gt, ft, dt, b't, a't$, respectively. The decision tree for which graph to provide to any algorithm is illustrated in Figure 5.13.

By construction, each coordinate of each vertex differs by at most $11\sqrt{\delta}$ from its nearest integer grid point. This implies that any edge connecting two different letters must have length of at least $1 - 22\sqrt{\delta}$. Moreover, eg and $c'a$ have length at least $2 - 22\sqrt{\delta}$. Furthermore, the nearly diagonal edges, $bs, f's, a't, gt$, have length at least $\sqrt{2}(1 - 22\sqrt{\delta})$. In any case, we append the shortest path to t from a leaf of the decision tree to get a routing ratio of at least $7(1 - 22\sqrt{\delta}) + \sqrt{2}(1 - 22\sqrt{\delta}) \geq 7 + \sqrt{2} - 186\sqrt{\delta}$ in the worst case. Setting $\delta < (\epsilon/186)^2$

guarantees that the routing ratio is at least $7 + \sqrt{2} - \epsilon$.

□

Chapter 6

The Spanning Ratio of the Directed $\vec{\Theta}_6$ -Graph is 5

Prosenjit Bose, Jean-Lou De Carufel, Darryl Hill, John Stuart

6.1 Abstract

Given a finite set $P \subset \mathbb{R}^2$, the directed Theta-6 graph, denoted $\vec{\Theta}_6(P)$, is a well-studied geometric graph, whose underlying graph is the union of two Delaunay triangulations where the empty region is an equilateral triangle. For two points $u, v \in P$, we can define ∇_u^v to be the unique equilateral triangle whose sides have slopes $-\sqrt{3}$, 0 , $\sqrt{3}$ with u on a corner and v on the opposite side. Then the $\vec{\Theta}_6(P)$ -graph has an edge from u to v if and only if the interior of ∇_u^v contains no vertices of P . It was previously shown that the spanning ratio of the $\vec{\Theta}_6(P)$ -graph is between 4 and 7 in the worst case (Akitaya, Biniaz, and Bose *Comput. Geom.*, 105-106:101881, 2022). We close this gap by showing a tight spanning ratio of 5. This is the first tight bound proven for the spanning ratio of any $\vec{\Theta}_k(P)$ -graph. Our lower bound models a long path by mapping it to a converging series. Our upper bound proof uses techniques novel to the area of spanners. We use linear programming to prove that among several candidate paths, there exists a path satisfying our bound.

6.2 Introduction

Studying the distance-preserving properties of particular families of graphs in the plane has been an active area of research for decades. This area has numerous applications, such as motion planning, or wireless routing. We will focus on geometric graphs where each vertex is a point in the plane, and each edge is weighted by the Euclidean distance between its endpoints. Given a geometric graph G , the spanning ratio of a subgraph H is a measure of how well distances of G are preserved. In particular, a subgraph H is called a c -*spanner* of G if for every edge uv in G , the shortest path in H from u to v is at most c times the length of the edge uv in G [39]. The smallest such c is referred to as the *spanning ratio* of H . Finding a tight bound on the spanning ratio of different geometric graphs is a fundamental problem in computational geometry. A particular graph of interest is the *Delaunay triangulation*. It has an edge uv when there exists a disk with u and v on its boundary and no vertices in its interior. The spanning ratio of the Delaunay triangulation has been studied for nearly 40 years, and the first constant

upper bound proven was $\frac{1+\sqrt{5}}{2}\pi \approx 5.083$ [31]. Currently, the spanning ratio of the Delaunay triangulation is only known to be between 1.593 [45] and 1.998 [44], and finding the exact value remains a major open problem in computational geometry [39]. The first plane geometric graph proven to have a constant spanning ratio is the $\square$ -Delaunay graph, whose empty region is an oriented square [26]. The proof was then generalized to the $\triangle$ -Delaunay graph, whose empty region is an oriented equilateral triangle [27].

Delaunay triangulations have many desirable properties. A Delaunay triangulation is a plane graph, meaning that it contains roughly $3|V|$ undirected edges, therefore the average vertex degree is approximately 6. However, it is possible for a vertex to have arbitrarily large degree. In many applications, such as signal transmission, such a large degree may be problematic. Instead, we focus on a closely-related geometric graph with bounded outdegree of 6.

Given a finite set $P \subset \mathbb{R}^2$ and integer $k > 1$, we can construct a geometric graph on P with at most k outgoing edges per vertex as follows. For each vertex $u \in P$, partition the plane into k equiangular cones about u . In particular, the cone boundaries form an angle with the horizontal which is a multiple of $2\pi/k$ radians. Then, add an edge from u to the *nearest* vertex in each cone about u . Repeat this process for each vertex of P . If *nearest* is measured by the Euclidean distance, then the resulting graph is known as a Yao- k graph, first studied by Yao [46]. The directed Theta- k graph, denoted $\vec{\Theta}_k$, is defined similarly, except that *nearest* is measured using a distance function whose unit disk is a regular k -gon. In particular, the $\vec{\Theta}_k$ -graph contains an edge from u to v in cone C when the orthogonal projection of uv onto the cone bisector is minimal among all vertices v in cone C . We refer to the undirected version as the Θ_k -graph. Since being introduced by Clarkson [28] and Keil and Gutwin [33], Θ_k -graphs have been studied extensively. For $k \geq 4$, the Θ_k -graph is known to be a spanner of the complete graph [3, 13, 14, 28, 33, 41]. On the other hand, the $\vec{\Theta}_k$ -graph is known to be a constant spanner for $k \geq 6$ [1, 14, 41] and also for $k = 4$ [13]. Various spanning ratio upper and lower bounds are plotted in Figure 6.1.

Despite extensive research, tight bounds on the spanning ratio are only known for the Θ_{4k+2} -graph when $k \geq 1$. In general, results claiming tight bounds on the spanning ratio of any geometric graph are rare. To the best of our knowledge, the only other geometric graphs with known tight spanning ratios are the three generalized Delaunay graphs whose empty regions are either a triangle, parallelogram, or hexagon. See Table 6.1.

Reference	Graph	Spanning Ratio	Notes
[14]	Θ_{4k+2} ($k \geq 1$)	$1 + 2 \sin(\theta/2)$	Also for constrained version [24]
[27]	$\triangle$ -Delaunay	2	Also for affine transformations [36, 42]
[9]	$\square$ -Delaunay	$\sqrt{4 + 2\sqrt{2}} \approx 2.613$	Also for affine transformations [15, 43]
[40]	$\hexagon$ -Delaunay	2	

Table 6.1: Other geometric graphs with known tight spanning ratios. Notice these graphs have bidirectional edges.

The $\vec{\Theta}_6$ -graph has a special symmetry. For two points $u, v \in P$, we can define ∇_u^v to be the

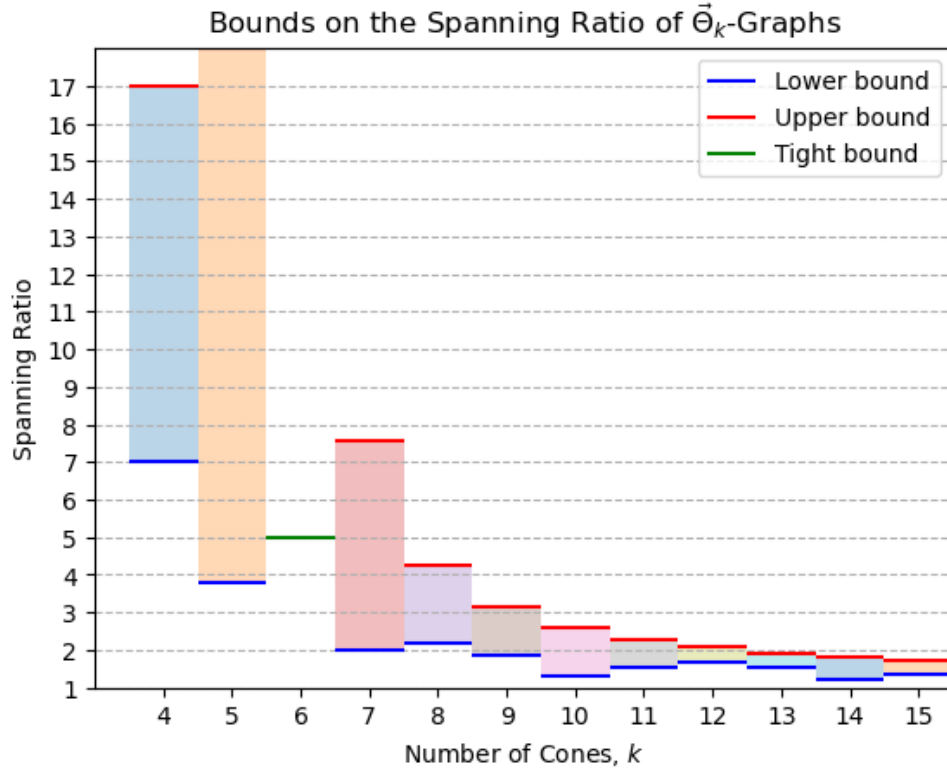


Fig. 6.1: We plot spanning ratio upper and lower bounds of the $\vec{\Theta}_k$ -graph for various k [13, 14, 23, 41].

unique equilateral triangle whose sides have slopes $-\sqrt{3}$, 0 , $\sqrt{3}$ with u on a corner and v on the opposite side. Then the $\vec{\Theta}_6$ -graph contains an edge from u to v exactly when the interior of ∇_u^v contains no vertices of P . Readers familiar with Delaunay triangulations may notice that the $\vec{\Theta}_6$ -graph is essentially two rotated copies of a Δ -Delaunay graph. The Δ -Delaunay graph, often referred to as the TD-Delaunay graph, has a spanning ratio of 2 in the worst case, hence the Theta-6 graph inherits this upper bound. On the other hand, the $\vec{\Theta}_6$ -graph does not preserve distances nearly as well as its undirected counterpart. The first bounds on the spanning ratio of the $\vec{\Theta}_6$ -graph were given by Akitaya, Biniarz, and Bose [1]. For any $\epsilon > 0$, they construct a $\vec{\Theta}_6$ -graph whose spanning ratio is at least $4 - \epsilon$. Then, they show that constructing a path in an arbitrary $\vec{\Theta}_6$ -graph is not as straightforward as selecting the outgoing edge in the cone containing the destination. This technique is known as *greedy routing*, and it produces efficient paths in $\vec{\Theta}_k$ -graphs only when $k \geq 7$ [41]. Greedy routing in $\vec{\Theta}_6$ -graphs can lead to a spiraling path of unbounded length. Instead, the authors of [1] give an upper bound of 7 for the spanning ratio of $\vec{\Theta}_6$ -graphs using the following technique. Given an edge from a to b , they show that the *greedy* path from b to a has length at most 6 times the length of the edge from a to b . Then, for any two vertices s, t , it suffices to find an intermediate vertex x such that there are short paths from s to x and t to x . Indeed, a path from s to t can be formed by concatenating the path from s to x with the greedy paths corresponding to each edge on the path from t to x .

The authors leave it as an open problem to close the gap between 4 and 7. As the title of our paper suggests, we close this gap by showing that the $\vec{\Theta}_6$ -graph has a tight spanning ratio of 5.

Our proof uses elements from [1] along with several novel techniques in the area of spanners. Most notably, we use linear programming to prove the existence of a short path. At a high level, our proof is an induction on the distance between points, and it proceeds as follows. We define a special region R between s and t , and show that if there exists a vertex $x \in R$, then induction provides two paths, from s to x and from x to t , which can be concatenated to form a sufficiently short path from s to t . Then, for the rest of the proof, we assume that no vertices are in the interior of R . Importantly, we are able to show that any edge uv on the greedy path that crosses R must pay a “toll”. More precisely, v is guaranteed to be at least twice as close to t as u . We leverage this property by applying induction after a candidate path crosses R . The difficulty is that there are multiple candidate paths, and determining which path is sufficiently short is non-trivial. Our solution is to assume that there is no sufficiently short candidate path. This allows us to prove inequalities corresponding to each candidate path, then we can use linear programming to show that the system is infeasible¹. In essence, assuming that there is no short path leads to a contradiction.

6.3 Preliminaries

The geometric graphs we study are weighted, directed graphs whose vertex sets contain points in the plane. We denote the directed edge from vertex u to vertex v by uv , and with a slight abuse of notation, when $p, q \in \mathbb{R}^2$, we denote the segment from p to q as pq . In general, for an edge uv and a norm $\|\cdot\|$, the value $\|uv\|$ is exactly the factor by which the unit disk of $\|\cdot\|$ must be scaled such that the unique translate with u in the center has v on the boundary. For example, $\|\cdot\|_2$ refers to the Euclidean norm whose unit disk is a circle with radius 1. We also let $\|uv\|_x$ (resp. $\|uv\|_y$) denote the absolute difference in x -coordinate (resp. y -coordinate) from u and v . The *Manhattan norm*, $\|uv\|_1 := \|uv\|_x + \|uv\|_y$, has a tipped square with side length $\sqrt{2}$ as its unit disk. For an edge uv in a geometric graph, we define its weight to be $\|uv\|_2$, the usual distance from u to v . Let $\|\cdot\|_{\hexagon}$ denote the *hexagonal norm* whose unit disk is a regular hexagon with east and west vertices equal to $(1, 0)$ and $(-1, 0)$, respectively. On the other hand, let $\|\cdot\|_{\diamond}$ denote the *tipped hexagonal norm* whose unit disk is a tipped regular hexagon with north and south vertices equal to $(0, \frac{2}{\sqrt{3}})$ and $(0, \frac{-2}{\sqrt{3}})$, respectively. See Figure 6.2. Notice that we always have $\|uv\|_{\diamond} \leq \|uv\|_2 \leq \|uv\|_{\hexagon}$, and equality holds exactly when uv has slope $-\sqrt{3}$, 0, or $\sqrt{3}$.

In a geometric graph G , we define the length of a path to be the sum of the weights of its edges. For any norm $\|\cdot\|$ and path $\mathcal{P}$, we define $\|\mathcal{P}\| := \sum_{uv \text{ on } \mathcal{P}} \|uv\|$. If v is a vertex on a path $\mathcal{P}$, then we define the *truncated path* $\mathcal{P}_v$ as the sub-path of $\mathcal{P}$ which ends at v . For two vertices u, v in a geometric graph G , we let $d_G(u, v)$ denote the length of the shortest path in G from u to v . Recall that all edges are weighted by the Euclidean norm. Then for a real number $c \geq 1$, G is said to be a c -spanner if for all points u, v in G , we have $d_G(u, v) \leq c\|uv\|_2$. The

¹Infeasibility of all linear programs was verified using [Mathematica](#). Since our coefficients are rational, then the computations were carried out in exact arithmetic.

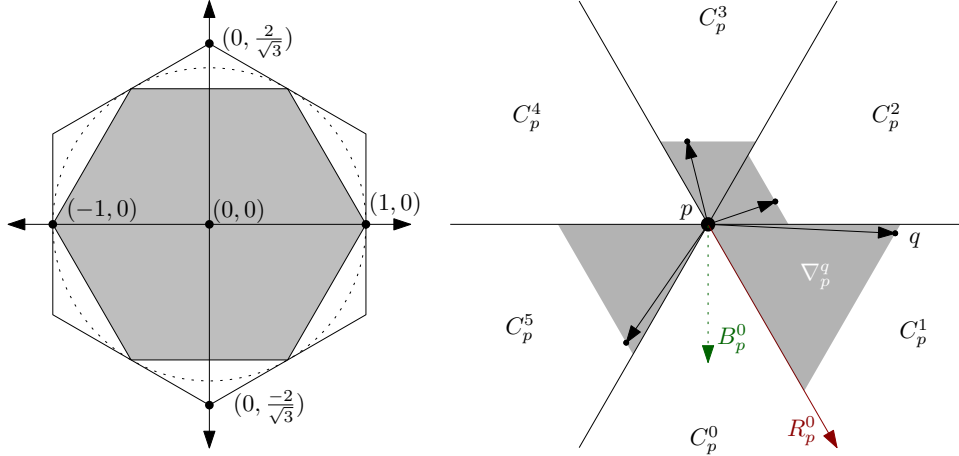


Fig. 6.2: Left: The dotted circle represents the unit disk in the standard Euclidean norm ($\|\cdot\|_2$). The shaded inner hexagon is the unit disk in the $\|\cdot\|_{\hexagon}$ -norm. The outer tipped hexagon is the unit disk in the $\|\cdot\|_{\diamond}$ -norm. Right: The six cones about p are labelled in counter-clockwise order from C_p^0 below p . The dotted green bisector is B_p^0 and the red ray is R_p^0 . The empty shaded triangle in C_p^1 is ∇_p^q . Each shaded region is empty.

spanning ratio of a class of graphs $\mathcal{G}$ is the infimum of $\{c \in \mathbb{R} \mid \forall G \in \mathcal{G}, G \text{ is a } c\text{-spanner}\}$.

For any point p , consider the three lines passing through p with slopes $-\sqrt{3}$, 0 , and $\sqrt{3}$. These lines partition the plane into six cones about p , which we label counter-clockwise by C_p^i for $i \in \mathbb{Z}_6$ starting with C_p^0 directly below p . Here, $\mathbb{Z}_6$ refers to the integers modulo 6. See Figure 6.2. For each cone C_p^i , we denote its bisector by B_p^i . Throughout this paper, we adopt the convention that cones are closed sets, otherwise we write $\text{int}(R)$ to denote the interior of region R . For $i \in \mathbb{Z}_6$, we define the ray $R_p^i := C_p^i \cap C_p^{i+1}$. For points $p, q \in \mathbb{R}^2$ with $q \in C_p^i$, we let ∇_p^q denote the triangular region $\{r \in C_p^i \mid \|pr\|_{\hexagon} \leq \|pq\|_{\hexagon}\}$. For any equilateral triangle T , we let $\|T\|$ denote the side length of T . Notice that $\|\nabla_p^q\| = \|pq\|_{\hexagon}$. For any finite point set $P \subset \mathbb{R}^2$, we make the *general position assumption* that no two vertices lie on a line with slope $-\sqrt{3}$, 0 , or $\sqrt{3}$. The directed Theta-6 graph of P , denoted $\vec{\Theta}_6(P)$, contains an edge from vertex u to vertex v exactly when $\text{int}(\nabla_u^v) \cap P = \emptyset$. Intuitively, a vertex has an outgoing edge to the nearest vertex in each of its six cones, measured using the $\|\cdot\|_{\hexagon}$ -norm. For any vertices $u, v \in P$, we let $\pi(u, v)$ denote the *greedy path* from u to v . In particular, every edge ab of $\pi(u, v)$ satisfies $b \in C_a^i \iff v \in C_a^i$. If w is a vertex along the path $\pi(u, v)$, then the sub-path terminating at w is denoted by $\pi(u, v)_w$. Note that if $w \neq v$, then $\pi(u, v)_w$ is not necessarily the same as $\pi(u, w)$.

6.4 Lower Bound

Lemma 1. *For any $\epsilon > 0$, there exists a set of points P such that the $\vec{\Theta}_6(P)$ -graph has spanning ratio at least $5 - \epsilon$.*

Proof. Let $\epsilon > 0$ be arbitrarily small. We will construct a $\vec{\Theta}_6$ -graph with spanning ratio at least $5 - \epsilon$, shown in Figure 6.3. For the following construction, set $0 < \delta < \frac{\epsilon}{72}$. Define the

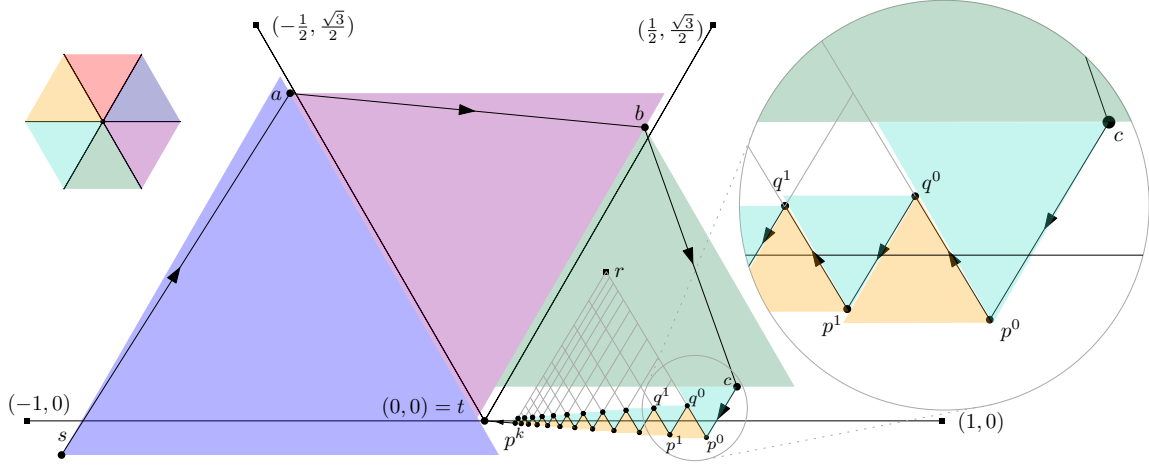


Fig. 6.3: Graph construction achieving a spanning ratio lower bound of $5 - \epsilon$. The shortest path from s to t is the greedy path $s, a, b, c, p^0, q^0, p^1, q^1, \dots, q^{k-1}, p^k, t$. Top Left: The colour coding for edges based on direction.

points $t := (0, 0)$, $s := (-1, 0) + \delta(1, -1)$, $a := (-\frac{1}{2}, \frac{\sqrt{3}}{2}) + \delta(1, -2)$, $b := (\frac{1}{2}, \frac{\sqrt{3}}{2}) + \delta(-2, -3)$, $c := (1, 0) + \delta(-6, 1)$. Next, let p^0 be the unique point such that tp^0 has slope $-\delta$ and p^0c has slope $\sqrt{3} - \delta$. Next, let q^0 be on segment ct such that p^0q^0 has slope $\delta - \sqrt{3}$. For $i > 0$, let q^i be on ct such that $\frac{\|q^{i-1}q^i\|_2}{\|tq^i\|_2} = \frac{\|q^0c\|_2}{\|tc\|_2}$. For $i > 0$, also define p^i to be on tp^0 such that p^iq^i has slope $\delta - \sqrt{3}$. We let k be the least integer such that p_k has x -coordinate less than δ . We have constructed the point set $P := \{t, s, a, b, c\} \cup \{p^0, p^1, \dots, p^k\} \cup \{q^0, q^1, \dots, q^{k-1}\}$. In $\vec{\Theta}_6(P)$, the shortest path from s to t is $s, a, b, c, p^0, q^0, p^1, q^1, \dots, q^{k-1}, p^k, t$ since it is the only path from s to t , ignoring cycles. Indeed, from the perspective of any vertex on the path, all subsequent vertices lie in the same cone. Now we will lower bound the length of this path. First, we have

$$\begin{aligned}
 1 &= \|(-1, 0)(-\frac{1}{2}, \frac{\sqrt{3}}{2})\|_2 && \text{since } \sqrt{\left(\frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} = 1, \\
 &\leq \|(-1, 0)s\|_2 + \|sa\|_2 + \|a(-\frac{1}{2}, \frac{\sqrt{3}}{2})\|_2 && \text{by triangle inequality,} \\
 &\leq \|(-1, 0)s\|_1 + \|sa\|_2 + \|a(-\frac{1}{2}, \frac{\sqrt{3}}{2})\|_1 && \text{since } \|\cdot\|_2 \leq \|\cdot\|_1, \\
 &= 2\delta + \|sa\|_2 + 3\delta && \text{by definition.} \quad (6.1)
 \end{aligned}$$

which implies $\|sa\|_2 \geq 1 - 5\delta$. Using the same technique, we can obtain the following bounds: $\|ab\|_2 \geq 1 - 8\delta$ and $\|bc\|_2 \geq 1 - 12\delta$. Next, let r be the unique point such that p^kr has slope $\sqrt{3} - \delta$ and rp^0 has slope $\delta - \sqrt{3}$. Then by unfolding, notice that we have $\sum_{i=0}^{k-1} \|p_iq_i\|_2 = \|p_0r\|_2$ and $\sum_{i=0}^{k-1} \|q_ip_{i+1}\|_2 = \|rp_k\|_2$, hence the path $p^0, q^0, p^1, q^1, \dots, p^{k-1}, q^{k-1}, p^k$ has length

$\|p^0 r\|_2 + \|r p^k\|_2$. See Figure 6.3. By the triangle inequality, we have $\|p^0(1,0)\|_1 \leq \|p^0 c\|_x + \|p^0 c\|_y + \|c(1,0)\|_1 \leq 2\delta + 2\delta + 7\delta = 11\delta$ and $\|t p^k\|_1 = \|t p^k\|_x + \|t p^k\|_y \leq \delta + \delta = 2\delta$. Next, we claim that r approaches $(\frac{1}{2}, \frac{\sqrt{3}}{2})$ as $\delta \rightarrow 0$.

Claim 1. We have $\|r(\frac{1}{2}, \frac{\sqrt{3}}{2})\|_2 \leq 17\delta$.

Proof. Indeed, first let r^1 be the unique point such that $p^k r^1$ has slope $\sqrt{3}$ and $r^1 p^0$ has slope $-\sqrt{3}$. See Figure 6.4. Since $\max(\|p^k r^1\|_x, \|r^1 p^0\|_x) \leq 1$ vertical distance from r to the lines through $p^k r^1$ and $r^1 p^0$ is at most δ . Then r^1 must be in the ball of radius δ in the $\|\cdot\|_1$ -norm centered at r , meaning $\|r r^1\|_1 \leq \delta$. Next, let r^2 be the unique point such that $p^k r^2$ has slope $\sqrt{3}$ and $r^2(1,0)$ has slope $-\sqrt{3}$. Then we have $\|r^1 r^2\|_2 \leq \frac{2}{\sqrt{3}}\|p^0(1,0)\|_2$. Similarly $\|r^2(\frac{1}{2}, \frac{\sqrt{3}}{2})\|_2 \leq \frac{2}{\sqrt{3}}\|t p^k\|_2$. Putting this together, we obtain

$$\|r(\frac{1}{2}, \frac{\sqrt{3}}{2})\|_2 \leq \|r r^1\|_1 + \|r^1 r^2\|_2 + \|r^2(\frac{1}{2}, \frac{\sqrt{3}}{2})\|_2 \leq \delta + \frac{2}{\sqrt{3}}(11\delta) + \frac{2}{\sqrt{3}}(2\delta) \leq 17\delta.$$

■

Now we can apply the same technique used to prove inequality (6.1) to obtain $\|p^0 r\|_2 \geq 1 - 28\delta$ and $\|r p^k\|_2 \geq 1 - 19\delta$. Therefore the total length of the path is at least

$$\begin{aligned} d_{\vec{\Theta}_6(P)}(s, t) &= \|s a\|_2 + \|a b\|_2 + \|b c\|_2 + \|c p^0\|_2 + \left(\sum_{i=0}^{k-1} \|p^i q^i\|_2 \right) + \|q^{k-1} p^k\|_2 + \|p_k t\|_2 \\ &\geq (1 - 5\delta) + (1 - 8\delta) + (1 - 12\delta) + (1 - 28\delta) + (1 - 19\delta) \\ &= 5 - 72\delta > 5 - \epsilon. \end{aligned}$$

Then since $\|st\|_2 \leq 1$, the spanning ratio of $\vec{\Theta}_6(P)$ is at least $5 - \epsilon$. □

6.5 Upper Bound

6.5.1 Main Argument

The goal of this section is to prove Theorem 4.

Theorem 4. For any point set $P \subset \mathbb{R}^2$ in general position, we have $d_{\vec{\Theta}_6(P)}(s, t) \leq 5\|st\|_{\diamond} \leq 5\|st\|_2$ $\forall s, t \in P$.

Let $P \subset \mathbb{R}^2$ be a point set in general position, and let $G := \vec{\Theta}_6(P)$. Recall that the $\|\cdot\|_{\diamond}$ -norm is used to define G , however Theorem 4 is expressed in terms of the $\|\cdot\|_{\diamond}$ -norm. The following lemma describes the exact relationship between $\|\cdot\|_{\diamond}$ and $\|\cdot\|_{\diamond}$.

Lemma 2. Let $u \in \mathbb{R}^2$, $i \in \mathbb{Z}_6$, and $v \in C_u^i$. Then

$$\|uv\|_{\diamond} = \|uv\|_{\diamond} + \frac{1}{2} \min(\|C_u^i \cap C_v^{i-2}\|, \|C_u^i \cap C_v^{i+2}\|).$$

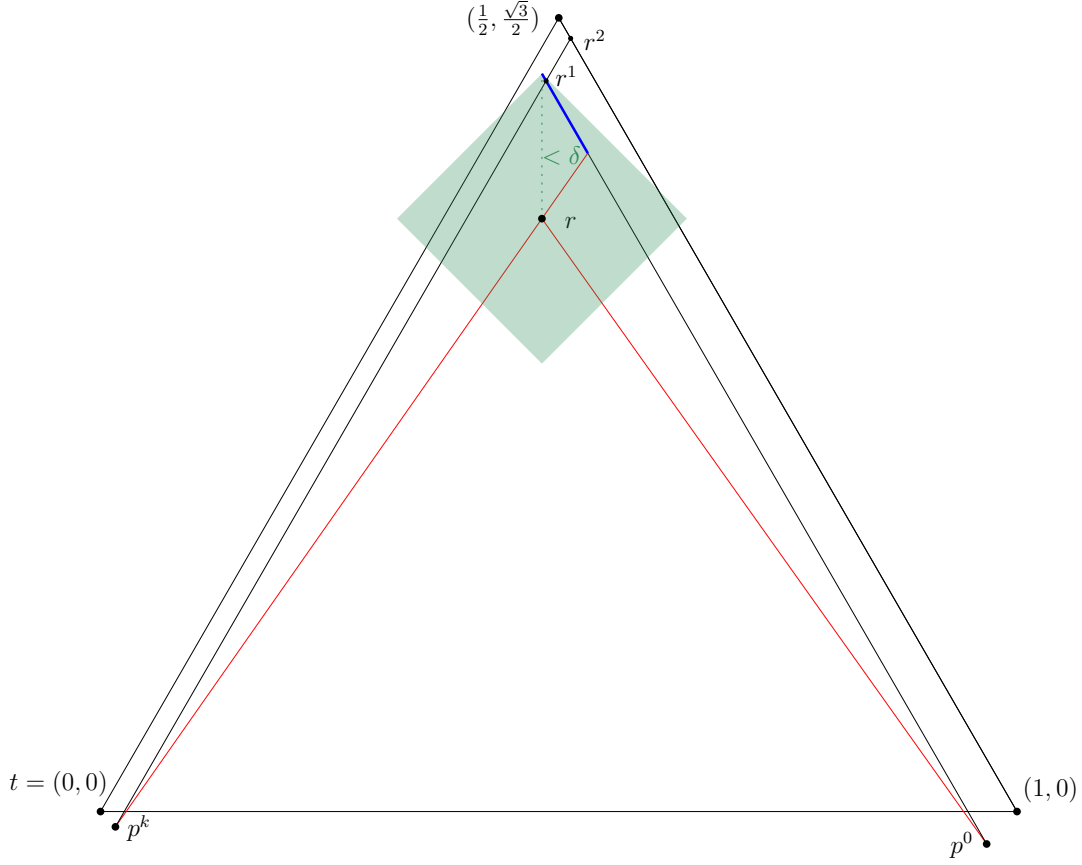


Fig. 6.4: In Claim 1, we define r^1 and r^2 in order to show that r approaches $(\frac{1}{2}, \frac{\sqrt{3}}{2})$. The red lines have slope $\pm(\sqrt{3} - \delta)$, whereas the black lines have slope $\pm\sqrt{3}$. In this example, the line through $r^1 p^0$ has a larger vertical distance to r than the line through $p^k r^1$. This distance is at most δ . Since r^1 is to the right of r in this example, r^1 must lie on the blue segment, which is inside the green disk centered at r with radius at most δ in the $\|\cdot\|_1$ -norm. If instead r^1 were to the left of r , then the diagram would be similar.

Proof. Suppose without loss of generality that $i = 0$ and v is to the right of the bisector B_u^0 . See Figure 6.5. Then $\|C_u^0 \cap C_v^2\| \leq \|C_u^0 \cap C_v^4\|$. Let a be the orthogonal projection of v on R_u^0 . Also let b be the intersection of R_u^0 with a horizontal line through v . Then

$$\|uv\|_{\diamond} + \frac{1}{2}\|C_u^0 \cap C_v^2\| = \|ua\|_2 + \|ab\|_2 = \|ub\|_2 = \|uv\|_{\heartsuit}.$$

The other cases follow by symmetry. □

The next lemma finds use in proving Theorem 4. It allows us to upper bound the length of an edge that may cross cone boundaries, but not cone bisectors.

Lemma 3. *Let $u, v, t \in P$ and $i \in \mathbb{Z}_6$. Suppose $u, v \in C_t^i \cup C_t^{i+1}$ between B_t^i and B_t^{i+1} . Then if $v \in C_u^{i+3} \cup C_u^{i+4}$, we have $\|uv\|_{\heartsuit} \leq 2(\|ut\|_{\diamond} - \|vt\|_{\diamond})$.*

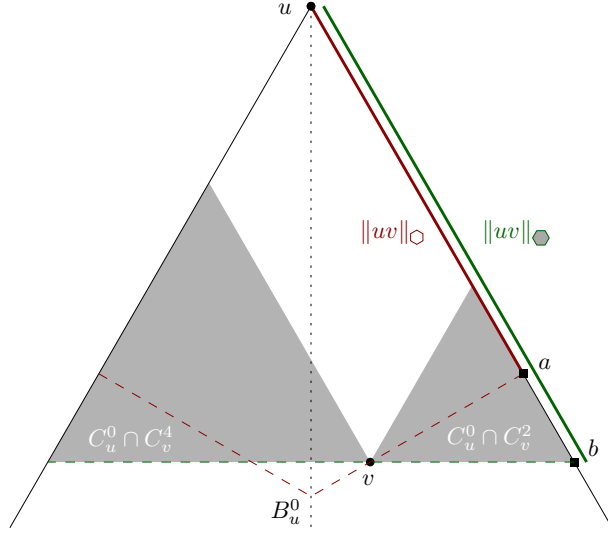


Fig. 6.5: Lemma 2 with $v \in C_u^0$. The dotted vertical ray is the bisector B_u^0 . The horizontal green dashed segment represents all points $p \in C_u^0$ such that $\|up\|_{\text{hex}} = \|uv\|_{\text{hex}}$. The red dashed segments represent all points $p \in C_u^0$ such that $\|up\|_{\text{diamond}} = \|uv\|_{\text{diamond}}$.

Proof. Without loss of generality, suppose $i = 0$. If $v \in C_u^3$, then define $p := R_u^2 \cap R_v^1$, otherwise, if $v \in C_u^4$, then define $p := R_u^4 \cap R_v^5$. Let v' and p' be the orthogonal projections of v and p on R_u^3 . See Figure 6.6. Then

$$\|uv\|_{\text{hex}} = \|up\|_2 = 2\|up'\|_2 \leq 2\|uv'\|_2 = 2(\|ut\|_{\text{diamond}} - \|vt\|_{\text{diamond}}).$$

□

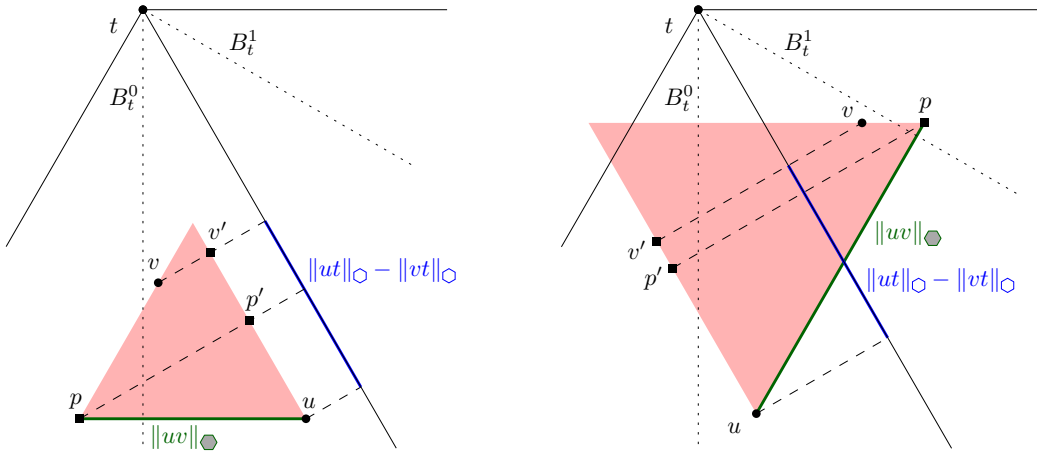


Fig. 6.6: For the proof of Lemma 3, we assume $i = 0$ and show the case when $v \in C_u^4$ on the left and $v \in C_u^3$ on the right.

We prove Theorem 4 by induction on pairs of vertices s, t ordered by hexagonal distance $\|st\|_{\text{hex}}$. The base case considers the closest pair of vertices $s, t \in P$ (i.e. $\|st\|_{\text{hex}}$ is minimal).

Minimality implies $\text{int}(\nabla_s^t)$ must not contain any vertices of P . Therefore there must be an edge from s to t , yielding $d_G(s, t) = \|st\|_2 \leq 5\|st\|_\diamond$.

For the remainder of the proof, fix $s, t \in P$ and suppose that for any vertices $u, v \in P$, if $\|uv\|_\diamond < \|st\|_\diamond$, then $d_G(u, v) \leq 5\|uv\|_\diamond$. We will assume $\|st\|_\diamond = 1$ since our argument holds when scaling the point set. Without loss of generality, suppose $s \in C_t^0$, with s being to the right of B_t^0 . Define the triangle $T := C_t^0 \cap C_s^2$, and denote its side length by $y_0 := \|T\|$. Notice that y_0 describes the *hex* distance from s to the nearest boundary, and by Lemma 2, we obtain $5\|st\|_\diamond = 5 - 2.5y_0$. Suppose towards a contradiction that $5\|st\|_\diamond < d_G(s, t)$. Our approach is to analyse paths from s to t , which will allow us to prove inequalities based on our faulty assumption. These inequalities will form a linear system, then linear programming will help us show that the system is infeasible. The first step in our proof is to establish an empty region which we refer to as R .

Definition 9 (Empty Region R). Consider the points $p := B_s^3 \cap B_t^1$ and $q = B_s^4 \cap B_t^0$. See Figure 6.7. Let Q denote the parallelogram with vertices s, q, t, p , and define the region $R := Q \cup \nabla_s^p \cup \nabla_s^q$.

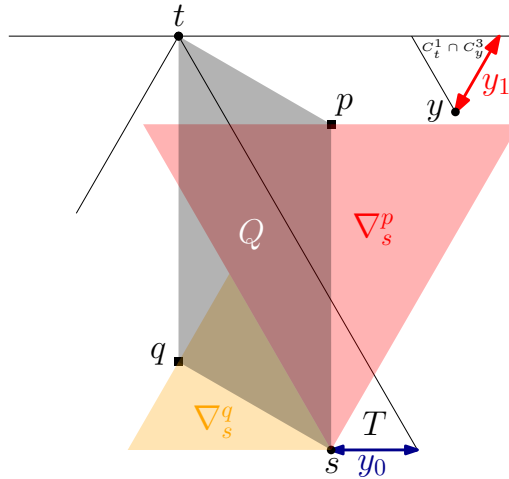


Fig. 6.7: The red triangle is ∇_s^p , the orange triangle is ∇_s^q , and the shaded parallelogram is Q . Their union is R , whose interior can be assumed to not contain any vertices by Lemma 4. Furthermore, there exists a vertex $y \in C_t^1$ close to the horizontal line through t . The triangle T has side-length y_0 and the triangle $C_t^1 \cap C_y^3$ has side-length y_1 .

Let y be the vertex such that sy is the edge in C_s^3 and consider the quantity $y_1 := 1 - \|\nabla_s^y\|$ describing the hexagonal distance $\|\cdot\|_\diamond$ from y to the horizontal line through t . In addition to proving that R is empty, the following lemma proves that y_0 and y_1 cannot both be large. Geometrically, s must be near R_t^0 and y must be near R_t^1 . See Figure 6.7.

Lemma 4. *If $d_G(s, t) > 5\|st\|_\diamond$, then $\text{int}(R)$ does not contain any vertices of P . Furthermore, $y \in C_t^1$ and*

$$2.5y_0 + 3.5y_1 < 1 \quad (6.2)$$

Proof. See Figure 6.8. If there exists a vertex $w \in Q$ with $w \notin \{s, t\}$, then $\|sw\|_{\odot} < 1$ and $\|wt\|_{\odot} < 1$. We obtain

$$\begin{aligned}
 d_G(s, t) &\leq d_G(s, w) + d_G(w, t) && \text{by triangle inequality,} \\
 &\leq 5\|sw\|_{\odot} + 5\|wt\|_{\odot} && \text{by induction,} \\
 &= 5\|st\|_{\odot} && \text{by } \|\cdot\|_{\odot} \text{ collinearity.}
 \end{aligned}$$

Therefore we may assume parallelogram Q contains no vertices. Next we will rule out the

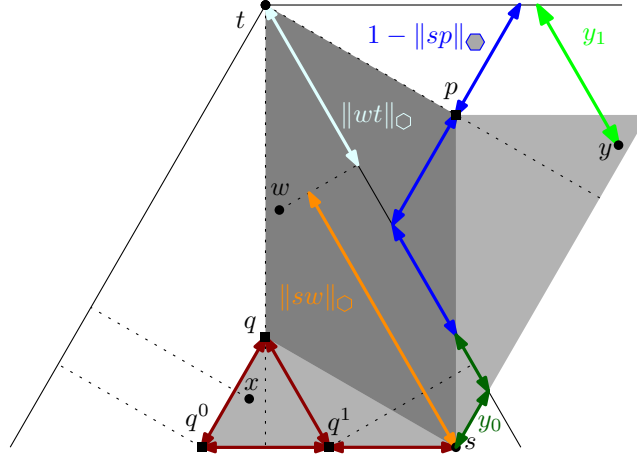


Fig. 6.8: If there is a vertex w in the dark shaded parallelogram, denoted Q , then induction yields paths from s to w and from w to t . If there is an edge sy in the light shaded triangle $\nabla_s^p \setminus Q$, then induction yields a path from y to t . Similarly, if there is an edge sx in the light shaded triangle $\nabla_s^q \setminus Q$, then induction yields a path from x to t .

possibility of $y \in C_t^5 \cup C_t^0$. If $y \in C_t^5$, then we have

$$\begin{aligned}
 d_G(s, t) &\leq \|sy\|_{\odot} + d_G(y, t) && \text{by triangle inequality,} \\
 &\leq 1 + 5y_0 && \text{by induction as } \|yt\|_{\odot} \leq \|y_t\|_{\odot} = \|\nabla_t^y\| \leq \|T\| = y_0 < 1, \\
 &< 5\|st\|_{\odot} && \text{since } y_0 \leq 0.5 \text{ and } 5 - 2.5y_0 = 5\|st\|_{\odot}.
 \end{aligned}$$

On the other hand if $y \in C_t^0$ or $y \in C_t^1$ below B_t^1 , then we use Lemma 3 to obtain $\|sy\|_{\odot} \leq 2(\|st\|_{\odot} - \|yt\|_{\odot})$. Indeed, if sy crosses B_t^0 , then let $z := sy \cap B_t^0$ and apply Lemma 3 to both sz and zy . We obtain

$$\begin{aligned}
 d_G(s, t) &\leq \|sy\|_{\odot} + d_G(y, t) && \text{by triangle inequality,} \\
 &\leq 2(\|st\|_{\odot} - \|yt\|_{\odot}) + 5\|yt\|_{\odot} && \text{by Lemma 3 and induction,} \\
 &= 2\|st\|_{\odot} + 3\|yt\|_{\odot} \\
 &< 5\|st\|_{\odot} && \text{since } \|yt\|_{\odot} < \|st\|_{\odot}.
 \end{aligned}$$

Therefore we may assume $y \in C_t^1$ above the bisector B_t^1 . We have $\|yt\|_{\odot} \leq \|C_t^1 \cap C_s^3\| = 1 - y_0$. Recall that $y_1 := 1 - \|\nabla_s^y\|$, and notice that we also have $y_1 = \|C_t^1 \cap C_y^3\|$. By Lemma 2,

$\|yt\|_{\odot} = \|yt\|_{\ominus} - \frac{1}{2}y_1$. This gives us

$$\begin{aligned}
5 - 2.5y_0 &< d_G(s, t) && \text{by assumption,} \\
&\leq \|sy\|_{\ominus} + d_G(y, t) && \text{by triangle inequality,} \\
&\leq (1 - y_1) + (5(1 - y_0) - 2.5y_1) && \text{by } \|\nabla_s^y\| = \|sy\|_{\ominus} \text{ and induction.} \quad (6.3)
\end{aligned}$$

By re-arranging inequality (6.3), we obtain $2.5y_0 + 3.5y_1 < 1$, proving (6.2). Notice that $2y_0 + 3(1 - \|sp\|_{\ominus}) = 1$ by symmetry of isosceles triangles. We have

$$y_1 < \frac{1 - 2.5y_0}{3.5} < \frac{1 - 2y_0}{3} = 1 - \|sp\|_{\ominus},$$

which implies that the region $\text{int}(\nabla_s^p)$ contains no vertices. Indeed, if there were a point $u \in \text{int}(\nabla_s^p)$, then $1 - y_1 = \|sy\|_{\ominus} \leq \|su\|_{\ominus} \leq \|sp\|_{\ominus}$. Thus far we have shown that Q and ∇_s^p do not contain vertices in their interior, hence it remains to argue that $\text{int}(\nabla_s^q)$ is also empty. Towards a contradiction, assume there is a vertex $x \in \text{int}(\nabla_s^q)$. If x is to the right of the bisector B_t^0 , then we have

$$\begin{aligned}
d_G(s, t) &\leq \|sx\|_{\ominus} + d_G(x, t) && \text{by triangle inequality,} \\
&\leq 2(\|st\|_{\odot} - \|xt\|_{\odot}) + 5\|xt\|_{\odot} && \text{by Lemma 3 and induction,} \\
&= 2\|st\|_{\odot} + 3\|xt\|_{\odot} \\
&< 5\|st\|_{\odot} && \text{since } \|xt\|_{\odot} < \|st\|_{\odot}.
\end{aligned}$$

On the other hand if x is to the left of the bisector B_t^0 , then let $q^0 := R_q^5 \cap R_s^4$ and $q^1 := R_q^0 \cap R_s^4$. Notice that $\|sq^0\|_{\ominus} = 2\|sq^1\|_{\ominus}$. We have

$$\begin{aligned}
d_G(s, t) &\leq \|sx\|_{\ominus} + d_G(x, t) && \text{by triangle inequality,} \\
&\leq \|sq^0\|_{\ominus} + 5\|xt\|_{\odot} && \text{by induction since } \|xt\|_{\odot} < 1, \\
&\leq 2\|sq^1\|_{\ominus} + 5\|q^0t\|_{\odot} && \text{since } \|xt\|_{\odot} \leq \|q^0t\|_{\odot}, \\
&= 2(2(\|st\|_{\odot} - \|q^1t\|_{\odot})) + 5\|q^1t\|_{\odot} && \text{by Lemma 3 and } \|q^0t\|_{\odot} = \|q^1t\|_{\odot}, \\
&= 4\|st\|_{\odot} + \|q^1t\|_{\odot} \\
&\leq 5\|st\|_{\odot} && \text{since } \|q^1t\|_{\odot} \leq \|st\|_{\odot}.
\end{aligned}$$

Therefore the region $\text{int}(\nabla_s^q)$ contains no vertices. This completes the proof. $\square$

Now that we have established an empty region R and the position of y , we would like to analyse the greedy path from y to t . Recall that for any vertices $u, v \in P$, the greedy path from u to v is denoted $\pi(u, v)$. Every edge ab of $\pi(u, v)$ satisfies $b \in C_a^i \iff v \in C_a^i$. Intuitively, the greedy path simply follows the edge in the direction of the destination. In [1] the authors show that for every edge uv of $\pi(s, t)$, we have $\|ut\|_{\ominus} > \|vt\|_{\ominus}$. However, this does not guarantee that the greedy path is an efficient path to t . The main reason is that $\pi(s, t)$ may spiral around t arbitrarily many times while making negligible progress. The authors of [1] show how to avoid a spiraling greedy path by assuming there is an edge from t to s and using the empty triangle ∇_t^s as an obstruction. This is done in Lemmas 2, 3 and 4 from [1], which we restate here using our notation.

Lemma 5 (Lemma 2 in [1]). *If edge uv is on $\pi(u, t)$ and $u \in C_t^i$, then $v \in C_t^{i-1} \cup C_t^i \cup C_t^{i+1}$.*

Lemma 6 (Lemma 3 in [1]). *Let $s, t \in P$ and assume ts is an edge of $\vec{\Theta}_6(P)$. Assume the vertices a, b, c appear in this order in $\pi(s, t)$, not necessarily distinct or consecutive. Then if a and c are both in the same cone C_t^i for $i \in \mathbb{Z}_6$, then $\text{int}(\nabla_a^b) \cap \text{int}(\nabla_t^c) = \emptyset$.*

Lemma 7 (Lemma 4 in [1]). *Let $s, t \in P$ and assume ts is an edge of $\vec{\Theta}_6(P)$. Then for $i \in \mathbb{Z}_6$, we have*

$$\sum_{\substack{ab \in \pi(s, t) \\ a \in C_t^i}} \|ab\| \leq \|\nabla_t^s\|$$

Instead of working with an empty triangle that forces the greedy path not to spiral as in [1], we will ignore any spiralling by only using the greedy path up to and including the first edge that crosses the boundary R_t^0 . Despite this change, the arguments in [1] can be applied in our setting to prove Lemma 8 which is a slightly modified version of Lemma 4 from [1].

Lemma 8. *Fix $j \in \mathbb{Z}_6$ and let $s, t, u, v \in P$ such that uv is an edge of $\pi(s, t)$ and no edge of the truncated greedy path $\pi(s, t)_u$ crosses R_t^j . For $i \in \mathbb{Z}_6$, let f_i denote the first vertex of $\pi(s, t)_u$ in C_t^i . If no such vertex f_i exists, let $f_i := t$. Then $\|\pi(s, t)_v\|_{\text{hex}} \leq \sum_{i=0}^5 \|f_i t\|_{\text{hex}}$.*

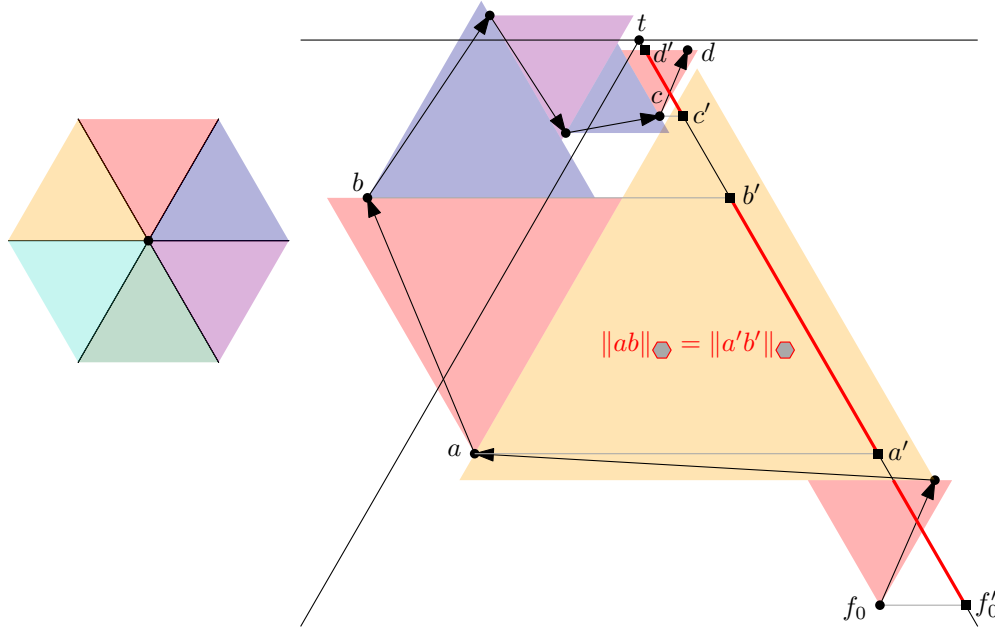


Fig. 6.9: The first vertex of $\pi(s, t)$ in C_t^0 is f_0 . For any point p below t , p' is the point intersecting R_t^i and the horizontal line through p . The total length of all edges starting in cone C_t^0 is at most $\|f_0 t\|_{\text{hex}} = \|f'_0 t\|_2$.

Proof. Suppose without loss of generality that $i = 0$. For any point p below t , let p' denote the intersection of the horizontal line through p with the ray R_t^0 . See Figure 6.9. We claim that if ab

and cd are edges of $\pi(s, t)_v$ such that $a, c \in C_t^0$, then the segments $a'b'$ and $c'd'$ are disjoint. Furthermore, both $a'b'$ and $c'd'$ are subsegments of $f_0't$. This claim follows from Lemma 6 of [1] since the path $\pi(s, t)_u$ is assumed to not cross R_t^j . We sum all edges with source in C_t^0 to obtain

$$\sum_{\substack{ab \text{ on } \pi(s, t)_v \\ a \in C_t^0}} \|ab\|_{\odot} = \sum_{\substack{ab \text{ on } \pi(s, t)_v \\ a \in C_t^0}} \|a'b'\|_{\odot} \leq \|f_0't\|_{\odot} = \|f_0t\|_{\odot}.$$

The lemma follows by summing over the edges in all cones C_t^i for $i \in \mathbb{Z}_6$. $\square$

For the rest of the section, we make Assumptions 1 and 2 to help us prove Theorem 4. We then consider the case where Assumption 1 is not satisfied in Section 6.5.2. On the other hand, Section 6.5.2 treats the case when Assumption 2 does not hold. The first assumption roughly states that the greedy path from y spirals counter-clockwise around t without making significant progress to t . We define $u^y v^y$ to be the first edge of $\pi(y, t)$ which crosses R_t^0 . If such an edge does not exist, let $u^y := t$ and $v^y := t$. Then the first assumption can be stated as follows:

Assumption 1. *There exists a vertex $q \in C_t^5$ on $\pi(y, t)_{v^y}$ with $\|qt\|_{\odot} > y_0$.*

For an example of when Assumption 1 holds, see Figure 6.18, whereas a non-example is illustrated in Figure 6.17. Recall the triangle T is defined as $T := C_t^0 \cap C_s^2$. Our second assumption is that all vertices in T have short outgoing edges in their fourth cone. See Figure 6.12 for an example of where Assumption 2 holds, and see Figure 6.13 for a non-example.

Assumption 2. *For every vertex $v \in T$, we have $P \cap \text{int}(C_t^0 \cap C_v^4) \neq \emptyset$.*

Let x be the vertex such that sx is the edge in C_s^4 . By Assumption 2, x exists and $x \in C_t^0$. Combining our two assumptions allows us to naturally consider two candidate paths, loosely referred to as the x -path and the y -path, and show that one of them is sufficiently short to force a contradiction with $d_G(s, t) > 5\|st\|_{\odot}$. Informally, the x -path travels clockwise around t , whereas the y -path travels in the opposite direction. Where these two paths meet, they must compete for space. The lengths of the paths are primarily dictated by their *long* edges which cross cone boundaries around t . These *long* edges come with large empty regions which squeeze the other path closer to t . We formalize this intuition below.

Recall that $u^y v^y$ is the first edge of $\pi(y, t)$ that crosses the boundary R_t^0 . Consider the candidate path sy concatenated with $\pi(y, t)_{v^y}$, which we loosely refer to as the y -path, ending at v^y . Overall, the y -path progresses counter-clockwise about t until C_t^5 , however it is still possible that the y -path reverses direction and ends up crossing R_t^0 from C_t^1 to C_t^0 . Maintaining full generality, we may only assume that for some $i \in \{0, 1\}$, we have $u^y \in C_t^i$ and $v^y \in C_t^{1-i}$, which follows from Lemma 5. In light of Lemma 8, we need not concern ourselves with every edge of this path. An upper bound can be established from only the first vertex of each cone. In fact, we only need to upper bound the value $\|f_i t\|_{\odot}$ where f_i is the first vertex of C_t^i , hence we look to the previous cone for structure. For $i \in \{1, 2, 3, 4, 5\}$, let $y^i \in C_t^i$ be the vertex of $\pi(y, t)_{u^y}$ which maximizes $\|C_t^i \cap C_{y^i}^{i-2}\|$. See Figure 6.10 for an example.

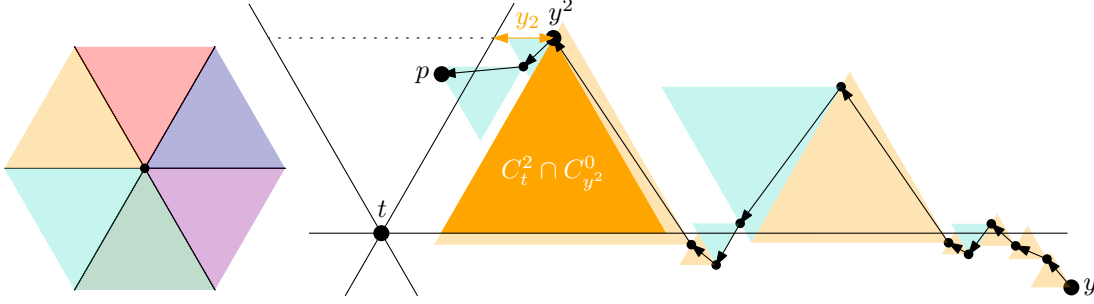


Fig. 6.10: In this example, p is the first vertex of $\pi(y, t)_{vy}$ in cone C_t^3 . The highest vertex of $\pi(y, t)_{vy}$ in C_t^2 is y^2 by definition. The maximal triangle $C_t^2 \cap C_{y^2}^0$ is coloured orange, and $y_2 = \|y^2 t\|_{\odot} - \|C_t^2 \cap C_{y^2}^0\|$. Observation 3 implies that y^2 has a greater y -coordinate than p , thus $\|C_t^2 \cap C_{y^2}^0\|$ provides an upper bound on $\|pt\|_{\odot}$.

Observation 3. For $i \in \{1, 2, 3, 4, 5\}$, no vertex of $\pi(y, t)_{yi}$ is in $\text{int}(C_t^{i+1})$.

Proof. The observation holds for $i = 1$ since $y^1 = y$. Then for $i \in \{2, 3, 4, 5\}$, consider the first vertex v of C_t^{i+1} on $\pi(y, t)_{uy}$. Then y^i and t must be on opposite sides of the line $R_v^{i-1} \cup R_v^{i+2}$, which implies $\|y^i t\|_{\odot} > \|vt\|_{\odot}$, meaning that y^i appears before v on $\pi(y, t)_{uy}$. ■

Observation 3 implies y^1, y^2, y^3, y^4, y^5 appear in order on $\pi(y, t)_{uy}$. Intuitively, y^i is the head of a *long* edge whose tail is in the previous cone C_t^{i-1} . Indeed, if the tail was in the same cone as y^i , then $\|C_t^i \cap C_{y^i}^{i-2}\|$ would not be maximal. We define the distance y_i from y^i to C_t^{i+1} as $y_i := \|C_t^i \cap C_{y^i}^{i+2}\|$. Equivalently, $y_i = \|y^i t\|_{\odot} - \|C_t^i \cap C_{y^i}^{i-2}\|$. By Assumption 1, the y -path must pass through cones $C_t^1, \dots, C_t^5$, hence y_i is well-defined for $i \in \{1, 2, 3, 4, 5\}$. Also notice that the region $\text{int}(C_t^i \cap C_{y^i}^{i-2})$ is empty since y^i is necessarily the head of an edge from C_t^{i-1} . Next, for each $i \in \mathbb{Z}_6$, we let Y_i be the value of $\|ut\|_{\odot}$ where u is the first vertex of $\pi(y, t)_{uy}$ with $u \in C_t^i$. If no such vertex $u \in C_t^i$ exists, then $Y_i := 0$. First, we have $Y_1 = \|yt\|_{\odot} \leq 1 - y_0$. Next, for $i \in \{1, 2, 3, 4, 5\}$, cone C_t^{i+1} can only be visited from cone C_t^i on $\pi(y, t)_{uy}$, therefore we see that for $i \in \{1, 2, 3, 4, 5\}$,

$$y_0 + Y_1 \leq 1 \quad \text{and} \quad y_i + Y_{i+1} \leq y_i + \|C_t^i \cap C_{y^i}^{i-2}\| = \|y^i t\|_{\odot} \leq Y_i \quad (6.4)$$

We include these inequalities in our system which will eventually be proven infeasible. Next we will discuss the second candidate path: the x -path. Recall sx is the edge in $C_s^4 \cap C_t^0$. We will define the analogous variables for x as we did for y . Let $u^x v^x$ be the first edge of $\pi(x, t)$ that crosses R_t^0 . As before, Lemma 5 implies that for some $i \in \{0, 1\}$, we have $u^x \in C_t^i$ and $v^x \in C_t^{1-i}$. If no such crossing exists, then set $u^x := t$ and $v^x := t$. For $i \in \{0, 5, 4, 3, 2\}$, let $x^i \in C_t^i$ be the vertex of $\pi(x, t)_{ux}$ which maximizes $\|C_t^i \cap C_{x^i}^{i+2}\|$. We define the distance x_i from x^i to C_t^{i-1} as $x_i := \|C_t^i \cap C_{x^i}^{i-2}\|$. If no such x^i exists, then define $x_i := 0$. However, we will later argue in Claim 3 that the vertices x^0, x^5, x^4, x^3 must exist, otherwise the x -path is sufficiently short to force a contradiction. Observe that for $i \in \{5, 4, 3, 2\}$ the region $C_t^i \cap C_{x^i}^{i+2}$

is empty since x^i is the head of an edge from C_t^{i+1} . Next, for each $i \in \mathbb{Z}_6$, we let X_i be the value of $\|ut\|_{\ominus}$ where u is the first vertex of $\pi(x, t)_{u^x}$ with $u \in C_t^i$. If no such vertex u exists, then $X_i := 0$. Since greedy paths decrease in $\|\cdot\|_{\ominus}$ -norm, then we have $X_0 = \|xt\|_{\ominus} \leq 1$. Next, for $i \in \{5, 4, 3, 2, 1\}$, if v is the first vertex of $\pi(x, t)_{u^x}$ in C_t^i , then its predecessor u must be in cone C_t^{i+1} , therefore we see that for $i \in \{5, 4, 3, 2, 1\}$,

$$X_0 \leq 1 \quad \text{and} \quad x_{i+1} + X_i \leq x_{i+1} + \|C_t^{i+1} \cap C_{x_{i+1}}^{i+3}\| = \|x^{i+1}t\|_{\ominus} \leq X_{i+1}. \quad (6.5)$$

Intuitively, the only vertices in cone C_t^i that are important to us are x^i and y^i . Now we consider possible paths from s to t . We could take the y -path from s to v^y , then apply induction at v^y . By Lemma 8, this yields

$$\begin{aligned} 5 - 2.5y_0 &\leq d_G(s, t) \\ &\leq \|sy\|_{\ominus} + \|\pi(y, t)_{v^y}\|_{\ominus} + d_G(v^y, t) \\ &\leq (1 - y_1) + (Y_1 + Y_2 + Y_3 + Y_4 + Y_5 + Y_0) + 5\|v^y t\|_{\ominus}. \end{aligned} \quad (6.6)$$

On the other hand, we could instead follow the x -path from s to v^x , then apply induction at v^x . Similarly, we obtain

$$\begin{aligned} 5 - 2.5y_0 &\leq d_G(s, t) \\ &\leq \|sx\|_{\ominus} + \|\pi(x, t)_{v^x}\|_{\ominus} + d_G(v^x, t) \\ &\leq (1 - y_0 - x_0) + (X_0 + X_5 + X_4 + X_3 + X_2 + X_1) + 5\|v^x t\|_{\ominus}. \end{aligned} \quad (6.7)$$

While we have proven several inequalities (6.2),(6.4),(6.5),(6.6),(6.7), they are not yet strong enough to form an infeasible system. First, notice that since R may not contain vertices, then $u^y v^y$ must cross both B_t^0 and B_t^1 . By the following lemma, this implies that the edge $u^y v^y$ makes significant progress towards t .

Lemma 9. *Let $u, v, t \in P$ with edge uv contained in ∇_u^t . Assume uv crosses both B_t^0 and B_t^1 , then $2\|vt\|_{\ominus} \leq \|uv\|_{\ominus}$.*

Proof. Suppose without loss of generality that $u \in C_t^0$ with u to the left of B_t^0 , and $v \in C_t^1$ with v above B_t^1 . See Figure 6.11. Consider the point $t' := B_t^0 \cap R_v^5$. Since u is to the left of B_t^0 , then the segment vt' is contained in ∇_u^v , hence $\|vt'\|_2 \leq \|\nabla_u^v\| = \|uv\|_{\ominus}$. On the other hand, consider the point $v' := R_t^1 \cap B_v^3$. Then considering that vt' and $v't$ have respective slopes $\sqrt{3}$ and 0, then we have $2\|vt\|_{\ominus} = 2\|v't\|_2 = \|vt'\|_2 \leq \|uv\|_{\ominus}$. $\square$

Lemma 9 quantifies the toll required for an edge to cross the empty region. Intuitively, the distance to t is halved each time the empty region R is crossed. In order to also apply Lemma 9 to $u^x v^x$, we require two observations, which follow from Assumptions 1 and 2.

Observation 4. *If x^1 exists, then $\|x^1 t\|_{\ominus} \leq 1 - y_0$.*

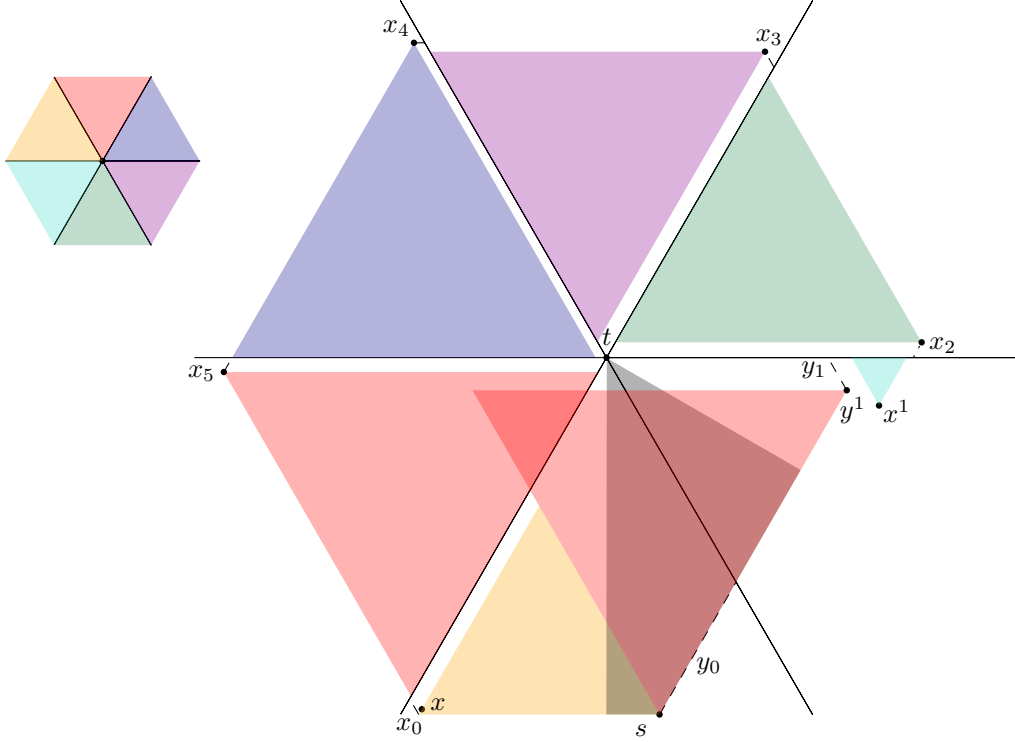


Fig. 6.12: In this diagram illustrating the proof of Observation 4, we have $\|x^1 t\|_{\text{hex}} > 1 - y_0$, and we can see that the y -path will be forced inside the x -path, and thus no vertex in C_t^5 on the y -path can be left of R_s^4 , contradicting Assumption 1. Notice that Assumption 2 holds here since $s \in T$ and $x \in C_t^0 \cap C_s^4$.

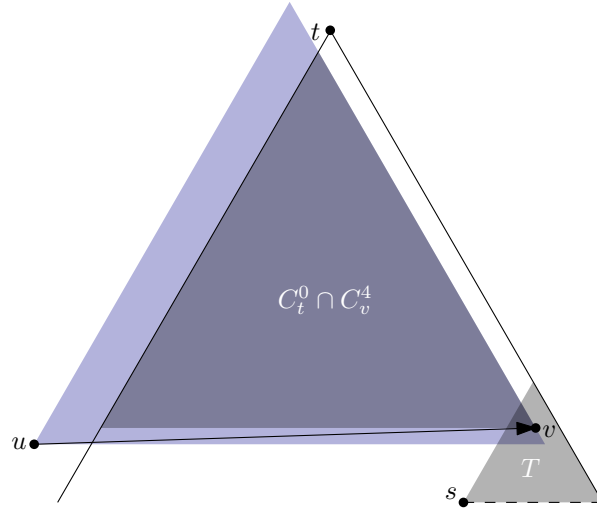


Fig. 6.13: In this example, $v \in T$, hence Assumption 2 states that $C_t^0 \cap C_v^4$ must contain at least one vertex in its interior. For Observation 5, if uv is an edge from C_t^5 to T , then the region $C_v^4 \cap C_t^0$ contains no vertices in its interior, contradicting Assumption 2.

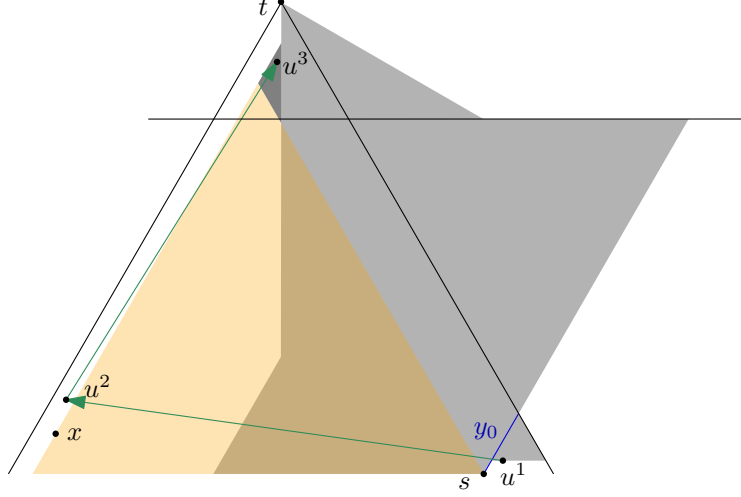


Fig. 6.14: Claim 2: If there is a vertex to the left of B_t^0 in C_x^2 (the dark grey region), then we can form a path that passes through s, u^1, u^2, u^3, t . Induction is used to bound $d_G(s, u^1)$ and $d_G(u^3, t)$, whereas u^1u^2 and u^2u^3 are edges in G (in green).

u^2u^3 in $C_{u^2}^2$. By assumption that there is at least one vertex to the left of B_t^0 in the interior of C_x^2 , then u^3 must exist. Furthermore, we must have $u^3 \in C_s^3$. Since u^3 is above ∇_s^y , then $\|u^3t\|_{\odot} \leq y_1$. We apply induction from u^3 to t . This yields

$$\begin{aligned} 5 - 2.5y_0 &< d_G(s, t) \\ &\leq d_G(s, u^1) + \|u^1u^2\|_{\odot} + \|u^2u^3\|_{\odot} + d_G(u^3, t) \\ &\leq 5y_0 + 1 + 1 + 5y_1. \end{aligned} \tag{6.8}$$

However, the system of inequalities (6.2, 6.8) is infeasible. Therefore we may assume that the region $C_x^2 \cap C_t^0$ to the left of B_t^0 is empty for the remainder of the proof. ■

In order to further restrict the set of solutions to our linear system of inequalities towards infeasibility, we will consider several cases with more granularity. We will first consider the direction in which the edge $u^y v^y$ crosses the boundary R_t^0 in order to strengthen our upper bound on $\|v^y t\|_{\odot}$. Recall that $\|v^y t\|_{\odot} < 1$, hence $5\|v^y t\|_{\odot}$ accounts for the path from v^y to t by induction. For this reason, $\|v^y t\|_{\odot}$ is an important quantity to analyse. We will apply the same analysis to $u^x v^x$ to also upper bound $\|v^x t\|_{\odot}$, then lastly we will consider the trade-off between the x -path and y -path. In Figure 6.15, we show an example of the x -path and y -path that compete for space.

Before proceeding to the cases, we show in Claim 3 that the x -path must visit the cone C_t^3 .

Claim 3. *The path $\pi(x, t)_{v^x}$ must contain a vertex in C_t^3 .*

Proof. If the claim did not hold, then $u^x v^x$ would necessarily cross R in the counter-clockwise direction. See Figure 6.16. If $\pi(x, t)_{v^x} \cap C_t^5 \neq \emptyset$, then we use Lemma 8 and apply induction at

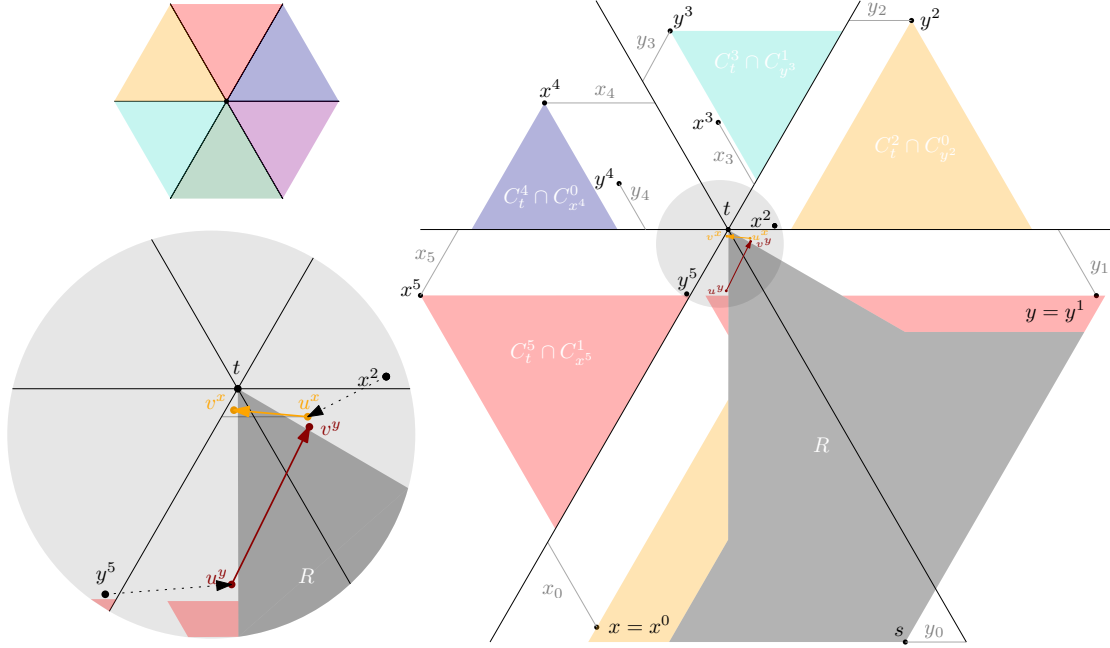


Fig. 6.15: Top Left: The colour coding for edges based on direction. Right: The grey region R contains no vertices in its interior by Lemma 4. The triangular regions are coloured according to their edge direction. This example shows the case when $v^y \in C_t^1$ (case Y1), $v^x \in C_t^0$ (case X0), and $j = 3$ (case J3). Notice that the empty turquoise region $C_t^3 \cap C_{y^3}^1$ constrains the position of x^3 , forcing x^2 to be considerably closer to t . Lower Left: This enlarged view shows the edges that cross the empty grey region R . Notice that the edges $u^x v^x$ and $u^y v^y$ both make considerable progress towards t (quantified in Lemma 9).

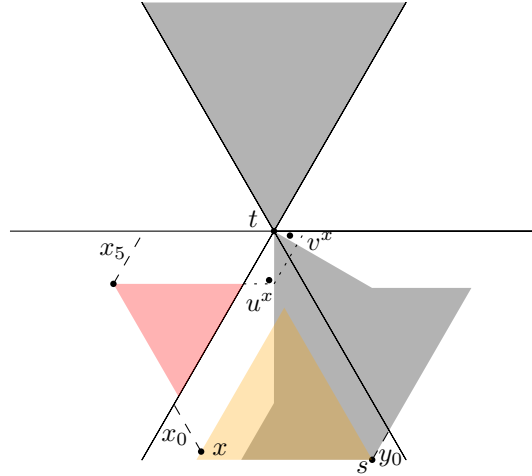


Fig. 6.16: Claim 3: If $\pi(x, t)_{v^x}$ does not enter C_t^3 , then we can form an infeasible system of linear inequalities.

v^x to obtain

$$\begin{aligned}
 5 - 2.5y_0 &< d_G(s, t) \\
 &\leq \|sx\|_{\odot} + \|\pi(x, t)_{v^x}\|_{\odot} + d_G(v^x, t) \\
 &\leq (1 - x_0 - y_0) + (X_0 + X_4) + 5\|v^x t\|_{\odot} \\
 &\leq (1 - x_0 - y_0) + 1 + (1 - x_0) + (1 - x_0 - x_5) + 5 \min(x_0, \frac{x_5}{2}). \tag{6.9}
 \end{aligned}$$

However, the linear system (6.2,6.9) is infeasible. On the other hand if $\pi(x, t)_{v^x} \cap C_t^5 = \emptyset$, then

$$\begin{aligned}
d_G(s, t) &\leq \|sx\|_{\square} + \|\pi(x, t)_{v^x}\|_{\square} + d_G(v^x, t) \\
&\leq (1 - x_0 - y_0) + X_0 + 5\|v^x t\|_{\square} \\
&\leq (1 - x_0 - y_0) + 1 + 5x_0 \\
&\leq 4 - y_0 && \text{since } x_0 \leq 1/2, \\
&\leq 5 - 2.5y_0 && \text{since } y_0 \leq 1/2.
\end{aligned}$$

which is a contradiction. To summarize, $\pi(x, t)_{v^x}$ must enter C_t^3 . ■

First, notice that $u^y v^y$ can cross R_t^0 either in the clockwise or counter-clockwise direction:

Case Y0: $v^y \in C_t^0$. Then we must have $u^y \in C_t^1$. We obtain the following two inequalities:

$$\|v^y t\|_{\square} \leq y_1, \quad \text{by empty region } \text{int}(\nabla_s^y). \quad (6.10)$$

$$\|v^y t\|_{\square} \leq \frac{1}{2} \min(y_2, y_3, y_4), \quad \text{by Assumption 1 and Lemma 9.} \quad (6.11)$$

Case Y1: $v^y \in C_t^1$. Then we must have $u^y \in C_t^0$. We obtain

$$\|v^y t\|_{\square} \leq x_0, \quad \text{by Claim 2.} \quad (6.12)$$

$$\|v^y t\|_{\square} \leq \frac{Y_0}{2}, \quad \text{by Lemma 9.} \quad (6.13)$$

Similarly, $u^x v^x$ can also cross R_t^0 in one of two possible ways.

Case X0: $v^x \in C_t^0$. Then we have

$$\|v^x t\|_{\square} \leq y_1, \quad \text{by Observation 4 and empty region } \text{int}(\nabla_s^y). \quad (6.14)$$

$$\|v^x t\|_{\square} \leq \frac{X_1}{2}, \quad \text{by Lemma 9.} \quad (6.15)$$

Case X1: $v^x \in C_t^1$. Then $u^y \in C_t^0$. We have

$$\|v^x t\|_{\square} \leq x_0, \quad \text{by Claim 2 and 5.} \quad (6.16)$$

$$\|v^x t\|_{\square} \leq \frac{1}{2} \min(x_5, x_4, x_3), \quad \text{by Claim 3 and Lemma 9.} \quad (6.17)$$

Next, we will derive several useful inequalities that reflect the balance between the two candidate paths. Intuitively, if $\pi(y, t)_{v^y}$ is long, then the empty regions $C_t^i \cap C_{y^i}^{i-2}$ must be large, leaving no room for the other path $\pi(x, t)_{v^x}$. We formalize this trade-off as follows. Let j be the greatest index in $\{2, 3, 4, 5\}$ such that $\|C_t^j \cap C_{y^j}^{j-2}\| > x_j$. For example when $j = 3$, see Figure 6.15.

Case: No such j exists. Then by definition we must have

$$Y_{i+1} \leq \|C_t^i \cap C_{y^i}^{i-2}\| \leq x_i \quad \text{for } 2 \leq i \leq 5. \quad (6.18)$$

Case: $j \in \{2, 3, 4, 5\}$. As j is maximal, then $\|C_t^i \cap C_{y^i}^{i-2}\| \leq x_i$ for $j < i \leq 5$. Next, since $\|C_t^j \cap C_{y^j}^{j-2}\| > x_j$, then we must have $\|y^k t\|_{\bullet} > \|x^k t\|_{\bullet}$ for $2 \leq k \leq j$. By the empty regions $C_t^i \cap C_{y^i}^{i-2}$, we must have

$$Y_{i+1} \leq \|C_t^i \cap C_{y^i}^{i-2}\| \leq x_i \quad \text{for } j < i \leq 5, \quad \text{and} \quad X_{k-1} \leq y_k \quad \text{for } 2 \leq k \leq j. \quad (6.19)$$

We denote the cases where $j \in \{2, 3, 4, 5\}$ as J_2, J_3, J_4, J_5 , and observe that for each case, we yield four inequalities from (6.19).

Under Assumptions 1 and 2 we have the following possibility space: $\{Y_0, Y_1\} \times \{X_0, X_1\} \times \{(6.18), J_2, J_3, J_4, J_5\}$. In total there are $2 \times 2 \times 5 = 20$ possibilities, and in every case, the inequalities (6.2), (6.4), (6.5), (6.6), (6.7) also hold. It can be shown that the linear system of inequalities is infeasible for 16 of the 20 possibilities. In Section 6.5.2, we show that for the remaining four cases, we can define a third candidate path which is sufficiently short for a contradiction. These remaining four cases are $(X_1, Y_1, J_2), (X_1, Y_1, J_3), (X_1, Y_1, J_4), (X_0, Y_0, J_3)$.

6.5.2 Remaining Cases for Upper Bound Proof

In Section 6.5.2, we prove Theorem 4 when Assumption 1 does not hold. Then, Section 6.5.2 proves Theorem 4 when Assumption 1 holds but Assumption 2 does not. Finally, in Section 6.5.2, we take care of the remaining four cases involved in proving Theorem 4 under Assumptions 1 and 2.

Proving Theorem 4 without Assumption 1

In this section, we assume that Assumption 1 does **not** hold. This means that if $q \in \pi(y, t)_{v^y} \cap C_t^5$, then $\|\nabla_t^q\| \leq y_0$. See Figure 6.17. We consider the following three cases:

Case $\pi(y, t)_{v^y} \cap C_t^2 = \emptyset$: Then $u^y v^y$ crosses from C_t^1 to C_t^0 and $Y_i = 0$ for $i \neq 1$. By Lemma 8, we obtain

$$\begin{aligned} 5 - 2.5y_0 &< d_G(s, t) \\ &\leq \|sy\|_{\bullet} + \|\pi(y, t)_{v^y}\|_{\bullet} + d_G(v^y, t) \\ &\leq (1 - y_1) + Y_1 + 5\|v^y t\|_{\bullet} \\ &\leq (1 - y_1) + (1 - y_0) + 5y_1. \end{aligned} \quad (6.20)$$

It can be verified that the linear system (6.2, 6.20) is infeasible.

Case $v^y \in C_t^0$: We can assume $\pi(y, t)_{v^y}$ visits C_t^2 , so y^2 is well-defined. Recall $y^2 \in C_t^2 \cap \pi(y, t)_{v^y}$ and y^2 maximizes $\|C_t^2 \cap C_{y^2}^0\|$. Also note that we defined $y_2 := \|C_t^2 \cap C_{y^2}^4\|$.

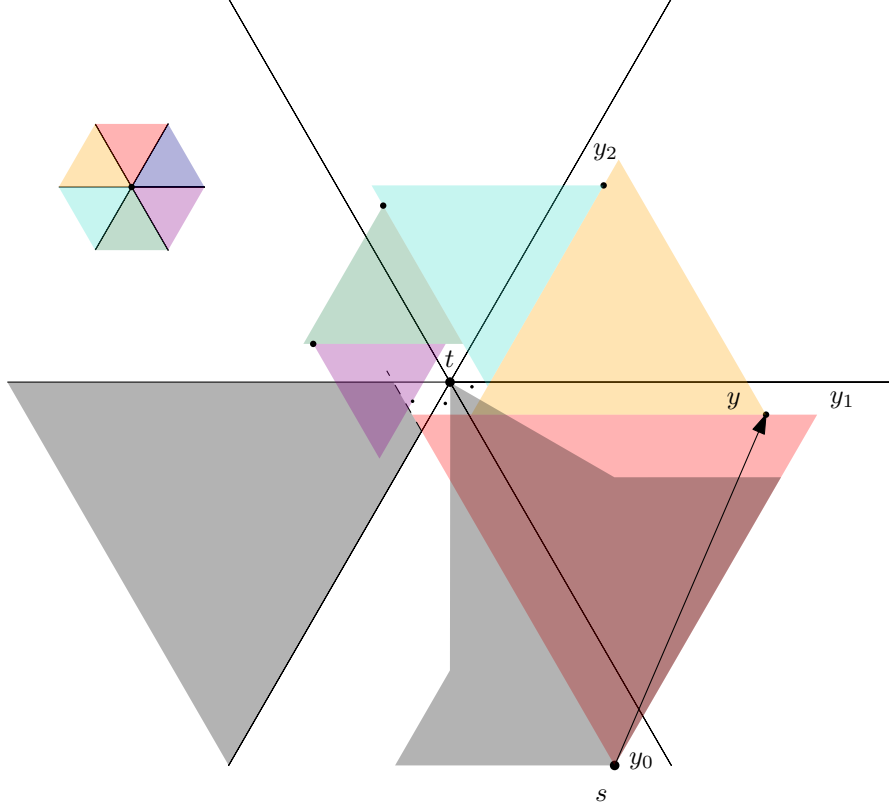


Fig. 6.17: When Assumption 1 does not hold, then the shaded region in C_t^5 does not contain vertices of $\pi(y, t)_{v^y}$.

Then $\|u^y t\|_{\odot} \leq y_2$, so by Lemma 9, $\|v^y t\|_{\odot} \leq \frac{y_2}{2}$. In addition, $u^y \in C_s^3$, hence we also have $\|v^y t\|_{\odot} \leq y_1$. We can explicitly upper bound Y_i as follows:

$$\begin{aligned}
 5 - 2.5y_0 &< d_G(s, t) \\
 &\leq \|sy\|_{\odot} + \|\pi(y, t)_{v^y}\|_{\odot} + d_G(v^y, t) \\
 &\leq (1 - y_1) + (Y_1 + Y_2 + Y_3 + Y_4 + Y_5 + Y_0) + 5\|v^y t\|_{\odot} \\
 &\leq (1 - y_1) + (1 - y_0) + (1 - y_0 - y_1) + (1 - y_0 - y_1 - y_2) \\
 &\quad + (1 - y_0 - y_1 - y_2) + y_0 + y_1 + 5 \min(y_1, \frac{y_2}{2}). \tag{6.21}
 \end{aligned}$$

It can be verified that the linear system (6.2, 6.21) is infeasible.

Case $v \in C_t^1$: Then $u^y \in C_t^0$. Since Assumption 1 does not hold, then $u^y \in C_s^3$, forcing $\|u^y t\|_{\odot} \leq y_1$. By Lemma 9, we have $\|v^y t\|_{\odot} \leq \frac{y_1}{2}$.

$$\begin{aligned}
 5 - 2.5y_0 &< d_G(s, t) \\
 &\leq \|sy\|_{\odot} + \|\pi(y, t)_{v^y}\|_{\odot} + d_G(v^y, t) \\
 &\leq (1 - y_1) + (Y_1 + Y_2 + Y_3 + Y_4 + Y_5 + Y_0) + 5\|v^y t\|_{\odot} \\
 &\leq (1 - y_1) + (1 - y_0) + (1 - y_0 - y_1) + 2(1 - y_0 - y_1 - y_2) + y_0 + y_1 + 5 \frac{y_1}{2}. \tag{6.22}
 \end{aligned}$$

It can be verified that the linear system (6.2,6.22) is infeasible, concluding this section.

Proving Theorem 4 using only Assumption 1

This section considers the case when Assumption 1 holds but Assumption 2 does not. Then there exists a vertex u in $T := C_t^0 \cap C_s^2$ with no vertices strictly in $C_u^4 \cap C_t^0$. Let u^0 be the vertex in T minimizing $\|u^0 t\|_{\text{hex}}$. Note that $u^0 := s$ if the interior of T is empty. Let $h := 1 - \|u^0 t\|_{\text{hex}}$ and $d := \|su^0\| - h$. See Figure 6.18. Since $u^0 \in T$, we have

$$d + h \leq y_0. \quad (6.23)$$

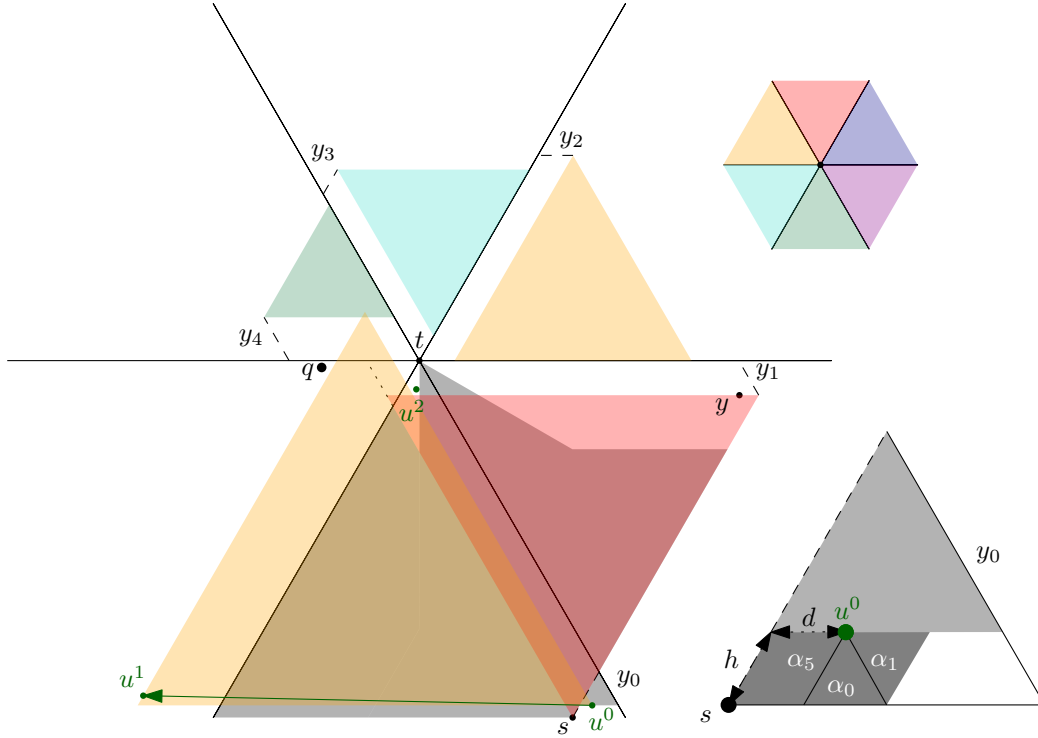


Fig. 6.18: In this example, Assumption 1 holds, whereas Assumption 2 does not hold. When $C_s^4 \cap C_t^0$ is empty, then we form a path through s, u^0, u^1, u^2, t .

We proceed depending on whether the greedy path $\pi(s, u^0)$ enters the cone $C_{u^0}^2$.

Case $C_{u^0}^2 \cap \pi(s, u^0) = \emptyset$: Let α_i be the maximum value of $\|u^0 u\|_{\text{hex}}$ over all vertices $u \in \pi(s, u^0) \cap C_{u^0}^i$. Considering the empty regions around u^0 we have $\alpha_5 \leq h + d, \alpha_0 \leq h, \alpha_1 \leq h$, and $\alpha_i = 0$ for $i \in \{2, 3, 4\}$. By Assumption 1, there must exist $q \in C_s^4 \cap \pi(y, t)_{vy}$. Notice that we must also have $q \in C_{u^0}^4$ since $\|u^0 t\|_{\text{hex}} \geq 1 - y_0 \geq \|yt\|_{\text{hex}} \geq \|qt\|_{\text{hex}}$. Therefore we may consider

the edge $u^0 u^1$ in $C_{u^0}^4$. See Figure 6.18. By defining $m := \|u^0 u^1\|_{\ominus}$ and $p := R_t^0 \cap R_s^1$, we obtain

$$\begin{aligned}
(y_0 - d) + m &\leq \|p u^0\|_{\ominus} + \|u^0 q\|_{\ominus} && \text{since } \|u^0 u^1\|_{\ominus} \leq \|u^0 q\|_{\ominus}, \\
&= \|p q\|_{\ominus} && \text{by } \|\cdot\|_{\ominus} \text{ collinearity,} \\
&\leq \|p t\|_{\ominus} + \|t q\|_{\ominus} && \text{by triangle inequality,} \\
&\leq 1 + (1 - y_0 - y_1 - y_2 - y_3 - y_4) && \text{since } q \in \pi(y, t)_{v^y} \cap C_t^5. \tag{6.24}
\end{aligned}$$

Let u^2 be the first point of $\pi(u^1, t)$ in C_t^0 . Then necessarily $u^2 \in C_s^3$, so $\|u^2 t\|_{\ominus} \leq y_1$. Let β_i be the maximum value of $\|u t\|_{\ominus}$ over all vertices $u \in \pi(u^1, t)_{u^2} \cap C_t^i$. By Lemma 8, we have

$$\begin{aligned}
5 - 2.5y_0 &< d_G(s, t) \\
&\leq \|\pi(s, u^0)\|_{\ominus} + \|u^0 u^1\|_{\ominus} + \|\pi(u^1, t)_{u^2}\|_{\ominus} + d_G(u^2, t) \\
&\leq (\alpha_5 + \alpha_0 + \alpha_1) + m + (\beta_5 + \beta_4 + \beta_3 + \beta_2 + \beta_1) + 5\|u^3 t\|_{\ominus} \\
&\leq (h + d) + h + h + m + (y_0 - h - d + m) + (y_4 + y_0 - h - d) + y_4 + y_3 + y_2 + 5y_1. \tag{6.25}
\end{aligned}$$

However the system (6.2, 6.23, 6.24, 6.25) is infeasible. Notice that 6.25 holds even if $u^1 \in C_t^4$. We re-bracket for clarity:

$$\begin{aligned}
5 - 2.5y_0 &< d_G(s, t) \\
&\leq \|\pi(s, u^0)\|_{\ominus} + \|u^0 u^1\|_{\ominus} + \|\pi(u^1, t)_{u^2}\|_{\ominus} + d_G(u^2, t) \\
&\leq (\alpha_5 + \alpha_0 + \alpha_1) + m + (\beta_5 + \beta_4 + \beta_3 + \beta_2 + \beta_1) + 5\|u^3 t\|_{\ominus} \\
&\leq (h + d) + h + h + m + (y_0 - h - d) + (m + y_4 + y_0 - h - d) + y_4 + y_3 + y_2 + 5y_1.
\end{aligned}$$

This concludes the case when $C_{u^0}^2 \cap \pi(s, u^0) = \emptyset$.

Case $C_{u^0}^2 \cap \pi(s, u^0) \neq \emptyset$: Then let v^0 be the first vertex of $\pi(s, u^0)$ in $C_{u^0}^2$. Next, let v^1 be the first vertex of $\pi(v^0, t)$ in C_t^0 , and define T_i to be the maximum value of $\|u t\|_{\ominus}$ over all vertices $u \in \pi(v^0, t)_{v^1} \cap C_t^i$. Notice that $\|v^0 v^1\|_{\ominus} \leq h$ and $u^0 \in C_t^0$. Therefore $T_i \leq h$ for $i \in \{2, 3, 4, 5\}$. Furthermore, we have $T_1 = \|v^0 t\|_{\ominus} \leq \|v^0 u^0\|_{\ominus} + \|u^0 t\|_{\ominus} \leq h + (1 - h) = 1$. We have

$$\begin{aligned}
5 - 2.5y_0 &< d_G(s, t) \\
&\leq \|\pi(s, u^0)_{v^0}\|_{\ominus} + \|\pi(v^0, t)_{v^1}\|_{\ominus} + d_G(v^1, t) \\
&\leq (\alpha_5 + \alpha_0 + \alpha_1) + (T_1 + T_2 + T_3 + T_4 + T_5) + 5\|v^1 t\|_{\ominus} \\
&\leq (h + d) + h + h + 1 + h + h + h + h + 5y_1. \tag{6.26}
\end{aligned}$$

However the system (6.2, 6.23, 6.24, 6.26) is infeasible. This concludes the proof.

Remaining Possibilities under Assumptions 1 and 2

Recall from Section 6.5 that we consider the 20 possibilities from the space $\{Y_0, Y_1\} \times \{X_0, X_1\} \times \{(6.18), J_2, J_3, J_4, J_5\}$. In each of these possibilities, the inequalities (6.2, 6.4, 6.5, 6.6, 6.7) hold.

However, only 16 of the 20 possibilities have infeasible systems. We therefore consider the remaining four cases more carefully: (X_1, Y_1, J_2) , (X_1, Y_1, J_3) , (X_1, Y_1, J_4) , (X_0, Y_0, J_3) .

Case (X_1, Y_1, J_2) : First, if $\|v^y t\|_{\odot} \leq \frac{y_1}{2}$, then $\|v^y t\|_{\odot} \leq \frac{y_1}{2}$ since $\|v^y t\|_{\odot} \leq \|v^y t\|_{\ominus}$. Including this new inequality yields an infeasible system. Otherwise, if $\|v^y t\|_{\ominus} > \frac{y_1}{2}$, then Lemma 9 implies $\|u^y t\|_{\ominus} > y_1$, meaning that $u^y \in C_s^4 \cap C_y^5$. See Figure 6.19. Let $\ell := \|C_s^4 \cap C_{u^y}^2\|$. Since

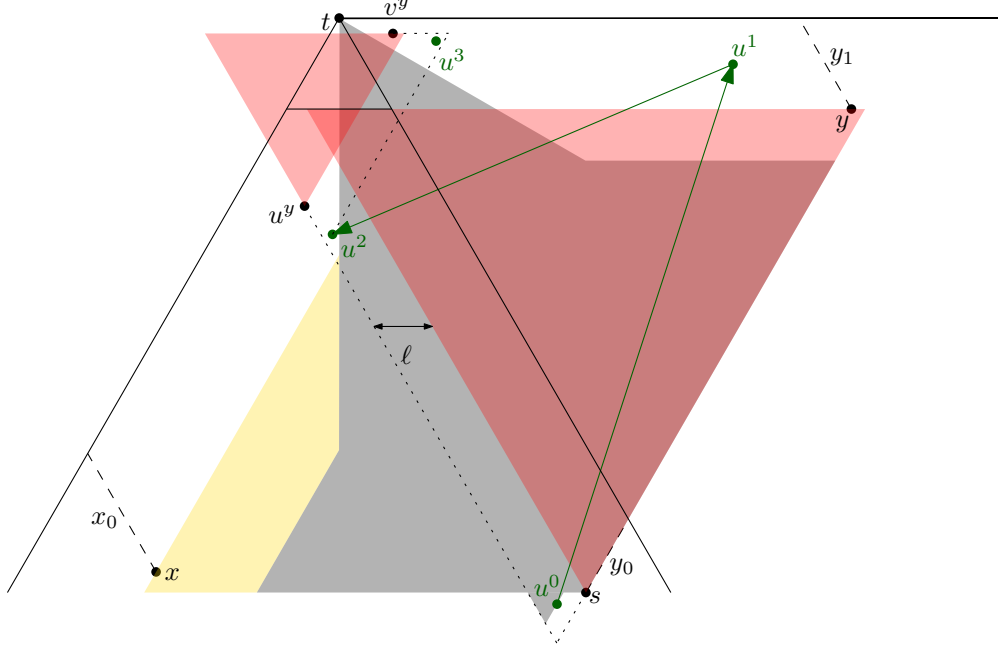


Fig. 6.19: Here we illustrate the case X_1, Y_1 when $\|u^y t\|_{\ominus} > y_1$. This implies $u^y \in C_y^5$, which allows a shortcut to be formed through vertices s, u^0, u^1, u^2, u^3, t .

$u^y v^y$ is an edge and $u^y \in C_t^0$, we must have

$$Y_0 \geq \|u^y v^y\|_{\ominus} \geq y_0 + \ell + \frac{\|v^y t\|_{\ominus}}{2} \geq y_0 + \ell + \frac{\|v^y t\|_{\odot}}{2}. \quad (6.27)$$

Now we describe how u^y provides a shortcut to t . Let u^0 be the vertex in the triangle $C_{u^y}^1 \cap C_s^5$ that minimizes $\|u^0 u^y\|_{\ominus}$. If this triangle is empty, then $u^0 := s$. Notice that $\|s u^0\|_{\ominus} \leq \ell$. Therefore we use induction to find a path from s to u^0 , then follow the edge in $C_{u^0}^3$ to a vertex u^1 . Notice that u^1 cannot be in C_t^5 from the empty region of edge $u^y v^y$. If $\|u^1 t\|_{\ominus} \leq x_0$, then apply induction (below set $u^2 := u^1, u^3 := u^1$). Otherwise if $u^1 \in C_t^0$, then let $u^2 := u^1$. If instead $u^1 \in C_t^1$, then follow $C_{u^1}^5$ to a vertex $u^2 \in C_t^0$. Notice that u^y guarantees the existence of u^2 and the empty region from edge $u^y v^y$ guarantees $u^2 \in C_t^0$. Finally, from u^2 , follow $\pi(u^2, t)$ until reaching a vertex $u^3 \in C_t^1$. Again, the empty triangle $\nabla_{u^y}^{v^y}$ guarantees that $u^3 \in C_t^1$. We have

$$\begin{aligned} 5 - 2.5y_0 &< d_G(s, t) \\ &\leq d_G(s, u^0) + \|u^0 u^1\|_{\ominus} + \|u^1 u^2\|_{\ominus} + \|\pi(u^2, t)_{u^3}\|_{\ominus} + d_G(u^3, t) \\ &\leq 5\ell + (1 + \ell) + (1 + \ell) + (y_0 + \ell + x_0) + 5x_0. \end{aligned} \quad (6.28)$$

It can be verified that the system of case (X1, Y1, J2) is infeasible with the additional inequalities (6.27) and (6.28) in the presence of (6.2, 6.4, 6.5, 6.6, 6.7).

Case (X1, Y1, J3): We use the exact same argument as for (X1, Y1, J2). A careful scan of the proof reveals that the argument only uses Y1 for structure.

Case (X1, Y1, J4): Again, we use the exact same argument as for (X1, Y1, J2).

Case (X0, Y0, J3): We will describe a symmetric argument to the previous three. First, if $\|v^x t\|_{\text{hex}} \leq \frac{x_0}{2}$, then the system is infeasible since $\|v^x t\|_{\text{hex}} \leq \|v^x t\|_{\text{hex}}$. Otherwise, we must have $\|u^x t\|_{\text{hex}} > x_0$, meaning that $u^x \in C_x^2$. See Figure 6.20. Let $\ell := \|C_t^1 \cap C_{u^x}^5\|$. Since $u^x v^x$ is an

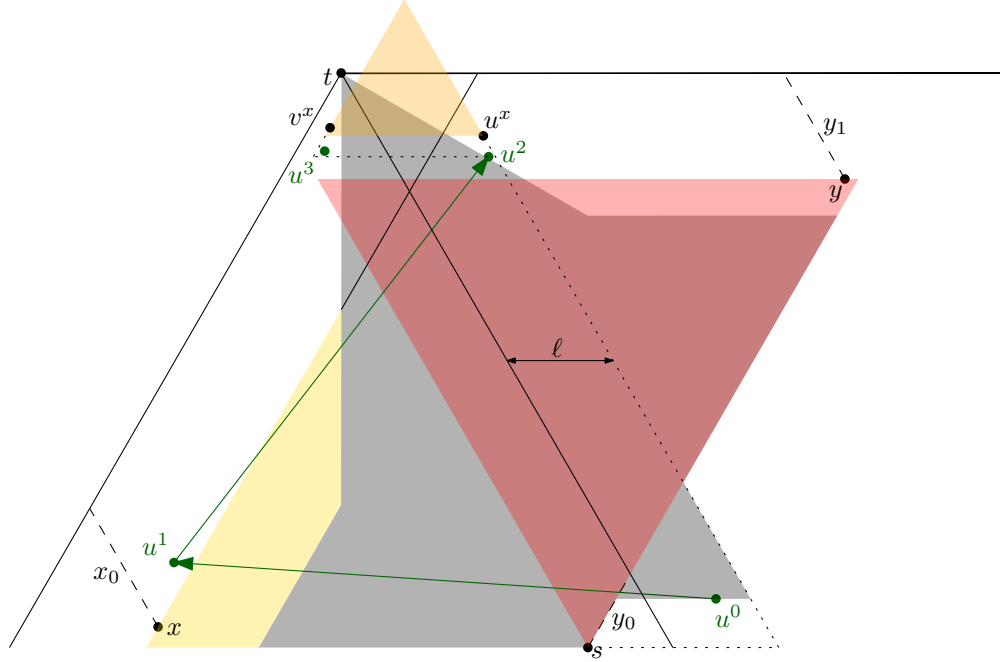


Fig. 6.20: Here we illustrate the case X0,Y0 when $\|u^x t\|_{\text{hex}} > x_0$. This implies $u^x \in C_x^2$, which allows a shortcut to be formed through vertices s, u^0, u^1, u^2, u^3, t .

edge and $u^x \in C_t^1$, we must have

$$X_1 \geq \|u^x v^x\|_{\text{hex}} \geq \ell + \frac{\|v^x t\|_{\text{hex}}}{2} \geq \ell + \frac{\|v^x t\|_{\text{hex}}}{2}. \quad (6.29)$$

Now we describe how to use u^x as a shortcut to get to t . Let u^0 be the vertex in the triangle $C_{u^x}^0 \cap C_s^2$ that minimizes $\|u^0 u^x\|_{\text{hex}}$. If this triangle is empty, then $u^0 := s$. Notice that $\|s u^0\|_{\text{hex}} \leq \ell + y_0$. Therefore we use induction to find a path from s to u^0 , then follow the edge in $C_{u^0}^4$ to a vertex u^1 . Notice that we cannot have $u^2 \in C_t^5$ by Assumption 2 and we cannot have $u^2 \in C_t^2$ by the empty region of $u^x v^x$. If $\|u^1 t\|_{\text{hex}} \leq y_1$, then apply induction (below set $u^2 := u^1, u^3 := u^1$). If instead $u^1 \in C_t^1$, then let $u^2 := u^1$. If instead $u^1 \in C_t^0$, then follow $C_{u^1}^2$ to a vertex $u^2 \in C_t^1$. Finally, from u^2 , follow $\pi(u^2, t)$ until reaching a vertex $u^3 \in C_t^0$. By the

empty triangle $\nabla_{u^x}^{v^x}$, vertex u^3 is well-defined. We have

$$\begin{aligned}
5 - 2.5y_0 &< d_G(s, t) \\
&\leq d_G(s, u^0) + \|u^0 u^1\|_{\odot} + \|u^1 u^2\|_{\odot} + \|\pi(u^2, t)_{u^3}\|_{\odot} + d_G(u^3, t) \\
&\leq 5(\ell + y_0) + (1 + \ell) + (1 + \ell) + (\ell + y_1) + 5y_1.
\end{aligned} \tag{6.30}$$

It can be verified that the system of case (Xo, Yo, J3) is infeasible with the additional inequalities (6.29) and (6.30) in the presence of (6.2, 6.4, 6.5, 6.6, 6.7).

6.6 Lower Bound on the Routing Ratio of the $\vec{\Theta}_6$ -Graph

In Theorem 26 of [1], the authors also state a lower bound of 6 on the routing ratio of any local routing algorithm for directed $\vec{\Theta}_6$ -graphs. However, we noticed a mistake in the proof, where the construction simultaneously requires there to be an edge from t to r , as well as a vertex $r' \in \text{int}(\nabla_t^r) = \emptyset$. With some minor changes, the argument provided can yield a lower bound of 5 on the routing ratio, however our Lemma 1 already proves a lower bound of 5 on the spanning ratio. We give an alternate construction that proves a lower bound of 6 on the routing ratio. This means that the statement of Theorem 26 in [1] remains true, and only the proof needs revision.

Lemma 27. *For any $\epsilon > 0$ and any local routing algorithm A for $\vec{\Theta}_6$ -graphs, the routing ratio of A is at least $6 - \epsilon$.*

Proof. Let $\epsilon > 0$ be arbitrarily small, and let A be a local routing algorithm. We will construct two directed Θ_6 -graphs G_1 and G_2 such that A will have a routing ratio of at least $6 - \epsilon$ in G_1 or G_2 . See Figure 6.21.

In the following construction, we reuse many of the same vertices from the proof of Lemma 1, while additionally defining $d := (1, 0) + \delta(-8, -1)$ and $f := (-\frac{1}{2}, \frac{\sqrt{3}}{2}) + \delta(0, -4)$. Let R be the linear map corresponding to a rotation of $\pi/3$ radians in the counter-clockwise direction, meaning $R(1, 0) := (\cos(\pi/3), \sin(\pi/3))$ and $R(0, 1) := (-\sin(\pi/3), \cos(\pi/3))$. Let X be the linear map corresponding to a reflection about the x -axis, meaning $X(1, 0) := (1, 0)$ and $X(0, 1) := (0, -1)$. For ease of exposition, we adopt the notation that a map M applied to a point p results in the point Mp . Define the following point sets

$$\begin{aligned}
P_1 &:= \{t, s, f, Xf, Rc, X Rc, d\} \cup \{Rp^i \mid 0 \leq i \leq k\} \cup \{Rq^i \mid 0 \leq i < k\}, \\
P_2 &:= \{t, s, f, Xf, Rc, X Rc, Xd\} \cup \{XRp^i \mid 0 \leq i \leq k\} \cup \{XRq^i \mid 0 \leq i < k\}.
\end{aligned}$$

Let $G_1 := \vec{\Theta}_6(P_1)$ and $G_2 := \vec{\Theta}_6(P_2)$. Consider a local routing algorithm A routing from s to t . Assume that in G_i , A produces a path P_i from s to t . Since the neighbourhoods of vertices s, f, Xf are identical in G_1 and G_2 , then P_1 and P_2 follow the same path until visiting either Rc or $X Rc$. Intuitively, if Rc is visited first, then P_2 has length at least $6 - \epsilon$ since $d_{G_2}(s, Rc) \approx 2$ and $d_{G_2}(Rc, t) \approx 4$. On the other hand if $X Rc$ is visited first, then P_1 has length at least $6 - \epsilon$ since $d_{G_1}(s, X Rc) \approx 2$ and $d_{G_1}(X Rc, t) \approx 4$.

We will proceed to justifying the above approximations. Consider the outgoing edge from d in G_1 in the cone C_d^4 . This edge must have target Rp^j for some j , and we will prove that the edge dRp^j has length arbitrarily close to 1.

Observation 6. $\|Rp^j(\frac{1}{2}, \frac{\sqrt{3}}{2})\|_2 \leq 24\delta$. First define $d' := R_t^2 \cap R_d^3$. Note that $\|d'(\frac{1}{2}, \frac{\sqrt{3}}{2})\|_2 \leq 2\|d(1, 0)\|_2 \leq 2(9\delta)$. Next, $\|Rp^j d'\|_2 \leq 6\delta$. The claim is completed by the triangle inequality.

By an argument analogous to the proof of Lemma 1, we observe that in G_1 , the edges $(s, f), (s, Xf), (f, Rc), (Xf, XRc), (XRc, d), (d, Rp^j)$ all have length arbitrarily close to 1. Moreover, the edges $(f, Xf), (Rc, s), (XRc, s), (d, Xf), (Rp^j, XRc)$ all have length strictly larger than 1. Combining this with $d_{G_1}(XRc, t) = d_{G_1}(XRc, Rp^j) + d_{G_1}(Rp^j, t) \approx 2 + 2 = 4$ and $d_{G_2}(Rc, t) = d_{G_2}(Rc, Rp^j) + d_{G_2}(Rp^j, t) \approx 2 + 2 = 4$ completes the proof. $\square$

6.7 Conclusion

This chapter proves a tight bound of 5 on the spanning ratio of the $\vec{\Theta}_6$ -graph. It appears to be the first directed geometric graph for which an exact spanning ratio is known, as other tight bounds on geometric graphs have all pertained to naturally undirected graphs [9, 14, 15, 24, 27, 36, 40, 42, 43]. Interestingly, our lower bound of 6 on the routing ratio confirms that it is impossible to design a local routing algorithm to find the shortest path in arbitrary $\vec{\Theta}_6$ -graphs. We leave it as an open problem to close the gap on the routing ratio of the $\vec{\Theta}_6$ -graph, which currently lies between 6 and 8 [1].

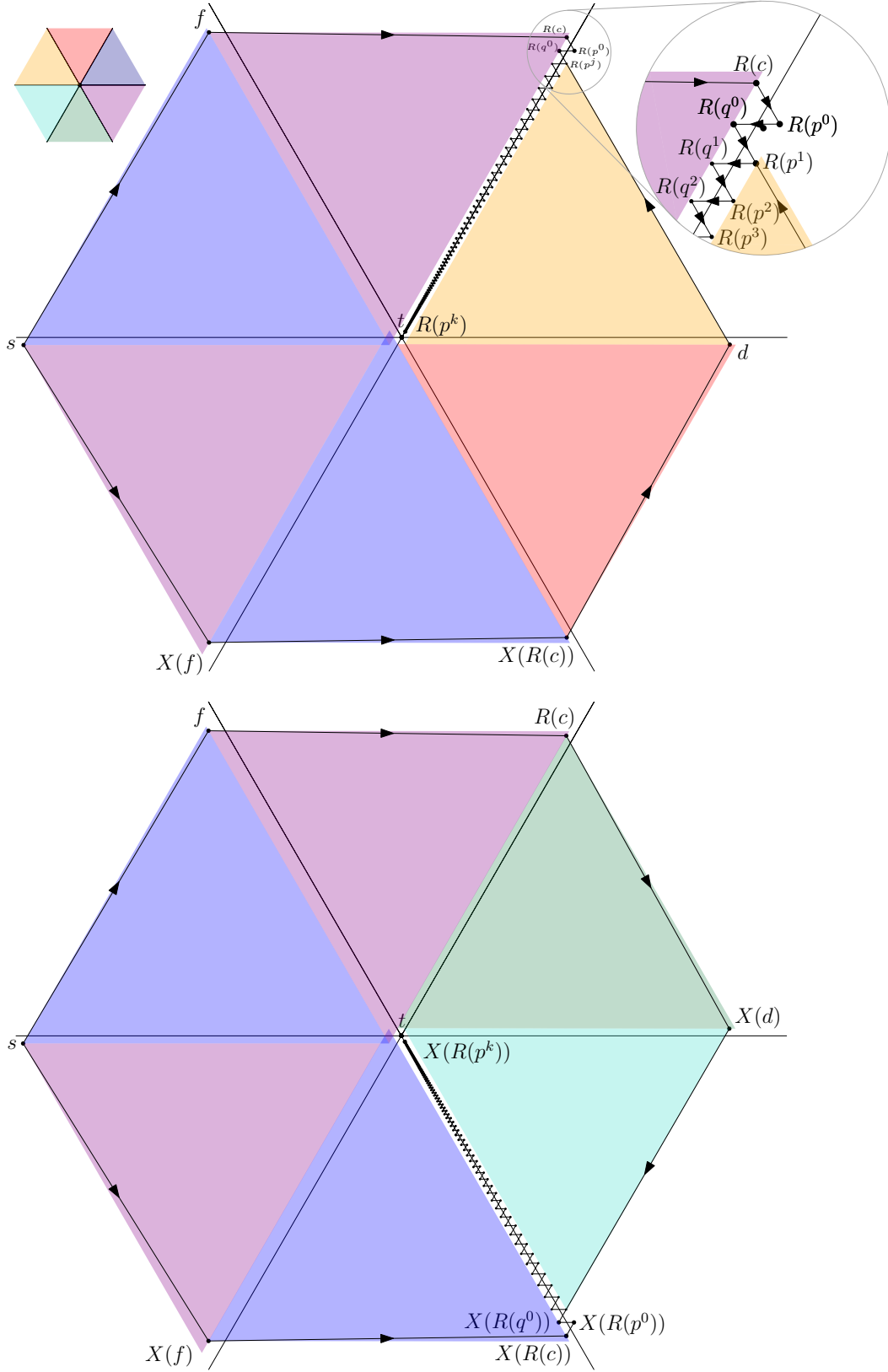


Fig. 6.21: Graph constructions achieving a routing ratio lower bound of $6 - \epsilon$.

Bibliography

- [1] Hugo A. Akitaya, Ahmad Biniaz, and Prosenjit Bose. On the spanning and routing ratios of the directed Θ_6 -graph. *Comput. Geom.*, 105-106:101881, 2022.
- [2] Vikrant Ashvinkumar, Joachim Gudmundsson, Christos Levkopoulos, Bengt J. Nilsson, and André van Renssen. Local routing in sparse and lightweight geometric graphs. *Algorithmica*, 84(5):1316–1340, 2022.
- [3] Luis Barba, Prosenjit Bose, Jean-Lou De Carufel, André van Renssen, and Sander Verdonschot. On the stretch factor of the theta-4 graph. In *WADS*, volume 8037 of *Lecture Notes in Computer Science*, pages 109–120. Springer, 2013.
- [4] Luis Barba, Prosenjit Bose, Mirela Damian, Rolf Fagerberg, Wah Loon Keng, Joseph O'Rourke, André van Renssen, Perouz Taslakian, Sander Verdonschot, and Ge Xia. New and improved spanning ratios for yao graphs. *J. Comput. Geom.*, 6(2):19–53, 2015.
- [5] Nicolas Bonichon, Prosenjit Bose, Jean-Lou De Carufel, Vincent Despré, Darryl Hill, and Michiel Smid. Improved routing on the Delaunay triangulation. *Discret. Comput. Geom.*, 70(3):495–549, 2023.
- [6] Nicolas Bonichon, Prosenjit Bose, Jean-Lou De Carufel, Ljubomir Perkovic, and André van Renssen. Upper and lower bounds for online routing on Delaunay triangulations. *Discret. Comput. Geom.*, 58(2):482–504, 2017.
- [7] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and David Ilcinkas. Connections between theta-graphs, Delaunay triangulations, and orthogonal surfaces. In *WG*, volume 6410 of *Lecture Notes in Computer Science*, pages 266–278, 2010.
- [8] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and Ljubomir Perkovic. Plane spanners of maximum degree six. In *ICALP (1)*, volume 6198 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 2010.
- [9] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and Ljubomir Perkovic. Tight stretch factors for L_1 - and L_∞ -Delaunay triangulations. *Comput. Geom.*, 48(3):237–250, 2015.
- [10] Nicolas Bonichon, Iyad A. Kanj, Ljubomir Perkovic, and Ge Xia. There are plane spanners of degree 4 and moderate stretch factor. *Discret. Comput. Geom.*, 53(3):514–546, 2015.

- [11] Prosenjit Bose, Paz Carmi, Sébastien Collette, and Michiel H. M. Smid. On the stretch factor of convex Delaunay graphs. *J. Comput. Geom.*, 1(1):41–56, 2010.
- [12] Prosenjit Bose, Jean-Lou De Carufel, Stephane Durocher, and Perouz Taslakian. Competitive online routing on Delaunay triangulations. *Int. J. Comput. Geom. Appl.*, 27(4):241–254, 2017.
- [13] Prosenjit Bose, Jean-Lou De Carufel, Darryl Hill, and Michiel Smid. On the spanning and routing ratio of the directed theta-four graph. *Discret. Comput. Geom.*, 71(3):872–892, 2024.
- [14] Prosenjit Bose, Jean-Lou De Carufel, Pat Morin, André van Renssen, and Sander Verdonschot. Towards tight bounds on theta-graphs: More is not always better. *Theor. Comput. Sci.*, 616:70–93, 2016.
- [15] Prosenjit Bose, Jean-Lou De Carufel, and Sandrine Njoo. The exact spanning ratio of the parallelogram delaunay graph. *Theor. Comput. Sci.*, 1037:115152, 2025.
- [16] Prosenjit Bose, Jean-Lou De Carufel, and John Stuart. Routing from pentagon to octagon Delaunay graphs. In *ISAAC*, volume 322 of *LIPICs*, pages 14:1–14:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.
- [17] Prosenjit Bose, Jean-Lou De Carufel, and John Stuart. Online routing in directed Yao₄[∞] graphs. In Pat Morin and Eunjin Oh, editors, *19th International Symposium on Algorithms and Data Structures, WADS 2025, York University, Toronto, Canada, August 11-15, 2025*, volume 349 of *LIPICs*, pages 9:1–9:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.
- [18] Prosenjit Bose, Jean-Lou De Carufel, John Stuart, and Darryl Hill. The spanning ratio of the directed $\vec{\Theta}_6$ graph is 5. In Hee-Kap Ahn, Michael Hoffmann, and Amir Nayyeri, editors, *42nd International Symposium on Computational Geometry, SoCG 2026, New Brunswick, NJ, USA, June 2-5, 2026*, volume 367 of *LIPICs*, pages 20:1–20:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026.
- [19] Prosenjit Bose, Mirela Damian, Karim Douïeb, Joseph O’Rourke, Ben Seamone, Michiel H. M. Smid, and Stefanie Wührer. $\pi/2$ -angle Yao graphs are spanners. *Int. J. Comput. Geom. Appl.*, 22(1):61–82, 2012.
- [20] Prosenjit Bose, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. Optimal local routing on Delaunay triangulations defined by empty equilateral triangles. *SIAM J. Comput.*, 44(6):1626–1649, 2015.
- [21] Prosenjit Bose, Darryl Hill, Michiel Smid, and Tyler Tuttle. On the spanning and routing ratios of the yao-four graph. In *ISAAC*, volume 322 of *LIPICs*, pages 15:1–15:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.
- [22] Prosenjit Bose and Pat Morin. Competitive online routing in geometric graphs. *Theor. Comput. Sci.*, 324(2-3):273–288, 2004.

- [23] Prosenjit Bose, Pat Morin, André van Renssen, and Sander Verdonschot. The theta 5-graph is a spanner. In Andreas Brandstädt, Klaus Jansen, and Rüdiger Reischuk, editors, *Graph-Theoretic Concepts in Computer Science - 39th International Workshop, WG 2013, Lübeck, Germany, June 19-21, 2013, Revised Papers*, volume 8165 of *Lecture Notes in Computer Science*, pages 100–114. Springer, 2013.
- [24] Prosenjit Bose and André van Renssen. Upper bounds on the spanning ratio of constrained theta-graphs. In *LATIN*, volume 8392 of *Lecture Notes in Computer Science*, pages 108–119. Springer, 2014.
- [25] Kevin Buchin, Joachim Gudmundsson, Antonia Kalb, Aleksandr Popov, Carolin Rehs, André van Renssen, and Sampson Wong. Oriented spanners. In *ESA*, volume 274 of *LIPICs*, pages 26:1–26:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.
- [26] Paul Chew. There is a planar graph almost as good as the complete graph. In *SCG*, pages 169–177. ACM, 1986.
- [27] Paul Chew. There are planar graphs almost as good as the complete graph. *J. Comput. Syst. Sci.*, 39(2):205–219, 1989.
- [28] Kenneth L. Clarkson. Approximation algorithms for shortest path motion planning (extended abstract). In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, STOC '87, pages 56–65, New York, NY, USA, 1987. ACM.
- [29] Mirela Damian and Naresh Nelavalli. Improved bounds on the stretch factor of y_4 . *Comput. Geom.*, 62:14–24, 2017.
- [30] Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational geometry: algorithms and applications, 3rd Edition*. Springer, 2008.
- [31] David P. Dobkin, Steven J. Friedman, and Kenneth J. Supowit. Delaunay graphs are almost as good as complete graphs. *Discret. Comput. Geom.*, 5:399–407, 1990.
- [32] William Michael Zoltan Kalnay. Routing ratio of the directed Yao-6 graphs, 2023.
- [33] J. Mark Keil and Carl A. Gutwin. Classes of graphs which approximate the complete euclidean graph. *Discret. Comput. Geom.*, 7(1):13–28, dec 1992.
- [34] Hung Le, Lazar Milenkovic, Shay Solomon, and Tianyi Zhang. Covering the euclidean plane by a pair of trees. *CoRR*, abs/2508.11507, 2025.
- [35] Hung Le, Lazar Milenkovic, Shay Solomon, and Tianyi Zhang. Covering the euclidean plane by a pair of trees. In Kasper Green Larsen and Barna Saha, editors, *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2026, Vancouver, BC, Canada, January 11-14, 2026*, pages 3472–3497. SIAM, 2026.

- [36] Anna Lubiw and Debajyoti Mondal. Construction and local routing for angle-monotone graphs. *J. Graph Algorithms Appl.*, 23(2):345–369, 2019.
- [37] Lihong Ma. *Bisectors and Voronoi Diagrams for Convex Distance Functions*. Phd thesis, FernUniversität Hagen, Germany, April 2000.
- [38] Lihong Ma. *Bisectors and Voronoi Diagrams for Convex Distance Functions*, volume 267 of *Informatik-Berichte*. FernUniversität in Hagen, Hagen, 2000. Zugl.: Dissertation, FernUniversität in Hagen, 2000.
- [39] Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.
- [40] Ljubomir Perkovic, Michael Dennis, and Duru Türkoglu. The stretch factor of hexagon-Delaunay triangulations. *J. Comput. Geom.*, 12(2):86–125, 2021.
- [41] J. Ruppert and R. Seidel. Approximating the d-dimensional complete euclidean graph. page 207–210, 1991.
- [42] John Stuart, Jean-Lou De Carufel, and Prosenjit Bose. The exact routing and spanning ratio of arbitrary triangle delaunay graphs. In Rahnuma Islam Nishat, editor, *Proceedings of the 36th Canadian Conference on Computational Geometry, CCCG 2024, Brock University, St. Catharines, Canada, July 17-19, 2024*, pages 83–89, 2024.
- [43] André van Renssen, Yuan Sha, Yucheng Sun, and Sampson Wong. The tight spanning ratio of the rectangle Delaunay triangulation. In *ESA*, volume 274 of *LIPIcs*, pages 99:1–99:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.
- [44] Ge Xia. The stretch factor of the Delaunay triangulation is less than 1.998. *SIAM J. Comput.*, 42(4):1620–1659, 2013.
- [45] Ge Xia and Liang Zhang. Toward the tight bound of the stretch factor of delaunay triangulations. In *Proceedings of the 23rd Annual Canadian Conference on Computational Geometry, Toronto, Ontario, Canada, August 10-12, 2011*, 2011.
- [46] Andrew Chi-Chih Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J. Comput.*, 11(4):721–736, 1982.