Online Routing in Directed $\overrightarrow{\text{Yao}}_4^\infty$ Graphs

Prosenjit Bose
School of Computer Science, Carleton University, Ottawa, Canada

Jean-Lou De Carufel
School of Electrical Engineering and Computer Science, University of Ottawa, Canada

John Stuart
School of Electrical Engineering and Computer Science, University of Ottawa, Canada

Abstract

The $\overrightarrow{\text{Yao}}_4^\infty$ and $\overrightarrow{\text{Yao}}_4$ 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 $\overrightarrow{\text{Yao}}_4$ graph, whereas with the L_∞-norm, we obtain the $\overrightarrow{\text{Yao}}_4^\infty$ graph, which is a sub-graph of the well-known L_∞-Delaunay graph.

In this paper, we provide a local routing algorithm with routing ratio at most 85.22 for $\overrightarrow{\text{Yao}}_4^\infty$ graphs. Prior to this work, no constant spanning or routing ratios for $\overrightarrow{\text{Yao}}_4^\infty$ graphs were previously known. Now, $\overrightarrow{\text{Yao}}_4^\infty$ graphs are the sparsest family of directed planar graphs supporting a competitive local routing strategy. Furthermore, we show that no local routing algorithm for $\overrightarrow{\text{Yao}}_4^\infty$ 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. 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 $\overrightarrow{\text{Yao}}_4$, respectively.

2012 ACM Subject Classification Theory of computation → Sparsification and spanners

Keywords and phrases Geometric Spanners, Yao Graphs, Local Routing Algorithms

Digital Object Identifier 10.4230/LIPIcs.WADS.2025.9

Funding This research is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

1 Introduction

A fundamental problem in geometric routing is to construct a directed planar graph on a given set of points that supports competitive local routing, using the fewest number of edges. A routing algorithm is competitive if the length of the path that it finds is at most a constant times the length of the shortest path, however 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 planar graphs. Disallowing bidirectional edges presents different challenges and has led to the study of oriented spanners [12]. Note that every undirected graph can be converted into a directed graph by replacing every

19th International Symposium on Algorithms and Data Structures (WADS 2025).
Editors: Pat Morin and Eunjin Oh; Article No. 9; pp. 9:1–9:22
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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 planar 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 $\overrightarrow{\text{Yao}}_4^\infty$ graph stems from its unique combination of properties: it is planar, has a bounded out-degree of 4, and supports a novel competitive local routing algorithm.

For decades, researchers have been studying the distance-preserving properties of geometric graphs, such as the Delaunay triangulation [17]. 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 [15]. 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 [18]. 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 [17]. 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 Yao_4 graph was first proved to be a spanner by Bose et al. [9] with a stretch factor of approximately 662. As part of the proof, the authors established that the undirected Yao_4^∞ graph is an 8-spanner. Then, Bonichon et al. [6] proved that the Yao_4^∞ graph has spanning ratio at most 6.31. Later, Damian and Nelavalli [14] improved both results by showing that the Yao_4 and Yao_4^∞ 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 Yao_4^∞ is denoted $\overrightarrow{\text{Yao}}_4^\infty$. Despite being a natural construction, the $\overrightarrow{\text{Yao}}_4^\infty$ 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 [13]. Then, Bonichon et al. [5] 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 Yao_4^∞ 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 planar graphs, Chew gave a local routing algorithm for L_∞-Delaunay graphs in 1986 [13]. An L_∞-Delaunay graph can contain nearly $6|V|$ directed edges in the worst case. Since then, Bonichon et al. [4] defined a number of bounded-degree planar spanners, including G_9 and G_{12}. In [10], local routing algorithms were provided for these graphs. A parametrized construction, denoted $\mathcal{MBDG}$, was used in [2] to attain several lightness properties. In all of these planar graphs, the number of directed edges required is still on the order of $6|V|$. In this paper, we define a local routing algorithm for the $\overrightarrow{\text{Yao}}_4^\infty$ graph, which contains at most $4|V|$ directed edges, which is the fewest number of edges among all planar directed graphs that admit a local routing algorithm.

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, the optimal local routing algorithm for $\overrightarrow{\text{Yao}}_k$ graphs is simply *cone routing*, where each decision is to move to the neighbour in the same cone as the destination. For smaller values of k, the $\overrightarrow{\text{Yao}}_k$ graph is less dense and cone routing can fail to produce short paths. However, [16] provides a local routing algorithm for the $\overrightarrow{\text{Yao}}_6$ graph with routing ratio at most 22.94. The smallest possible k for which the $\overrightarrow{\text{Yao}}_k$ graph is strongly connected is $k = 4$, and Bose et al. [11] recently provided a local routing algorithm achieving a routing ratio of at most 23.36 for $\overrightarrow{\text{Yao}}_4$ 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 [7] 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 $\overrightarrow{\text{Yao}}_4^\infty$ 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 [11] adapted the Greedy/Sweep approach to $\overrightarrow{\text{Yao}}_4$ graphs by replacing the right triangle with a wedge. However, in the case of $\overrightarrow{\text{Yao}}_4^\infty$ 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 85.22 on the routing ratio of our local routing algorithm.

Notice that our routing ratio of 85.22 is also the best known upper bound on the spanning ratio for $\overrightarrow{\text{Yao}}_4^\infty$ graphs. Next, we give a lower bound of $5 + \sqrt{2} \approx 6.41$ on the spanning ratio of $\overrightarrow{\text{Yao}}_4^\infty$ 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 $\overrightarrow{\text{Yao}}_4^\infty$ 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 $\overrightarrow{\text{Yao}}_4$, respectively.

2 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 $\|uv\|_p :=$

Table 1 Comparison of upper bounds on the routing ratio of geometric graphs with bounded out-degree. Routing in $\mathcal{MBDG}$ and G_9 requires extra information to be stored at each vertex.

Reference	Graph	Planar	Max out-degree	Routing Ratio
[11]	$\overrightarrow{\text{Yao}}_4$	No	4	23.36
[16]	$\overrightarrow{\text{Yao}}_6$	No	6	22.94
[3]	$\overrightarrow{\text{Yao}}_k$ $(k \geq 7)$	No	k	$\leq 1/(1-2\sin(\pi/k))$
[7]	$\vec{\Theta}_4$	No	4	17
[1]	$\vec{\Theta}_6$	No	6	8
[8]	$\vec{\Theta}_k$ $(k \geq 7)$	No	k	$\leq 1/(1-2\sin(\pi/k))$
[2]	$\mathcal{MBDG}$	Yes	≥ 20	≥ 9.27
[10]	G_{12}	Yes	12	54.85
[10]	G_9	Yes	9	8.66
This paper	$\overrightarrow{\text{Yao}}_4^\infty$	Yes	4	85.22

$(\|uv\|_x^p + \|uv\|_y^p)^{\frac{1}{p}}$. 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 $\overrightarrow{\text{Yao}}_4^\infty$ graph are well-defined. 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 constant $c \geq 1$, 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 G is the least c for which G is a c-spanner. The spanning ratio of a class of graphs $\mathcal{G}$ is the least c for which all graphs in $\mathcal{G}$ are c-spanners.

We make the assumption that the graph is embedded on a polynomial-sized grid and therefore the coordinates of the points require $O(\log(n))$ bits. Formally, an 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 outgoing 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 $c\|st\|_2$. The routing ratio of an algorithm is the least c for which the algorithm is c-competitive for $\mathcal{G}$. Note that the routing ratio is an upper bound on the spanning ratio.

3 A Local Routing Algorithm for $\overrightarrow{\text{Yao}}_4^\infty$ Graphs

We will now describe our local routing algorithm for $\overrightarrow{\text{Yao}}_4^\infty$ 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 , are the regions close to the diagonals, excluding the green square. More precisely, let refer to the set of points $p \in \mathbb{R}^2$ such that $p \notin \blacksquare$ and $h(p) \leq \delta$. Refer to Figure 1. Note that $\boxtimes \subset \mathbb{R}^2$ and is composed of four regions, which we label $\boxtimes, \boxtimes, \boxtimes, \boxtimes$ clockwise from the north-eastern region. Then define the truncated wedges $\bowtie$ as those points $p \in \mathbb{R}^2$ such that $p \notin \blacksquare \cup \boxtimes$. We have $\bowtie \subset \mathbb{R}^2$ and since $\bowtie$ is composed of four regions, we label them $\bowtie, \bowtie, \bowtie, \bowtie$ clockwise from the northern region.

Next we describe the terminology used in our local routing algorithm, which we will refer to as DIRTY Algorithm 1 (**D**irected **I**nfinity **R**outing **T**hrough **Y**ao). For any vertex $u \in \boxtimes$, a *band step* from u follows the edge in the cone of u containing t. For any vertex $u \in \bowtie$, 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 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 $\textbf{Sweep}(u, t, N(u), \delta)$. 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 $\textbf{Greedy}(u, t, N(u))$. 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 $\boxtimes$ or is clean. When the current vertex is not clean, we take a sweep edge. A partial trace of Algorithm 1 is shown in Figure 1. The constant $0 < c < 1$ in step 1 of Algorithm 1 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 1.

Algorithm 1 DIRTY (**D**irected **I**nfinity **R**outing **T**hrough **Y**ao)
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 $u \in \boxtimes$ or u is clean, then $v := \textbf{Greedy}(u, t, N(u))$. Else, $v := \textbf{Sweep}(u, t, N(u), \delta)$.

3.1 Analysis of DIRTY Algorithm 1

If edge uv is chosen by Algorithm 1, 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 1. If uv is an edge chosen by Algorithm 1, then $\|ut\|_\infty > \|vt\|_\infty$. That is, the vertex v is inside the square centered at t with u on the boundary.

The vertex set being finite means that choosing each successive edge according to Algorithm 1 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 1 of Algorithm 1. 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 $t' \in \blacksquare$. We will focus on upper bounding the length of $\mathcal{P}_{s,t'}$, which is a path such that

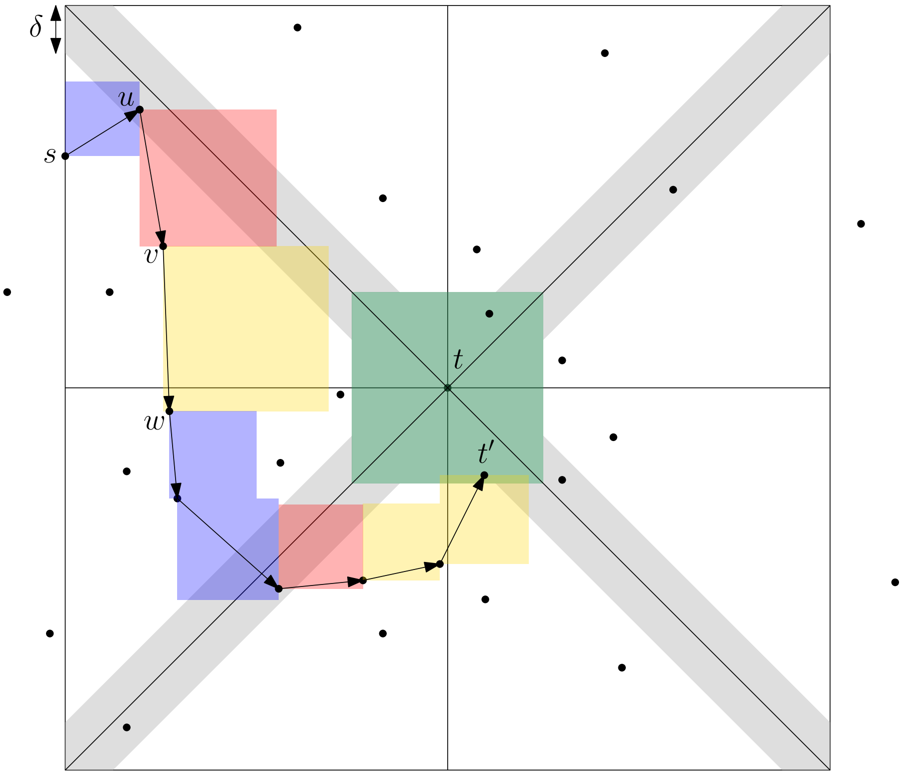

Figure 1 We show the partial trace of DIRTY Algorithm 1 until we reach a vertex in the green square, . The grey regions are the *bands*, denoted as , and the white regions are the truncated wedges, denoted . The sweep edges are blue, the band edges are red, and the greedy edges are gold. For example, su is a sweep step since s is not clean. Next, uv is a band step as $u \in$. Then v is clean, hence vw is a greedy edge.

each edge is chosen according to step 2 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 $u \in$. Next we define the set of band edges, $\mathcal{B}$, as the greedy edges uv of $\mathcal{P}_{s,t'}$, where $u \in$. 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 2.** *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 , , , , then $\|uv\|_2 \leq \|uv\|_1 = h(u) - h(v)$. ◀

▶ **Lemma 3.** *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 $u \in$ and uv crosses ℓ_t^+. This means that $v \in$, as in Figure 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, $u \in$ and $v \in$ together imply $h(u) = x(u) - y(u)$, so we have $\|ut\|_\infty - \|vt\|_\infty = x(u) - y(v) = h(u) + y(u) - y(v) \geq \|uv\|_\infty + \delta + y(u) - y(v) \geq \delta$. ◀

▶ **Lemma 4.** *If $uv \in \mathcal{G}$, then $h(v) - h(u) \leq \|uv\|_\infty$.*

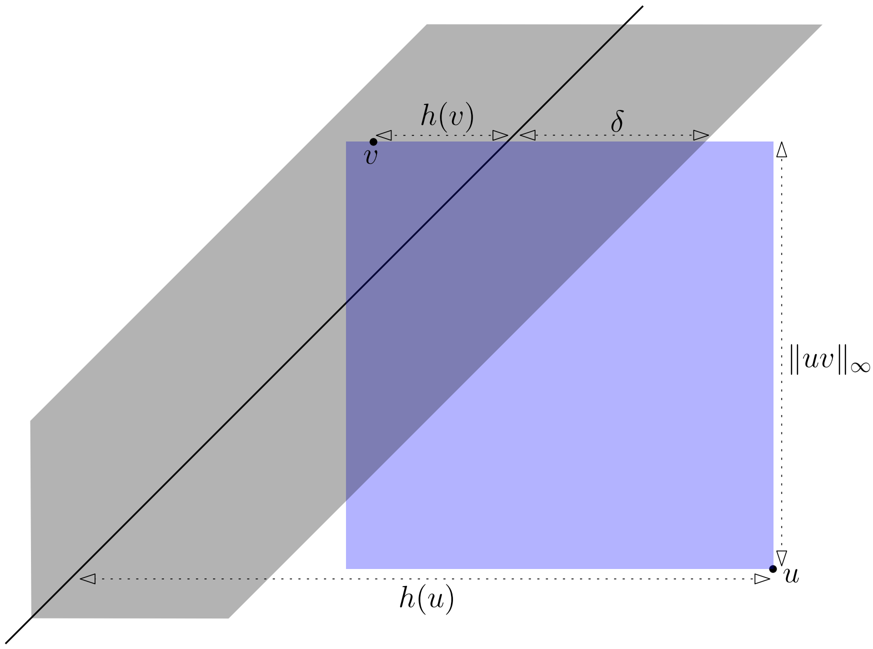

Figure 2 Lemma 3: An example of a sweep edge uv that crosses the diagonal ℓ_t^+ of t.

Proof. Suppose without loss of generality that $u \in$ 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 $h(v) = ||x(v)| - |y(v)|| \leq |x(v) - y(v)|$. ◀

Next we will characterize the behaviour of band edges.

▶ **Lemma 5.** *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 $u \in$. Notice that $\|uv\|_y + \|vt\|_y - \|ut\|_y = (y(u) - y(v)) + |y(v)| - y(u) = |y(v)| - y(v)$, 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 $\psi = x$. ◀

Lemma 5 tells us that when uv is a band edge with $u \in$, then $v \in \cup \cup \cup$. 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 $\cup \cup \cup$. In other words, a sequence of consecutive band edges remains in the same region of 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 sYaoub-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 $u \in$ , $v \in$ , 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 7.** *For all $(u,v) \in \mathcal{B}'$, we have $h(v) - h(u) \le \|uv\|_\infty$.*

Proof. Suppose without loss of generality that $u \in$  and $v \in$ . Then we have $x(v) < x(u) \le x(v) + \|uv\|_\infty$ and $y(v) < y(u) \le y(v) + \|uv\|_\infty$. The rest of the proof is identical to that of Lemma 4. ◀

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

▶ **Definition 8** (Vertices v' and u'). *Let v' be the last vertex of $\mathcal{P}_{s,t'}$ in* . *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 9.** *We have $\sum_{uv \in \mathcal{G}} (\|ut\|_\infty - \|vt\|_\infty) + \sum_{(u,v) \in \mathcal{B}'} (\|ut\|_\infty - \|vt\|_\infty) \le 1$.*

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 1.

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

Dropping the terms corresponding to sweep edges yields the desired inequality. ◀

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

▶ **Lemma 10.** *Let $k := \lfloor \frac{1-2\delta}{\delta} \rfloor + 1$. We have*

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

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^-.

$$\begin{aligned}
\sum_{uv \in \mathcal{S}} \|uv\|_2 &= \sum_{uv \in \mathcal{S} \setminus \mathcal{X}} \|uv\|_2 + \sum_{uv \in \mathcal{X}} \|uv\|_2 \\
&\le \sum_{uv \in \mathcal{S} \setminus \mathcal{X}} (h(u) - h(v)) + \sum_{uv \in \mathcal{X}} \|uv\|_2 && \text{By Lemma 2} \\
&= \sum_{uv \in \mathcal{S}} (h(u) - h(v)) + \sum_{uv \in \mathcal{X}} (\|uv\|_2 - h(u) + h(v))
\end{aligned}$$

Next, by Definition 8, 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

$$\sum_{uv \in \mathcal{S} \cup \mathcal{G}} (h(u) - h(v)) + \sum_{(u,v) \in \mathcal{B}'} (h(u) - h(v)) = h(s) - h(u') \tag{1}$$

Next we rearrange (1) and use Lemmas 4 and 7:

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

Now, let $m := |\mathcal{X}|-1$, and label the edges of $\mathcal{X}$ as $u_0v_0, ..., u_mv_m$ in the order they appear on the path $\mathcal{P}_{s,t'}$. By Lemma 3, we have $\|u_it\|_\infty - \|v_it\|_\infty \geq \delta$ for all $i \in \{0, ..., m\}$. Additionally, u_{i+1} appears after v_i on the path $\mathcal{P}_{s,t'}$, then by Remark 1 we have $\|u_{i+1}t\|_\infty \leq \|v_it\|_\infty$ for $i \in \{0, ..., m-1\}$. Since $\|u_0t\|_\infty \leq \|st\|_\infty = 1$, the above inequalities imply that $\|u_it\|_\infty \leq 1-i\delta$ for $i \in \{0, ..., m\}$. 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 3 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:

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

From Lemma 10, 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 11.** *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 $u \in$ 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 3.

Firstly, label the edges of $\mathcal{G}'$ as $u_1v_1, ..., u_mv_m$. Note that the edges are labelled in the order they appear, and not necessarily consecutive. For $i \in \{1, .., m\}$, 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 $F : \mathbb{R}^2 \to \mathbb{R}^2$ 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 $i \in \{1, ..., m-1\}$. 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 1 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 $p_m \in$, then $y(F(p_m)) \geq 0$. Also, we have $y(F(u_1)) \leq 2-\delta$ given that $u_1 \in$.

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 $\mathcal{G}'$. ◀

At this stage, we could multiply the bound from Lemma 11 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 12.

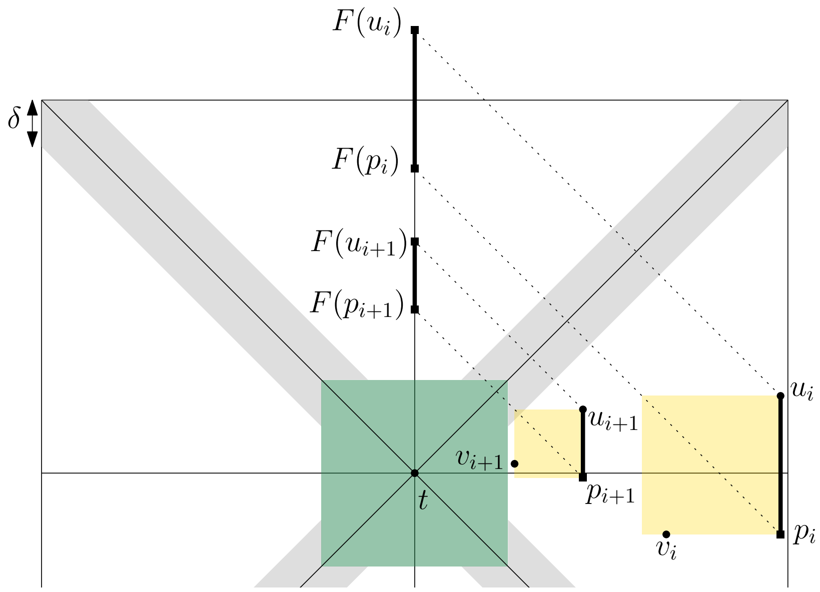

Figure 3 Lemma 11: The vertical components of the edges u_iv_i and $u_{i+1}v_{i+1}$ can be projected onto the y-axis using the projection F. Furthermore, the projections are disjoint.

▶ **Lemma 12.** *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 4. Suppose without loss of generality that $uv \in \mathcal{G}$ and $u \in$  with $y(u) > 0$. Then the edge uv is Type A if $v \in$ , Type B if $v \in$ , and Type C if $v \in$ . 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 $u \in$ , $0 < x(u)$ and $v \in$  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. If $uv \in \mathbf{A} \cup \mathbf{B}$ then $\|uv\|_1 \leq \|uv\|_\infty + \|ut\|_\infty - \|vt\|_\infty$.

Proof. Suppose without loss of generality that $u \in$  and $y(u) > 0$. If $uv \in \mathbf{A}$ then $v \in$ , 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 $v \in$ , 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 $u_1v_1, \ldots, u_mv_m \in \mathbf{C}$ with $u_i \in$  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_iv_i), we have $\|u_{i+1}t\|_\infty \leq \|u_it\|_\infty - \|u_iv_i\|_\infty$ for $i \in \{1, \ldots, m-1\}$ and $\|u_mv_m\|_\infty \leq \|u_mt\|_\infty$. Then we have $\sum_{i=1}^{m} \|u_iv_i\|_\infty \leq \|u_mt\|_\infty + \sum_{i=1}^{m-1} \|u_it\|_\infty - \|u_{i+1}t\|_\infty = \|u_1t\|_\infty \leq 1$. The final step is multiplication by 4 to account for the four connected components of .

Now we combine the above claim with Lemma 11 to complete the proof.

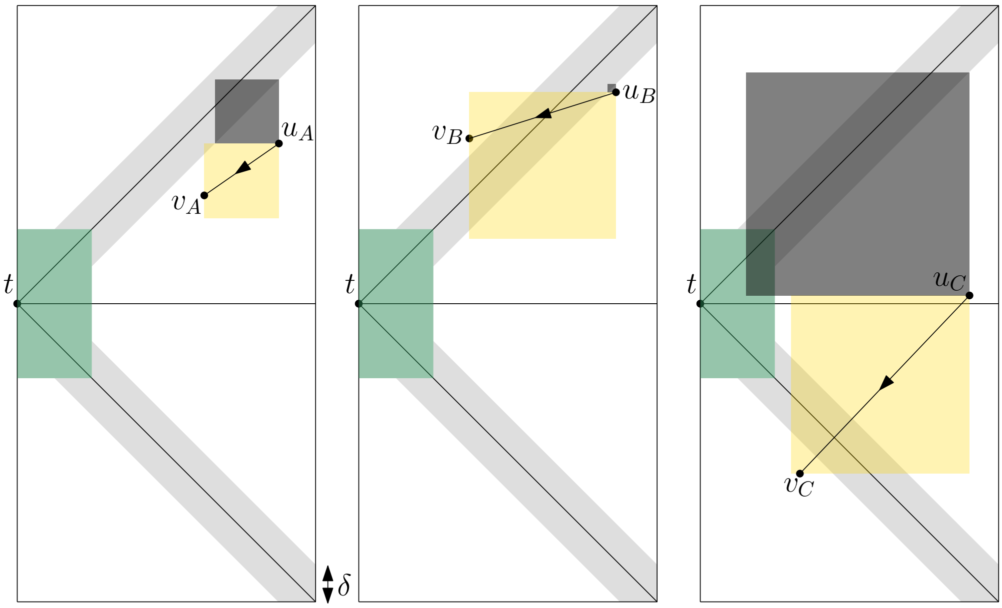

Figure 4 Lemma 12: Edges u_Av_A, u_Bv_B, u_Cv_C are examples of Type A, B and C greedy edges, respectively. Notice that u_A, u_B, u_C are all clean with respect to ℓ_t^+, meaning that the dark grey regions are empty.

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

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

▶ **Definition 13.** *For $2\delta \leq r \leq 1$, define the eastern hexagon* ◖$_r$ *to be the set of points $p \in$* ◀ *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 ◖$_r$, either $\mathcal{P}$ becomes significantly closer to t, or $\mathcal{P}$ never revisits the region ⧗◖.

▶ **Lemma 14.** *Suppose vertex $w \in$* ⧗◖*, and let v be the next vertex of $\mathcal{P}$ outside of* ◖$_{\|wt\|_\infty}$*. Then at least one of the following must be true:*

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

Proof. Throughout the proof, let u denote the vertex before v in $\mathcal{P}$, meaning $u \in$ ◖$_{\|wt\|_\infty}$. Since $u \in$ ◖$_{\|wt\|_\infty} \subset$ ◀, then we must have $v \in$ ▼ $\cup$ ■ $\cup$ ▲. Without loss of generality, assume $v \in$ ▼ $\setminus$ ■, 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.

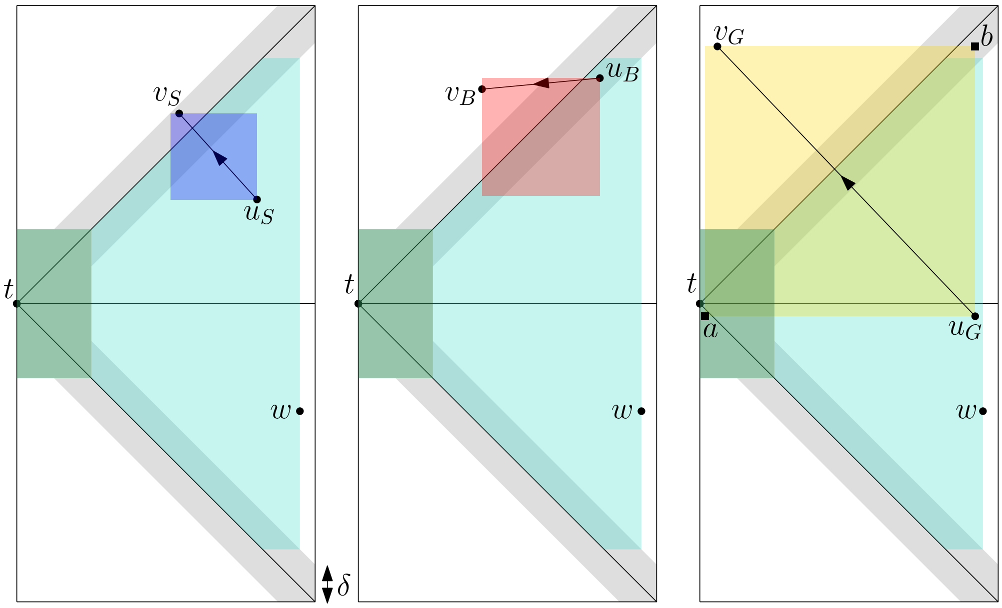

Figure 5 Lemma 14: Edges u_Sv_S, u_Bv_B, u_Gv_G are examples of sweep, band, and greedy edges, respectively. The turquoise region is $_{\|wt\|_\infty}$. The labels a and b refer to the southwest and northeast corners of the gold square corresponding to edge u_Gv_G.

Case: uv is a sweep edge. Then we must have $u \in$ 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 $v \in \setminus$, we must have $u \in$, meaning $y(v) < y(u)$. By definition of $u \in _{\|wt\|_\infty}$, 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 $\{p \in \mid \|pt\|_\infty < \|ut\|_\infty\}$ contains no vertices, implying Item (3). Since $v \in$, 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 $y(a) = y(u) < 0$ 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 $b \in$. This leads to $d_1(a, \ell_t^+) < \delta$, so $a \in$. Then since u is clean with respect to ℓ_t^-, and also $a \in$ and $b \in$ then the region $\{p \in \mid \|pt\|_\infty < \|ut\|_\infty\}$ contains no vertices. Item (3) follows since $\|vt\|_\infty < \|ut\|_\infty$ by Remark 1. ◀

We will also make use of the following remark:

▶ Remark 15. 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 containing u by Lemma 5.

Now we have the tools to prove Lemma 16.

▶ **Lemma 16.** *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 $v \in$. Let $(u_1, v_1), \ldots, (u_m, v_m) \in \mathcal{B}'$ be labelled according to their order on $\mathcal{P}$, where $v_i \in$. Recall that all edges between u_i and v_i are band edges, so by Lemma 5, we have $u_i \in \cup$. Then $\|u_iv_i\|_\infty = \|u_iv_i\|_y$. 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 6. Furthermore, we will show that all projections lie on a sub-segment of length $4 - 5\delta$.

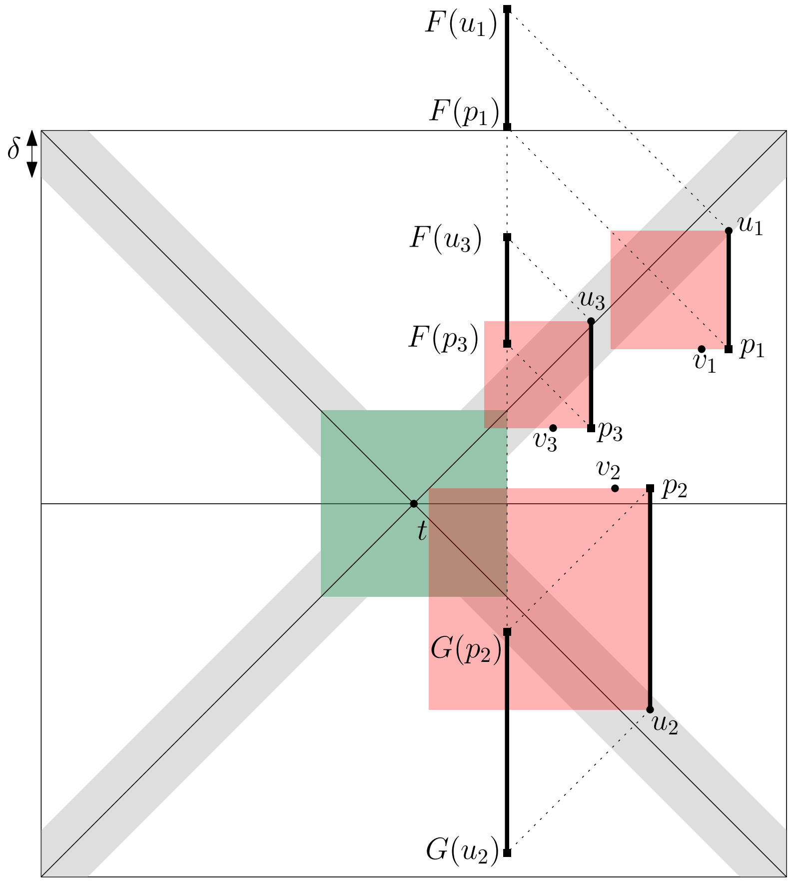

Figure 6 Lemma 16: We project the vertical components of banded sequences onto the dotted line, and prove that the projections are disjoint.

Define the projections $F, G : \mathbb{R}^2 \to \mathbb{R}^2$ 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

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

Let $1 \leq i < j \leq m$. Now we show that that $\mathrm{proj}(i)$ and $\mathrm{proj}(j)$ are disjoint by cases. Assume without loss of generality that $u_i \in$, so by Remark 15, v_i is clean with respect to ℓ_t^+. We will show that $\mathrm{proj}(j)$ is entirely below $\mathrm{proj}(i)$ in both cases.

Case: $u_j \in$. By Lemma 14, we have $y(u_j) < x(v_i) - \delta$. Indeed, $v_i \in$, therefore either $\mathcal{P}_{v_i,u_j} \subset$ $_{\|v_i t\|_\infty}$, or $\|u_j t\|_\infty \leq \|v_i t\|_\infty - \delta$ since $u_j \notin$ and $v_j \in$. 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: $u_j \in$. 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 $v_j \in$ with $y(v_i) = y(p_i) < y(p_j) = y(v_j)$ and $x(v_j) < x(v_i)$.

Now that we have proven that the projections 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 $u_1 \in$. 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 $u_i \in$ 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 $\{2, .., m\}$ such that $u_j \in$. Then all projections are above $y(G(u_j))$, and by Lemma 14 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 proj(i) have y-coordinates between $3\delta - 2$ and $2 - 2\delta$. Summarizing, we have shown

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

Multiplying by 4 for the four connected components of concludes the proof. ◀

Now we can use Lemma 16 to bound the length of all band edges.

▶ **Lemma 17.** *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 8, 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 $u' \in$ and $t' \in \cap$. Then $0 \leq x(t') < x(u')$ and by Lemma 5, $-\delta < y(t')$ 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 by Lemma 5, yielding

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

◀

The bounds we have proved for sweep, greedy and band edges can now be combined to prove Theorem 18.

▶ **Theorem 18.** *DIRTY Algorithm 1 is an* $O(\log(n))$*-memory local routing algorithm with routing ratio of at most* 85.22 *for* $\overrightarrow{\text{Yao}}_4^\infty$ *graphs.*

Proof. We bound the total L_2 distance of sweep edges by combining Lemmas 10, 11 and 16:

$$\sum_{uv\in\mathcal{S}} \|uv\|_2 \le 33 - 28\delta + \sqrt{2}(k - \delta\frac{k(k+1)}{2}) - 2\delta k \tag{2}$$

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

$$\begin{aligned}
\|\mathcal{P}_{s,t'}\|_2 &= \sum_{uv\in\mathcal{S}} \|uv\|_2 + \sum_{uv\in\mathcal{G}} \|uv\|_2 + \sum_{uv\in\mathcal{B}} \|uv\|_2 \\
&\le (33 - 28\delta + \sqrt{2}(k - \delta\frac{k(k+1)}{2}) - 2\delta k) && \text{by (2)} \\
&+ (12 + 4\sqrt{2} - 8\delta + \sum_{uv\in\mathcal{G}} (\|ut\|_\infty - \|vt\|_\infty)) && \text{by Lemma 12} \\
&+ (18 - 19\delta + \sum_{(u,v)\in\mathcal{B}'} (\|ut\|_\infty - \|vt\|_\infty)) && \text{by Lemma 17} \\
&\le 64 - 55\delta + \sqrt{2}(k - \delta\frac{k(k+1)}{2} + 4) - 2\delta k =: f(\delta) && \text{by Lemma 9.}
\end{aligned}$$

Notice that by the definition of step 1 of Algorithm 1, the path $\mathcal{P}_{t',t}$ is identical to the path that Algorithm 1 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 $\frac{f(c)}{1-2c}$ because δ is initially set to $c\|st\|_\infty$ until an edge is chosen in step 1. The ratio evaluates to 85.22 when $c = 0.08$. Furthermore, $\|st\|_2 \ge \|st\|_\infty = 1$ 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 85.22 for $\overrightarrow{\text{Yao}}_4^\infty$ using at most $O(\log(n))$ bits of memory. ◀

4 Lower Bounds

▶ **Lemma 19.** *Let $\epsilon > 0$. There exists a $\overrightarrow{\text{Yao}}_4^\infty$ 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 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 $\overrightarrow{\text{Yao}}_4^\infty$ 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 path from s to t must pass by e. The shortest path from s to e is (s,a,b,c,d,e) since

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

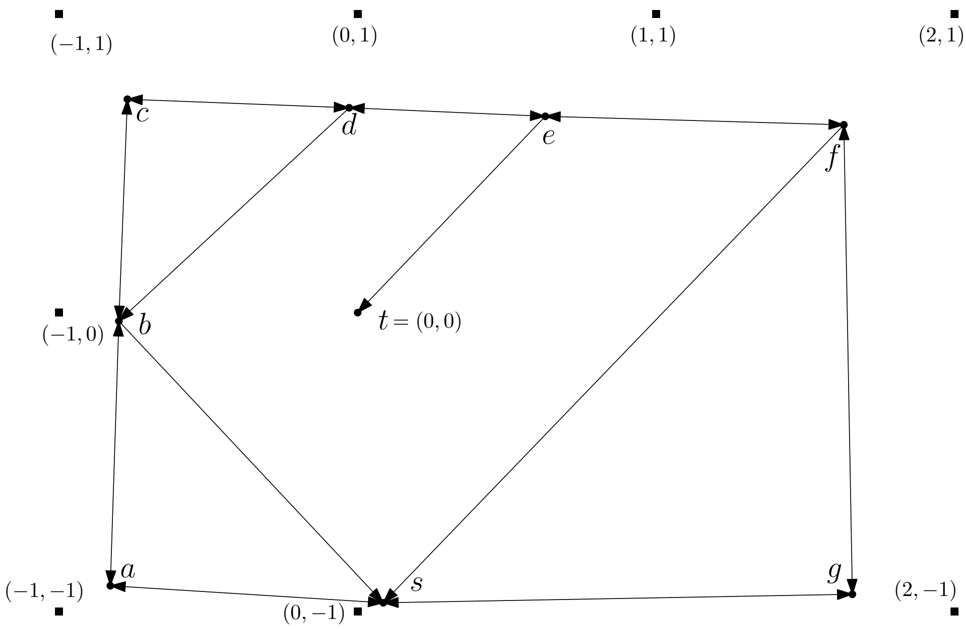

Figure 7 Pictured is the $\vec{\text{Yao}}_4^\infty$ graph with vertices $\{s, a, b, c, d, e, f, g, t\}$. The shortest path from s to t is (s, a, b, c, d, e, t) and its length can be made arbitrarily close to $5 + \sqrt{2}$. The outgoing edges from t are omitted.

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 $5 + \sqrt{2} - \epsilon$. ◀

▶ **Lemma 20.** *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 9. Let $\delta > 0$ be arbitrarily small.

We will start by defining the *zip-zag paths*, illustrated in Figure 8, which will act as subgraphs. 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 WestZag $:= \{p_1, p_2, ..., p_k, q_1, q_2, ..., q_{k-1}, t\}$.

The shortest path from p_1 to t in the $\vec{\text{Yao}}_4^\infty$ graph of WestZag is $p_1, q_1, p_2, q_2, ..., q_{k-1}, p_k, t$, 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

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

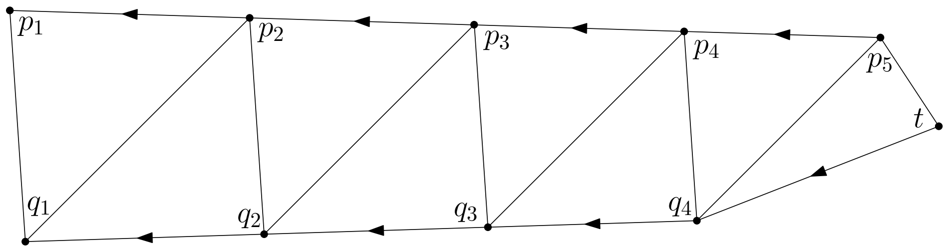

Figure 8 The shortest path from p_1 to t in the $\vec{\text{Yao}}_4^\infty$ graph of WestZag has length arbitrarily close to $1+\sqrt{2}$. Note, all $p_1, p_2, ..., p_k$ lie on segment s_1, and all $q_1, q_2, ..., q_{k-1}$ lie on segment s_2.

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 $V(G_W) :=$ WestZag $\cup \{s,a,b,c',d',e,f,g\}$, $V(G_N) :=$ NorthZag $\cup \{s,a,b,c,d',e,f,g\}$, $V(G_E) :=$ 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 illustrated in Figure 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 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. ◀

▶ **Lemma 21.** *Let $\epsilon > 0$. There exists a $\vec{\text{Yao}}_4$ 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 11.

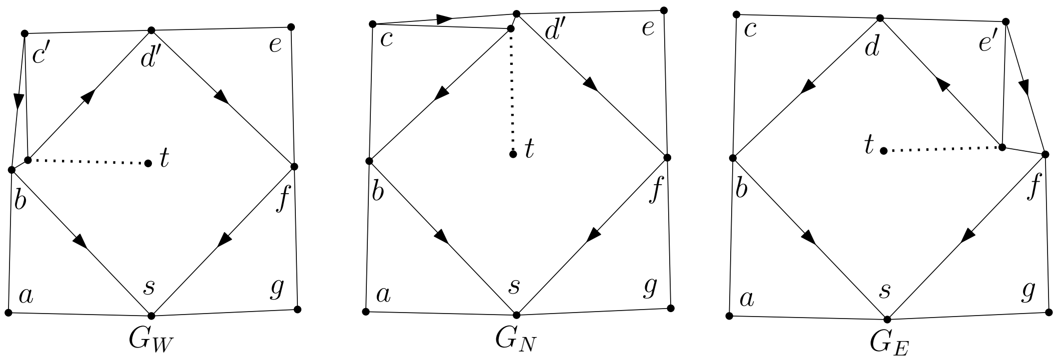

Figure 9 The graphs are G_W, G_N, G_E from left to right. When routing from s to t, the routing ratio of A is at least $7 + \sqrt{2} - \epsilon$ in one of the graphs. The dotted lines represent the zig-zag paths to t. All outgoing edges from the zig-zag paths are omitted.

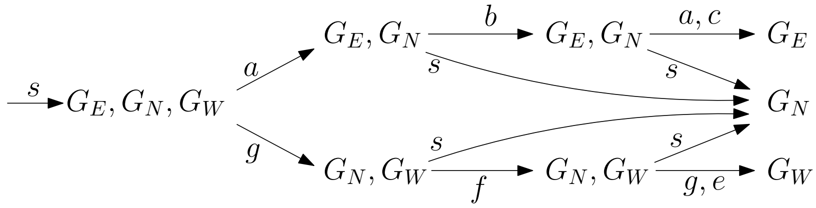

Figure 10 The decision tree describing the adversarial strategy for which graph to provide to A. For example, suppose A first visits the vertices s, a, b. Notice that the neighbourhoods around vertices s, a, b are all identical between graphs G_E, G_N. Then if A chooses c for the next vertex, then A would have a routing ratio of at least $7 + \sqrt{2} - \epsilon$ in G_E.

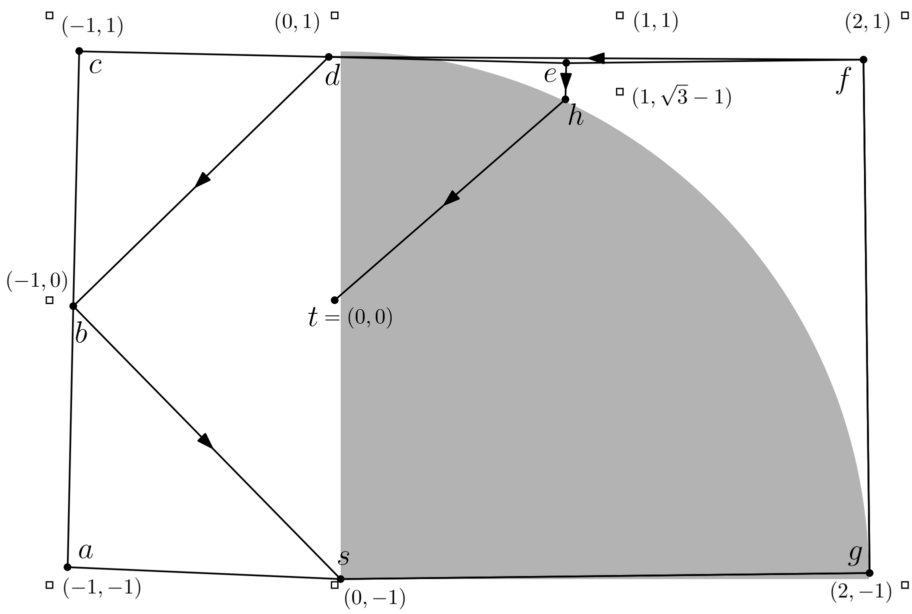

Figure 11 Pictured is the $\vec{\text{Yao}}_4$ graph with vertices $\{s, a, b, c, d, e, f, g, h, t\}$. The shortest path from s to t must pass by h, and its length can be made arbitrarily close to $7 - \sqrt{3} + \sqrt{5 - 2\sqrt{3}} \approx 6.51$. Several edges are omitted for clarity.

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 $\overrightarrow{\text{Yao}}_4$ 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

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

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

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

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

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

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

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

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

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

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

▶ **Lemma 22.** *Let $\epsilon > 0$. Then any local routing algorithm for $\overrightarrow{\text{Yao}}_4$ 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 $\overrightarrow{\text{Yao}}_4$ 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 12. Let $\delta < 0$ be arbitrarily small. Define

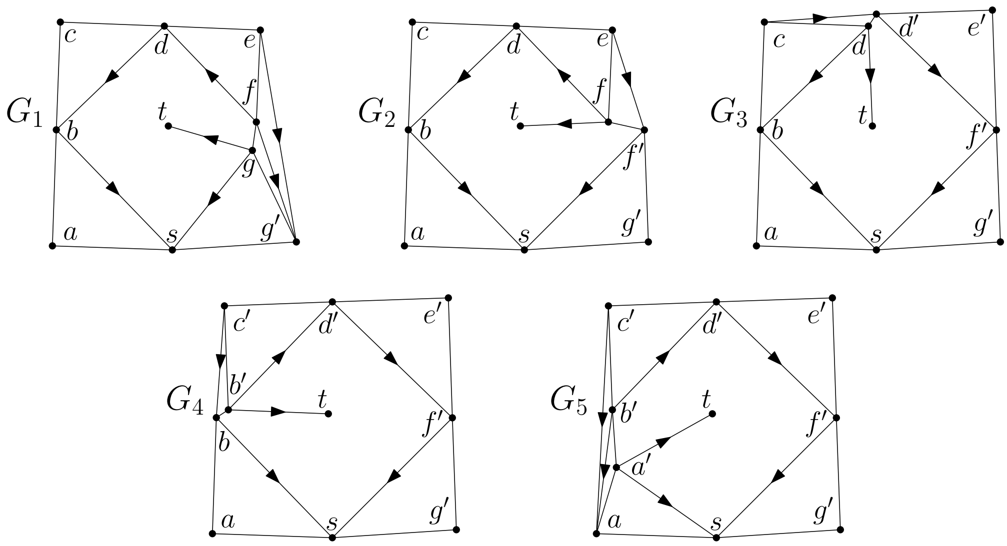

Figure 12 Any local routing algorithm for $\vec{\text{Yao}}_4$ has a routing ratio of at least $7 + \sqrt{2} - \epsilon$ in one of the five graphs. The outgoing edges from t are omitted.

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: $V(G_1) := \{s,t,a,b,c,d,e,f,g,g'\}$, $V(G_2) := \{s,t,a,b,c,d,e,f,f',g'\}$, $V(G_3) := \{s,t,a,b,c,d,d',e',f',g'\}$, $V(G_4) := \{s,t,a,b,b',c',d',e',f',g'\}$, $V(G_5) := \{s,t,a,a',b',c',d',e',f',g'\}$. 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 13.

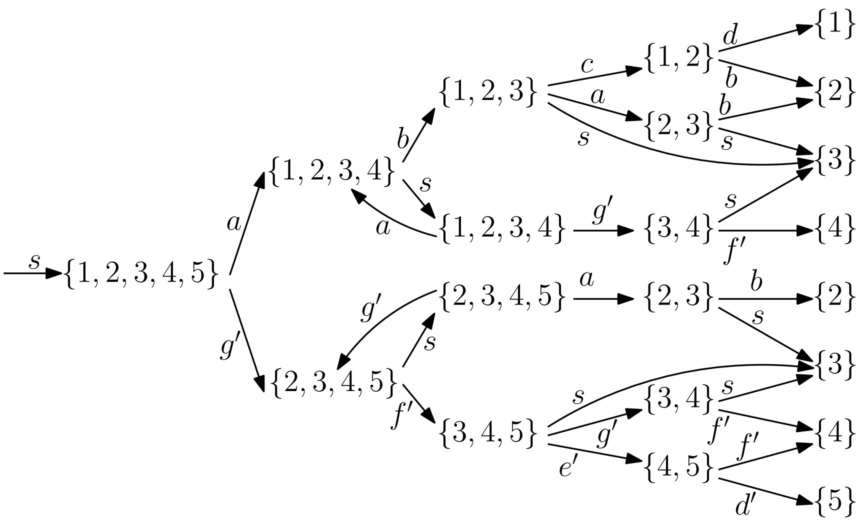

Figure 13 The adversarial decision tree for which graph to provide to any algorithm.

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 $7+\sqrt{2}-\epsilon$. ◀

References

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. doi:10.1016/j.comgeo.2022.101881.

2 Vikrant Ashvinkumar, Joachim Gudmundsson, Christos Levcopoulos, Bengt J. Nilsson, and André van Renssen. Local routing in sparse and lightweight geometric graphs. *Algorithmica*, 84(5):1316–1340, 2022. doi:10.1007/s00453-022-00930-2.

3 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. doi:10.20382/jocg.v6i2a3.

4 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. doi:10.1007/978-3-642-14165-2_3.

5 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. doi:10.1016/j.comgeo.2014.10.005.

6 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. doi:10.1007/s00454-015-9676-z.

7 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. doi:10.1007/s00454-023-00597-8.

8 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. doi:10.1016/j.tcs.2015.12.017.

9 Prosenjit Bose, Mirela Damian, Karim Douïeb, Joseph O'Rourke, Ben Seamone, Michiel H. M. Smid, and Stefanie Wuhrer. $\pi/2$-angle Yao graphs are spanners. *Int. J. Comput. Geom. Appl.*, 22(1):61–82, 2012. doi:10.1142/S0218195912600047.

10 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. doi:10.1137/140988103.

11 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. doi:10.4230/LIPIcs.ISAAC.2024.15.

12 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. doi:10.4230/LIPIcs.ESA.2023.26.

13 Paul Chew. There is a planar graph almost as good as the complete graph. In *SCG*, pages 169–177. ACM, 1986. doi:10.1145/10515.10534.

14 Mirela Damian and Naresh Nelavalli. Improved bounds on the stretch factor of y_4. *Comput. Geom.*, 62:14–24, 2017. doi:10.1016/j.comgeo.2016.12.001.

15 Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational geometry: algorithms and applications, 3rd Edition.* Springer, 2008. doi:10.1007/978-3-540-77974-2.

16 William Michael Zoltan Kalnay. Routing ratio of the directed Yao-6 graphs. Master's thesis, Carleton University, Ottawa, 2023.

17 Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks.* Cambridge University Press, 2007. doi:10.1017/CBO9780511546884.

18 Andrew Chi-Chih Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J. Comput.*, 11(4):721–736, 1982. doi:10.1137/0211059.