The θ_5-graph is a spanner

Prosenjit Bose* Pat Morin* André van Renssen* Sander Verdonschot*

July 10, 2014

Abstract

Given a set of points in the plane, we show that the θ-graph with 5 cones is a geometric spanner with spanning ratio at most $\sqrt{50+22\sqrt{5}} \approx 9.960$. This is the first constant upper bound on the spanning ratio of this graph. The upper bound uses a constructive argument that gives a (possibly self-intersecting) path between any two vertices, of length at most $\sqrt{50+22\sqrt{5}}$ times the Euclidean distance between the vertices. We also give a lower bound on the spanning ratio of $\frac{1}{2}(11\sqrt{5}-17) \approx 3.798$.

1 Introduction

A *t-spanner* of a weighted graph G is a spanning subgraph H with the property that for all pairs of vertices, the weight of the shortest path between the vertices in H is at most t times the weight of the shortest path in G. The *spanning ratio* of H is the smallest t for which it is a t-spanner. We say that a graph is a *spanner* if it has a finite spanning ratio. The graph G is referred to as the *underlying graph*. In this paper, the underlying graph is the complete graph on a finite set of n points in the plane and the weight of an edge is the Euclidean distance between its endpoints. A spanner of such a graph is called a *geometric spanner*. For a comprehensive overview of geometric spanners, we refer the reader to the book by Narasimhan and Smid [1].

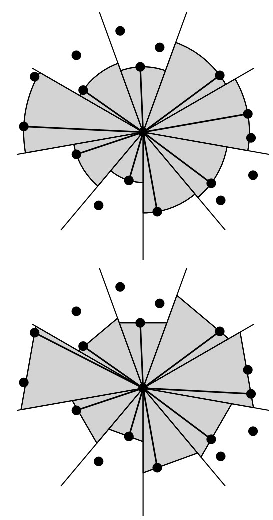


Figure 1: (Top) The construction of the Yao graph. (Bottom) The construction of the θ-graph.

One simple way to build a geometric spanner is to first partition the plane around each vertex into a fixed number of cones and then add an edge between the vertex and the closest vertex in each cone (see Figure 1, top). The resulting graph is called a Yao graph, and is typically denoted by Y_k, where k is the number of cones around each vertex. If the number of cones is sufficiently large, we can find a path between any two vertices by starting at one and walking to the closest vertex in the cone that contains the other, then repeating this until we reach the destination. Intuitively, this results in a short path because we are always walking approximately in the right direction, and, since our neighbour is the

*School of Computer Science, Carleton University. Email: jit@scs.carleton.ca, morin@scs.carleton.ca, andre@cg.scs.carleton.ca, sander@cg.scs.carleton.ca. Research supported in part by NSERC.

closest vertex in that direction, never too far. However, when the number of cones is small, the path found in this way can be very long (see Section 5).

Yao graphs were introduced independently by Flinchbaugh and Jones [2] and Yao [3], before the concept of spanners was even introduced by Chew in 1986 [4]. To the best of our knowledge, the first proof that Yao graphs are geometric spanners was published in 1993, by Althöfer *et al.* [5]. In particular, they showed that for every $t > 1$, there exists a k such that Y_k is a t-spanner. It appears that some form of this result was known earlier, as Clarkson [6] already remarked in 1987 that Y_{12} is a $1+\sqrt{3}$-spanner, albeit without providing a proof or reference. In 2004, Bose *et al.* [7] provided a more specific bound on the spanning ratio, by showing that for $k > 8$, Y_k is a geometric spanner with spanning ratio at most $1/(\cos\theta - \sin\theta)$, where $\theta = 2\pi/k$. This bound was later improved to $1/(1 - 2\sin(\theta/2))$, for $k > 6$ [8].

If we modify the definition of Yao graphs slightly, by connecting not to the closest point in each cone, but to the point whose projection on the bisector of that cone is closest (see Figure 1, bottom), we obtain another type of geometric spanner, called a θ-graph. These graphs were introduced independently by Clarkson [6] and Keil [9, 10], who preferred them to Yao graphs because they are easier to compute. It turns out that θ-graphs share most of the properties of Yao graphs: there is a constant k for every $t > 1$ such that θ_k is a t-spanner, and the spanning ratio for $k > 6$ is $1/(1 - 2\sin(\theta/2))$ [11]. Very recently, the bounds on θ-graphs were even pushed beyond those on Yao graphs [12], including a matching upper- and lower bound of $1 + 2\sin(\theta/2)$ for all θ_k-graphs with $k \geq 6$ and $k \equiv 2 \pmod 4$ [13].

Although most early research focused on Yao and θ-graphs with a large number of cones, using the smallest possible number of cones is important for many practical applications, where the cost of a network is mostly determined by the number of edges. One such example is point-to-point wireless networks. These networks use narrow directional wireless transceivers that can transmit over long distances (up to 50km [14, 15]). The cost of an edge in such a network is therefore equal to the cost of the two transceivers that are used at each endpoint of that edge. In such networks, the cost of building a θ_6-graph is approximately 29% higher than the cost of building a θ_5-graph if the transceivers are randomly distributed [16]. This leads to the natural question: for which values of k are Y_k and θ_k spanners? Kanj [17] presented this question as one of the main open problems in the area of geometric spanners.

Surprisingly, this question was not studied until quite recently. In 2009, El Molla [18] showed that, for $k < 4$, there is no constant t such that Y_k is a t-spanner. These proofs translate to θ-graphs as well. Bonichon *et al.* [19] showed that θ_6 is a 2-spanner, and this is tight. This result was later used by Damian and Raudonis [20] to show that Y_6 is a spanner as well. Bose *et al.* [21] showed that Y_4 is a spanner, and a flurry of recent activity has led to the same result for both θ_4 [22] and Y_5 [23].

In this paper we present the final piece of this puzzle, by giving the first constant upper bound on the spanning ratio of the θ_5-graph, thereby proving that it is a geometric spanner. Since the proof is constructive, it gives us a path between any two vertices, u and w, of length at most $\sqrt{50 + 22\sqrt{5}} \approx 9.960 \cdot |uw|$. Surprisingly, this path can cross itself, a property we observed for the shortest path as well (see Figure 2).

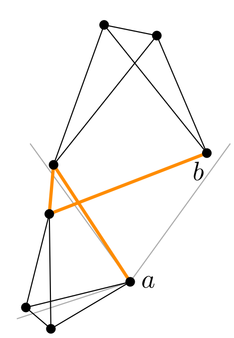


Figure 2: A θ_5-graph where the shortest path between two vertices (in bold) crosses itself.

After completion of this research, we discovered that some form of this result appears to have been known already in 1991, as Ruppert and

Seidel [11] mention that they could prove a bound near 10 on the spanning ratio of θ_5. However, their paper does not include a proof and, to the best of our knowledge, they have not published one since.

In addition to the upper bound on the spanning ratio, we prove two lower bounds. We give an example of a point set where the θ_5-graph has a spanning ratio of $\frac{1}{2}(11\sqrt{5} - 17) \approx 3.798$, and we show that the traditional θ-routing algorithm (follow the edge to the closest vertex in the cone that contains the destination) can result in very long paths, even though a short path exists.

2 Connectivity

In this section we first give a more precise definition of the θ_5-graph, before proving that it is connected.

Given a set P of points in the plane, we consider each point $u \in P$ and partition the plane into 5 cones (regions in the plane between two rays originating from the same point) with apex u, each defined by two rays at consecutive multiples of $\theta = 2\pi/5$ radians from the negative y-axis. We label the cones C_0 through C_4, in clockwise order around u, starting from the top (see Figure 3a). If the apex is not clear from the context, we use C_i^u to denote cone C_i with apex u. For the sake of brevity, we typically write "a cone *of* u" instead of "a cone with apex u".

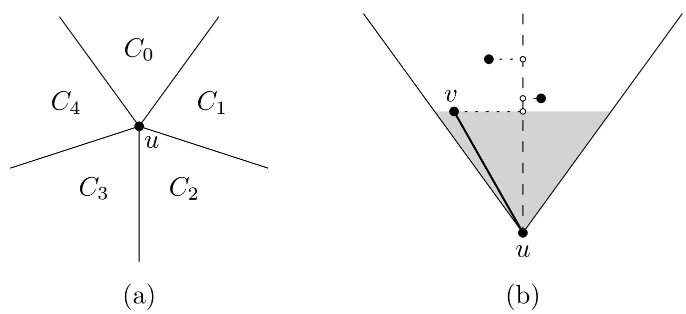


Figure 3: (a) The cones of a vertex u. (b) The vertex v is closest to u. The shaded region is the canonical triangle T_{uv}.

The θ_5-graph is then built by considering each vertex u and connecting it with an edge to the 'closest' vertex in each of its cones, where distance is measured by projecting each vertex onto the bisector of that cone (see Figure 3b). We use this definition of *closest* in the remainder of the paper.

For simplicity, we assume that no two points lie on a line parallel or perpendicular to a cone boundary. This guarantees that each vertex connects to at most one vertex in each cone, and thus that the graph has at most $5n$ edges. For any set of points that does not satisfy this assumption, there exists a tiny angle such that the assumption holds if we rotate all cones by this angle. In terms of the graph, this rotation is equivalent to a tie-breaking rule that always selects the candidate that comes last in clockwise order. Thus, our conclusions about the spanning ratio hold in either case, even though our proofs rely on the general position assumption.

Given two vertices u and v, we define their *canonical triangle* T_{uv} to be the triangle bounded by the cone of u that contains v and the line through v perpendicular to the bisector of that cone. For example, the shaded region in Figure 3b is the canonical triangle T_{uv}. Note that for any pair of vertices u and v, there are two canonical triangles: T_{uv} and T_{vu}. We equate the size $|T_{uv}|$ of a canonical triangle to the length of one of the sides incident to the apex u. This gives us the useful property that any line segment between u and a point inside the triangle has length at most $|T_{uv}|$.

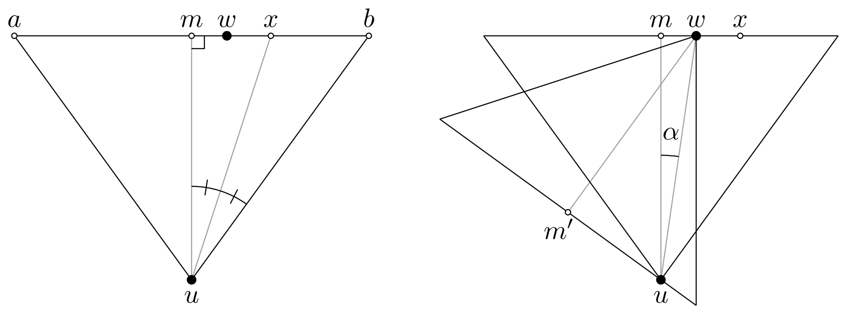
To introduce the structure of the main proof, we first show that the θ_5-graph is connected.

Theorem 1 *The θ_5-graph is connected.*

Proof. We prove that there is a path between any (ordered) pair of vertices in the θ_5-graph, using induction on the size of their canonical triangle. Formally, given two vertices u and w, we perform induction on the rank (relative position) of T_{uw} among the canonical triangles of all pairs of vertices, when ordered by size. For ease of description, we assume that w lies in the right half of C_0^u. The other cases are analogous.

If T_{uw} has rank 1, it is the smallest canonical triangle. Therefore there can be no point closer to u in C_0^u, so the θ_5-graph must contain the edge (u, w). This proves the base case.

If T_{uw} has a larger rank, our inductive hypothesis is that there exists a path between any pair of vertices with a smaller canonical triangle. Let a and b be the left and right corners of T_{uw}. Let m be the midpoint of ab and let x be the intersection of ab and the bisector of $\angle mub$ (see Figure 4a).



(a) (b)

Figure 4: (a) The canonical triangle T_{uw}. (b) If w lies between m and x, T_{wu} is smaller than T_{uw}.

If w lies to the left of x, consider the canonical triangle T_{wu}. Let m' be the midpoint of the side of T_{wu} opposite w and let $\alpha = \angle muw$ (see Figure 4b). Note that $\angle uwm' = \frac{\pi}{5} - \alpha$, since um and the vertical border of T_{wu} are parallel and both are intersected by uw. Using basic trigonometry, we can express the size of T_{wu} as follows.

$$|T_{wu}| = \frac{|wm'|}{\cos\frac{\pi}{5}} = \frac{\cos\angle uwm' \cdot |uw|}{\cos\frac{\pi}{5}} = \frac{\cos\left(\frac{\pi}{5}-\alpha\right)\cdot\frac{|um|}{\cos\alpha}}{\cos\frac{\pi}{5}} = \frac{\cos\left(\frac{\pi}{5}-\alpha\right)}{\cos\alpha}\cdot|T_{uw}|$$

Since w lies to the left of x, the angle α is less than $\pi/10$, which means that $\cos(\frac{\pi}{5}-\alpha)/\cos\alpha$ is less than 1. Hence T_{wu} is smaller than T_{uw} and by induction, there is a path between w and u. Since the θ_5-graph is undirected, we are done in this case. The rest of the proof deals with the case where w lies on or to the right of x.

If T_{wu} is empty, there is an edge between u and w and we are done, so assume that this is not the case. Then there is a vertex v_w that is closest to w in C_3^w (the cone of w that contains u). This gives rise to four cases, depending on the location of v_w (see Figure 5a). In each case, we will show that T_{uv_w} is smaller than T_{uw} and hence we can apply induction to obtain a path between u and

v_w. Since v_w is the closest vertex to w in C_3, there is an edge between v_w and w, completing the path between u and w.

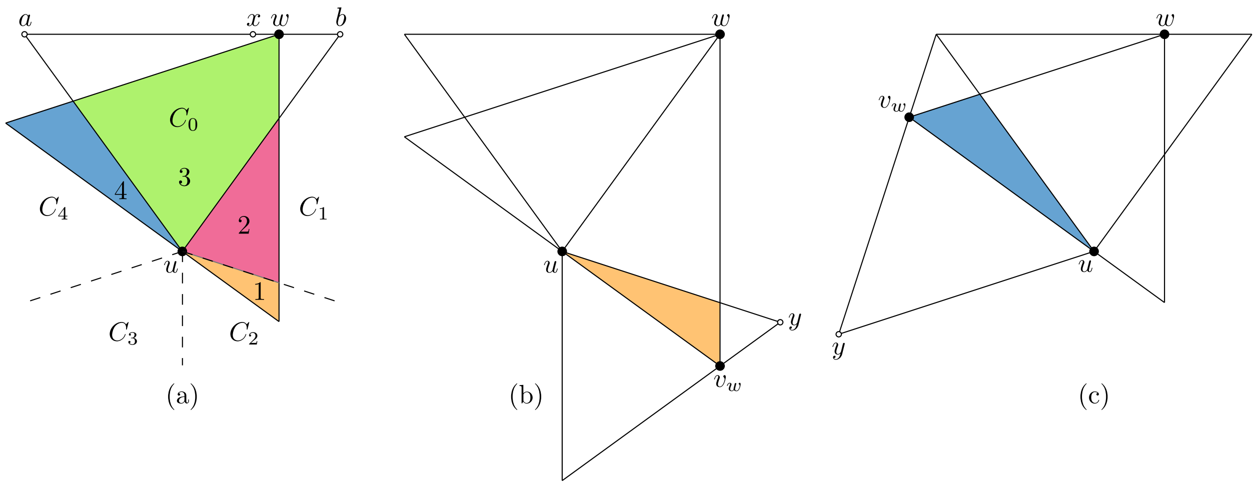


Figure 5: (a) The four cases for v_w. (b) Case 1: The situation that maximizes $|T_{uv_w}|$ when v_w lies in C_2^u. (c) Case 4: The situation that maximizes $|T_{uv_w}|$ when v_w lies in C_4^u.

Case 1. v_w lies in C_2^u. In this case, the size of T_{uv_w} is maximized when v_w lies in the bottom right corner of T_{wu} and w lies on b. Let y be the rightmost corner of T_{uv_w} (see Figure 5b). Using the law of sines, we can express the size of T_{uv_w} as follows.

$$|T_{uv_w}| \quad = \quad |uy| \quad = \quad \frac{\sin \angle uv_w y}{\sin \angle uyv_w} \cdot |uv_w| \quad = \quad \frac{\sin \frac{3\pi}{5}}{\sin \frac{3\pi}{10}} \cdot \tan \frac{\pi}{5} \cdot |T_{uw}| \quad < \quad |T_{uw}|$$

Case 2. v_w lies in C_1^u. In this case, the size of T_{uv_w} is maximized when w lies on b and v_w lies almost on w. By symmetry, this gives $|T_{uv_w}| = |T_{uw}|$. However, v_w cannot lie precisely on w and must therefore lie a little closer to u, giving us that $|T_{uv_w}| < |T_{uw}|$.

Case 3. v_w lies in C_0^u. As in the previous case, the size of T_{uv_w} is maximized when v_w lies almost on w, but since v_w must lie closer to u, we have that $|T_{uv_w}| < |T_{uw}|$.

Case 4. v_w lies in C_4^u. In this case, the size of T_{uv_w} is maximized when v_w lies in the left corner of T_{wu} and w lies on x. Let y be the bottom corner of T_{uv_w} (see Figure 5c). Since x is the point where $|T_{uw}| = |T_{wu}|$, and $v_w yuw$ forms a parallelogram, $|T_{uv_w}| = |T_{uw}|$. However, by general position, v_w cannot lie on the boundary of T_{wu}, so it must lie a little closer to u, giving us that $|T_{uv_w}| < |T_{uw}|$.

Since any vertex in C_3^u would be further from w than u itself, these four cases are exhaustive. □

3 Spanning ratio

In this section, we prove an upper bound on the spanning ratio of the θ_5-graph.

Lemma 2 *Between any pair of vertices u and w of a θ_5-graph, there is a path of length at most $c \cdot |T_{uw}|$, where $c = 2\left(2+\sqrt{5}\right) \approx 8.472$.*

Proof. We begin in a way similar to the proof of Theorem 1. Given an ordered pair of vertices u and w, we perform induction on the size of their canonical triangle. If $|T_{uw}|$ is minimal, there must be a direct edge between them. Since $c > 1$ and any edge inside T_{uw} with endpoint u has length at most $|T_{uw}|$, this proves the base case. The rest of the proof deals with the inductive step, where we assume that there exists a path of length at most $c \cdot |T|$ between every pair of vertices whose canonical triangle T is smaller than T_{uw}. As in the proof of Theorem 1, we assume that w lies in the right half of C_0^u. If w lies to the left of x, we have seen that T_{wu} is smaller than T_{uw}. Therefore we can apply induction to obtain a path of length at most $c \cdot |T_{wu}| < c \cdot |T_{uw}|$ between u and w. Hence we need to concern ourselves only with the case where w lies on or to the right of x.

If u is the vertex closest to w in C_3^w or w is the closest vertex to u in C_0^u, there is a direct edge between them and we are done by the same reasoning as in the base case. Therefore assume that this is not the case and let v_w be the vertex closest to w in C_3^w. We distinguish the same four cases for the location of v_w (see Figure 5a). We already showed that we can apply induction on T_{uv_w} in each case. This is a crucial part of the proof for the first three cases.

Most of the cases come down to finding a path between u and w of length at most $(g+h\cdot c)\cdot|T_{uw}|$, for constants g and h with $h < 1$. The smallest value of c for which this is bounded by $c \cdot |T_{uw}|$ is $g/(1-h)$. If this is at most $2\left(2+\sqrt{5}\right) \approx 8.472$, we are done.

Case 1. v_w lies in C_2^u. By induction, there exists a path between u and v_w of length at most $c \cdot |T_{uv_w}|$. Since v_w is the closest vertex to w in C_3^w, there is a direct edge between them, giving a path between u and w of length at most $|wv_w| + c \cdot |T_{uv_w}|$.

Given any initial position of v_w in C_2^u, we can increase $|wv_w|$ by moving w to the right. Since this does not change $|T_{uv_w}|$, the worst case occurs when w lies on b. Then we can increase both $|wv_w|$ and $|T_{uv_w}|$ by moving v_w into the bottom corner of T_{wu}. This gives rise to the same worst-case configuration as in the proof of Theorem 1, depicted in Figure 5b. Building on the analysis there, we can bound the worst-case length of the path as follows.

$$|wv_w| + c \cdot |T_{uv_w}| \quad = \quad \frac{|T_{uw}|}{\cos\frac{\pi}{5}} + c \cdot \frac{\sin\frac{3\pi}{5}}{\sin\frac{3\pi}{10}} \cdot \tan\frac{\pi}{5} \cdot |T_{uw}|$$

This is at most $c \cdot |T_{uw}|$ for $c \geq 2\left(2+\sqrt{5}\right)$. Since we picked $c = 2\left(2+\sqrt{5}\right)$, the theorem holds in this case. Note that this is one of the cases that determines the value of c.

Case 2. v_w lies in C_1^u. By the same reasoning as in the previous case, we have a path of length at most $|wv_w| + c \cdot |T_{uv_w}|$ between u and w and we need to bound this length by $c \cdot |T_{uw}|$.

Given any initial position of v_w in C_1^u, we can increase $|wv_w|$ by moving w to the right. Since this does not change $|T_{uv_w}|$, the worst case occurs when w lies on b. We can further increase $|wv_w|$ by moving v_w down along the side of T_{uv_w} opposite u until it hits the boundary of C_1^u or C_3^w, whichever comes first (see Figure 6a).

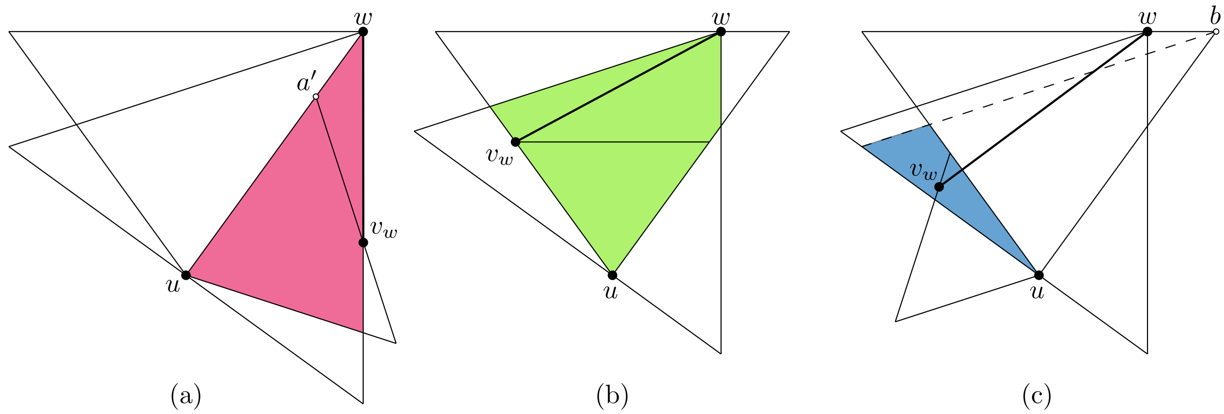


Figure 6: (a) Case 2: Vertex v_w lies on the boundary of C_3^w after moving it down along the side of T_{uv_w}. (b) Case 3: Vertex v_w lies on the boundary of C_0^u after moving it left along the side of T_{uv_w}. (c) Case 4: Vertex v_w lies in $C_4^u \cap C_3^b$.

Now consider what happens when we move v_w along these boundaries. If v_w lies on the boundary of C_1^u and we move it away from u by Δ, $|T_{uv_w}|$ increases by Δ. At the same time, $|wv_w|$ might decrease, but not by more than Δ. Since $c > 1$, the total path length is maximized by moving v_w as far from u as possible, until it hits the boundary of C_3^w. Once v_w lies on the boundary of C_3^w, we can express the size of T_{uv_w} as follows, where a' is the top corner of T_{uv_w}.

$$|T_{uv_w}| \quad = \quad |T_{uw}| - |wa'| \quad = \quad |T_{uw}| - |wv_w| \cdot \frac{\sin \angle wv_w a'}{\sin \angle wa'v_w} \quad = \quad |T_{uw}| - |wv_w| \cdot \frac{\sin \frac{\pi}{10}}{\sin \frac{7\pi}{10}}$$

Now we can express the length of the complete path as follows.

$$|wv_w| + c \cdot |T_{uv_w}| \quad = \quad |wv_w| + c \cdot \left(|T_{uw}| - |wv_w| \cdot \frac{\sin \frac{\pi}{10}}{\sin \frac{7\pi}{10}} \right) \quad = \quad c \cdot |T_{uw}| - \left(c \cdot \frac{\sin \frac{\pi}{10}}{\sin \frac{7\pi}{10}} - 1 \right) \cdot |wv_w|$$

Since $c > \sin \frac{7\pi}{10} / \sin \frac{\pi}{10} \approx 2.618$, we have that $c \cdot (\sin \frac{\pi}{10} / \sin \frac{7\pi}{10}) - 1 > 0$. Therefore $|wv_w| + c \cdot |T_{uv_w}| < c \cdot |T_{uw}|$.

Case 3. v_w lies in C_0^u. Again, we have a path of length at most $|wv_w| + c \cdot |T_{uv_w}|$ between u and w and we need to bound this length by $c \cdot |T_{uw}|$.

Given any initial position of v_w in C_0^u, moving v_w to the left increases $|wv_w|$ while leaving $|T_{uv_w}|$ unchanged. Therefore the path length is maximized when v_w lies on the boundary of either C_0^u or C_3^w, whichever it hits first (see Figure 6b).

Again, consider what happens when we move v_w along these boundaries. Similar to the previous case, if v_w lies on the boundary of C_0^u and we move it away from u by Δ, $|T_{uv_w}|$ increases by Δ, while $|wv_w|$ might decrease by at most Δ. Since $c > 1$, the total path length is maximized by moving v_w as far from u as possible, until it hits the boundary of C_3^w. Once there, the situation is symmetric to the previous case, with $|T_{uv_w}| = |T_{uw}| - |wv_w| \cdot (\sin \frac{\pi}{10} / \sin \frac{7\pi}{10})$. Therefore the theorem holds in this case as well.

Case 4. v_w lies in C_4^u. This is the hardest case. Similar to the previous two cases, the size of T_{uv_w} can be arbitrarily close to that of T_{uw}, but in this case $|wv_w|$ does not approach 0. This means that simply invoking the inductive hypothesis on T_{uv_w} does not work, so another strategy is required. We first look at a subcase where we *can* apply induction directly, before considering the position of v_u, the closest vertex to u in C_0.

Case 4a. v_w lies in $C_4^u \cap C_3^b$. This situation is illustrated in Figure 6c. Given any initial position of v_w, moving w to the right onto b increases the total path length by increasing $|wv_w|$ while not affecting $|T_{uv_w}|$. Here we use the fact that v_w already lies in C_3^b, otherwise we would not be able to move w onto b while keeping v_w in C_3^w. Now the total path length is maximized by placing v_w on the left corner of T_{wu}. Since this situation is symmetrical to the worst-case situation in Case 1, the theorem holds by the same analysis.

Next, we distinguish four cases for the position of v_u (the closest vertex to u in C_0), illustrated in Figure 7a. The cases are: (4b) w lies in $C_4^{v_u}$, (4c) w lies in $C_0^{v_u}$, (4d) w lies in $C_1^{v_u}$ and v_u lies in C_3^w, and (4e) w lies in $C_1^{v_u}$ and v_u lies in C_4^w. These are exhaustive, since the cones C_4, C_0 and C_1 are the only ones that can contain a vertex above the current vertex, and w must lie above v_u, as v_u is closer to u. Further, if w lies in $C_1^{v_u}$, v_u must lie in one of the two opposite cones of w.

We can solve the first two cases by applying our inductive hypothesis to T_{v_uw}.

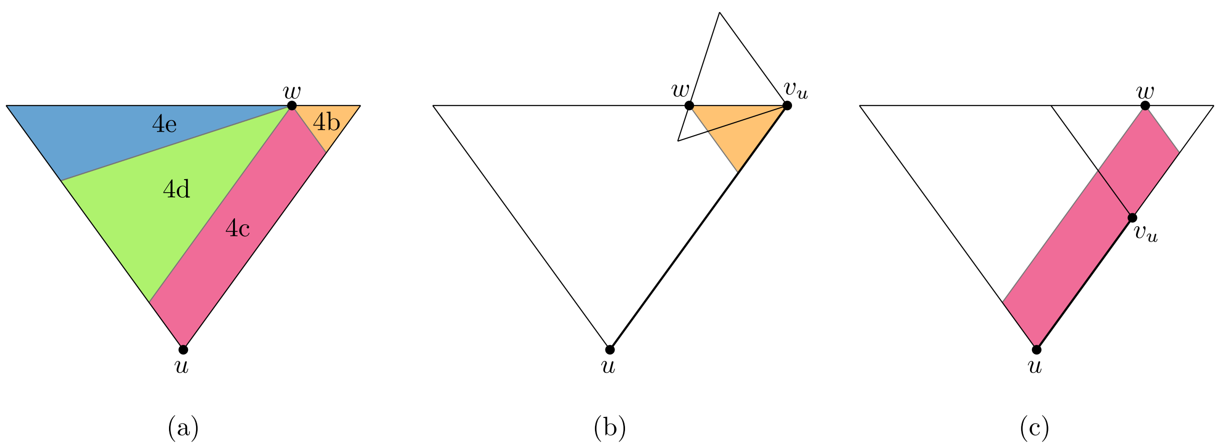


Figure 7: (a) The four different cases for the position of v_u. (b) The worst-case configuration with w in $C_4^{v_u}$. (c) A configuration with w in $C_0^{v_u}$, after moving v_u onto the right side of C_0^u.

Case 4b. w lies in $C_4^{v_u}$. To apply our inductive hypothesis, we need to show that $|T_{v_uw}| < |T_{uw}|$. If that is the case, we obtain a path between v_u and w of length at most $c \cdot |T_{v_uw}|$. Since v_u is the closest vertex to u, there is a direct edge from u to v_u, resulting in a path between u and w of length at most $|uv_u| + c \cdot |T_{v_uw}|$.

Given any intial positions for v_u and w, moving w to the left increases $|T_{v_uw}|$ while leaving $|uv_u|$ unchanged. Moving v_u closer to b increases both. Therefore the path length is maximal when w lies on x and v_u lies on b (see Figure 7b). Using the law of sines, we can express $|T_{v_uw}|$ as follows.

$$|T_{v_uw}| = \frac{\sin\frac{3\pi}{5}}{\sin\frac{3\pi}{10}} \cdot |wv_u| = \frac{\sin\frac{3\pi}{5}}{\sin\frac{3\pi}{10}} \cdot \frac{\sin\frac{\pi}{10}}{\sin\frac{3\pi}{5}} \cdot |T_{uw}| = \frac{\sin\frac{\pi}{10}}{\sin\frac{3\pi}{10}} \cdot |T_{uw}| = \frac{1}{2}\left(3-\sqrt{5}\right) \cdot |T_{uw}|$$

Since $\frac{1}{2}\left(3-\sqrt{5}\right) < 1$, we have that $|T_{v_u w}| < |T_{uw}|$ and we can apply our inductive hypothesis to $T_{v_u w}$. Since $|uv_u| = |T_{uw}|$, the complete path has length at most $c \cdot |T_{uw}|$ for

$$c \quad \geq \quad \frac{1}{1-\frac{1}{2}\left(3-\sqrt{5}\right)} \quad = \quad \frac{1}{2}\left(1+\sqrt{5}\right) \quad \approx \quad 1.618.$$

Case 4c. w lies in $C_0^{v_u}$. Since v_u lies in C_0^u, it is clear that $|T_{v_u w}| < |T_{uw}|$, which allows us to apply our inductive hypothesis. This gives us a path between u and w of length at most $|uv_u| + c \cdot |T_{v_u w}|$. For any initial location of v_u, we can increase the total path length by moving v_u to the right until it hits the side of C_0^u (see Figure 7c), since $|T_{v_u w}|$ stays the same and $|uv_u|$ increases. Once there, we have that $|uv_u| + |T_{v_u w}| = |T_{uw}|$. Since $c > 1$, this immediately implies that $|uv_u| + c \cdot |T_{v_u w}| \leq c \cdot |T_{uw}|$, proving the theorem for this case.

To solve the last two cases, we need to consider the positions of both v_u and v_w. Recall that for v_w, there is only a small region left where we have not yet proved the existance of a short path between u and w. In particular, this is the case when v_w lies in cone C_4^u, but not in C_3^b.

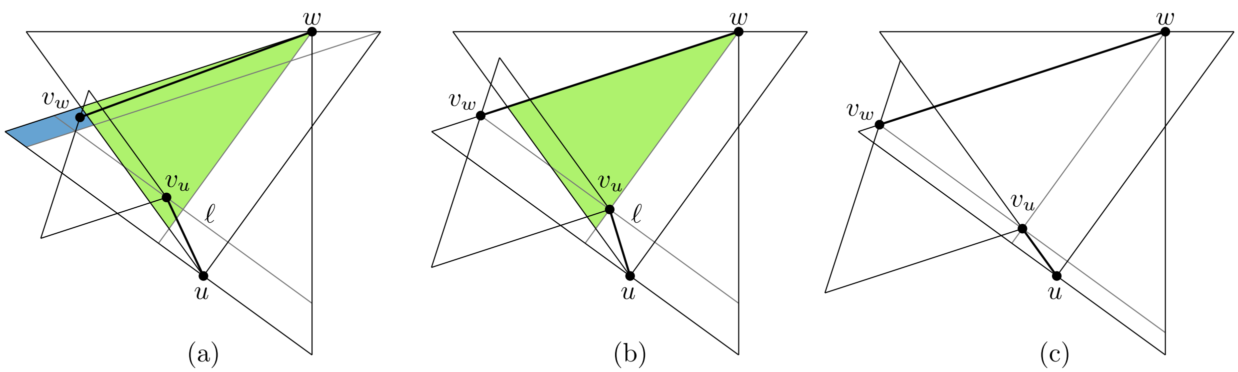


Figure 8: (a) The regions where v_u (light) and v_w (dark) can lie. (b) The worst case when v_u lies on a given line ℓ. (c) The worst case for a fixed position of w.

Case 4d. w lies in $C_1^{v_u}$ and v_u lies in C_3^w. We would like to apply our inductive hypothesis to $T_{v_u v_w}$, resulting in a path between v_u and v_w of length at most $c \cdot |T_{v_u v_w}|$. The edges (w, v_w) and (u, v_u) complete this to a path between u and w, giving a total length of at most $|uv_u| + c \cdot |T_{v_u v_w}| + |v_w w|$.

First, note that v_u cannot lie in $T_{w v_w}$, as this region is empty by definition. Since v_w lies in C_4^u, this means that v_w must lie in $C_4^{v_u}$. We first show that $T_{v_u v_w}$ is always smaller than T_{uw}, which means that we are allowed to use induction. Given any initial position for v_u, consider the line ℓ through v_u, perpendicular to the bisector of C_3 (see Figure 8a). Since v_w cannot be further from w than v_u, the size of $T_{v_u v_w}$ is maximized when v_w lies on the intersection of ℓ and the top boundary of T_{wu}. We can increase $|T_{v_u v_w}|$ further by moving v_u along ℓ until it reaches the bisector of C_3^w (see Figure 8b). Since the top boundary of T_{wu} and the bisector of C_3^w approach each other as they get closer to w, the size of $T_{v_u v_w}$ is maximized when v_u lies on the bottom boundary of T_{wu} (ignoring for now that this would move v_u out of T_{uw}). Now it is clear that $|T_{v_u v_w}| < |T_{u v_w}|$. Since we already established that $T_{u v_w}$ is smaller than T_{uw} in the proof of Theorem 1, this holds for $T_{v_u v_w}$ as well and we can use induction.

All that is left is to bound the total length of the path. Given any initial position of v_u, the path length is maximized when we place v_w at the intersection of ℓ and the top boundary of T_{wu}, as this maximizes both $|T_{v_u v_w}|$ and $|wv_w|$. When we move v_u away from v_w along ℓ by Δ, $|uv_u|$ decreases by at most Δ, while $|T_{v_u v_w}|$ increases by $\sin \frac{3\pi}{5} / \sin \frac{3\pi}{10} \cdot \Delta > \Delta$. Since $c > 1$, this increases the total path length. Therefore the worst case again occurs when v_u lies on the bisector of C_3^w, as depicted in Figure 8b. Moving v_u down along the bisector of T_{wu} by Δ decreases $|uv_u|$ by at most Δ, while increasing $|wv_w|$ by $1/\sin \frac{3\pi}{10} \cdot \Delta > \Delta$ and increasing $|T_{v_u v_w}|$. Therefore this increases the total path length and the worst case occurs when v_u lies on the left boundary of T_{uw} (see Figure 8c).

Finally, consider what happens when we move v_u Δ towards u, while moving w and v_w such that the construction stays intact. This causes w to move to the right. Since v_u, w and the left corner of T_{uw} form an isosceles triangle with apex v_u, this also moves v_u Δ further from w. We saw before that moving v_u away from w increases the size of $T_{v_u v_w}$. Finally, it also increases $|wv_w|$ by $1/\sin \frac{3\pi}{10} \cdot \Delta > \Delta$. Thus, the increase in $|wv_w|$ cancels the decrease in $|uv_u|$ and the total path length increases. Therefore the worst case occurs when v_u lies almost on u and v_w lies in the corner of T_{wu}, which is symmetric to the worst case of Case 1. Thus the theorem holds by the same analysis.

Case 4e. w lies in $C_1^{v_u}$ and v_u lies in C_4^w. We split this case into three final subcases, based on the position of v_u. These cases are illustrated in Figure 9a. Note that v_u cannot lie in C_2 or C_3 of v_w, as it lies above v_w. It also cannot lie in $C_4^{v_w}$, as $C_4^{v_w}$ is completely contained in C_4^u, whereas v_u lies in C_0^u. Thus the cases presented below are exhaustive.

Case 4e-1. $|T_{wv_u}| \leq \frac{c-1}{c} \cdot |T_{uw}|$. If T_{wv_u} is small enough, we can apply our inductive hypothesis to obtain a path between v_u and w of length at most $c \cdot |T_{wv_u}|$. Since there is a direct edge between u and v_u, we obtain a path between u and w of length at most $|uv_u| + c \cdot |T_{wv_u}|$. Any edge from u to a point inside T_{uw} has length at most $|T_{uw}|$, so we can bound the length of the path as follows.

$$|uv_u| + c \cdot |T_{wv_u}| \quad \leq \quad |T_{uw}| + c \cdot \frac{c-1}{c} \cdot |T_{uw}| \quad = \quad |T_{uw}| + (c-1) \cdot |T_{uw}| \quad = \quad c \cdot |T_{uw}|$$

In the other two cases, we use induction on $T_{v_w v_u}$ to obtain a path between v_w and v_u of length at most $c \cdot |T_{v_w v_u}|$. The edges (u, v_u) and (w, v_w) complete this to a (self-intersecting) path between u and w. We can bound the length of these edges by the size of the canonical triangle that contains them, as follows.

$$|uv_u| + |wv_w| \quad \leq \quad |T_{uw}| + |T_{wu}| \quad \leq \quad |T_{uw}| + \frac{1}{\cos \frac{\pi}{5}} \cdot |T_{uw}| \quad = \quad \sqrt{5} \cdot |T_{uw}|$$

All that is left now is to bound the size of $T_{v_w v_u}$ and express it in terms of T_{uw}.

Case 4e-2. v_u lies in $C_0^{v_w}$. In this case, the size of $T_{v_w v_u}$ is maximal when v_u lies on the top boundary of T_{uw} and v_w lies at the lowest point in its possible region: the left corner of T_{bu} (see Figure 9b). Now we can express $|T_{v_w v_u}|$ as follows.

$$|T_{v_w v_u}| \quad = \quad \frac{\sin \frac{\pi}{10}}{\sin \frac{7\pi}{10}} \cdot |bv_w| \quad = \quad \frac{\sin \frac{\pi}{10}}{\sin \frac{7\pi}{10}} \cdot \frac{1}{\cos \frac{\pi}{5}} \cdot |T_{uw}| \quad = \quad 2\left(\sqrt{5} - 2\right) \cdot |T_{uw}|$$

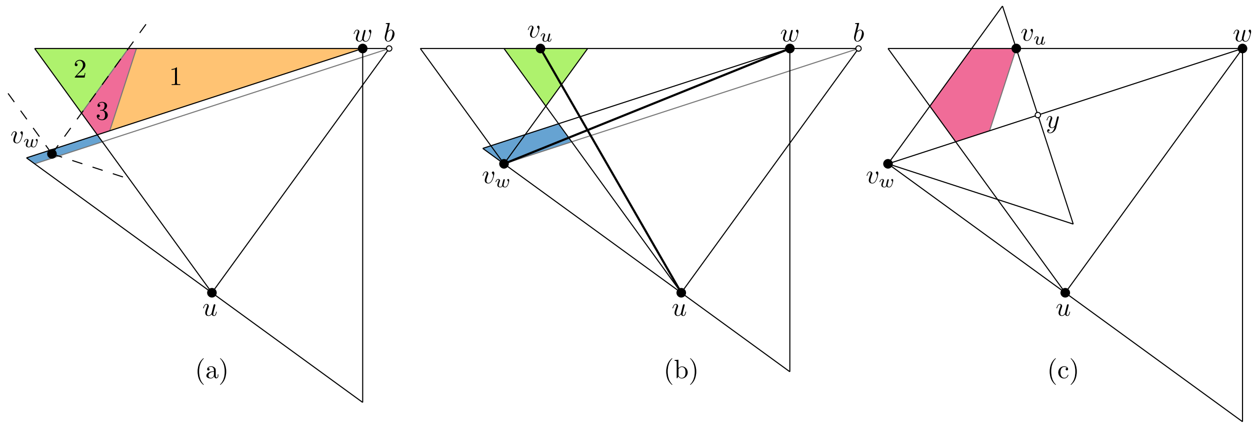


Figure 9: (a) The three subcases for the position of v_u. (b) The situation that maximizes $T_{v_w v_u}$ when v_u lies in $C_0^{v_w}$. (c) The worst case when v_u lies in $C_1^{v_w}$.

Since $2\left(\sqrt{5}-2\right) < 1$, we can use induction. The total path length is bounded by $c \cdot |T_{uw}|$ for

$$c \quad \geq \quad \frac{\sqrt{5}}{1-2\left(\sqrt{5}-2\right)} \quad = \quad 2+\sqrt{5} \quad \approx \quad 4.236.$$

Case 4e-3. v_u lies in $C_1^{v_w}$. Since $|T_{wv_u}| > \frac{c-1}{c} \cdot |T_{uw}|$, $T_{v_w v_u}$ is maximal when v_w lies on the left corner of T_{wu} and v_u lies on the top boundary of T_{uw}, such that $|T_{wv_u}| = \frac{c-1}{c} \cdot |T_{uw}|$ (see Figure 9c). Let y be the intersection of $T_{v_w v_u}$ and T_{wu}. Note that since v_w lies on the corner of T_{wu}, y is also the midpoint of the side of $T_{v_w v_u}$ opposite v_w. We can express the size of $T_{v_w v_u}$ as follows.

$$|T_{v_w v_u}| \quad = \quad \frac{|v_w y|}{\cos \frac{\pi}{5}} \quad = \quad \frac{|wv_w| - |wy|}{\cos \frac{\pi}{5}} \quad = \quad \frac{\frac{|T_{uw}|}{\cos \frac{\pi}{5}} - \cos \frac{\pi}{10} \cdot |wv_u|}{\cos \frac{\pi}{5}}$$

$$= \quad \frac{\frac{|T_{uw}|}{\cos \frac{\pi}{5}} - \cos \frac{\pi}{10} \cdot \frac{\sin \frac{3\pi}{10}}{\sin \frac{3\pi}{5}} \cdot |T_{wv_u}|}{\cos \frac{\pi}{5}} \quad = \quad \frac{\frac{|T_{uw}|}{\cos \frac{\pi}{5}} - \cos \frac{\pi}{10} \cdot \frac{\sin \frac{3\pi}{10}}{\sin \frac{3\pi}{5}} \cdot \frac{c-1}{c} \cdot |T_{uw}|}{\cos \frac{\pi}{5}}$$

$$= \quad \left(\frac{1}{c} + 5 - 2\sqrt{5}\right) \cdot |T_{uw}|$$

Thus we can use induction for $c > 1/\left(2\sqrt{5}-4\right) \approx 2.118$ and the total path length can be bounded by $c \cdot |T_{uw}|$ for

$$c \quad \geq \quad \frac{\sqrt{5}+1}{2\sqrt{5}-4} \quad = \quad \frac{1}{2}\left(7+3\sqrt{5}\right) \quad \approx \quad 6.854.$$

$\square$

Using this result, we can compute the exact spanning ratio.

Theorem 3 *The θ_5-graph is a spanner with spanning ratio at most* $\sqrt{50+22\sqrt{5}} \approx 9.960$.

Proof. Given two vertices u and w, we know from Lemma 2 that there is a path between them of length at most $c \cdot \min(|T_{uw}|, |T_{wu}|)$, where $c = 2\left(2+\sqrt{5}\right) \approx 8.472$. This gives an upper bound on the spanning ratio of $c \cdot \min(|T_{uw}|, |T_{wu}|)/|uw|$. We assume without loss of generality that w lies in the right half of C_0^u. Let α be the angle between the bisector of C_0^u and the line uw (see Figure 4b). In the proof of Theorem 1, we saw that we can express $|T_{wu}|$ and $|uw|$ in terms of α and $|T_{uw}|$, as $|T_{wu}| = (\cos(\frac{\pi}{5}-\alpha)/\cos\alpha)\cdot|T_{uw}|$ and $|uw| = (\cos\frac{\pi}{5}/\cos\alpha)\cdot|T_{uw}|$, respectively. Using these expressions, we can write the spanning ratio in terms of α.

$$\frac{c \cdot \min(|T_{uw}|, |T_{wu}|)}{|uw|} = \frac{c \cdot \min\left(|T_{uw}|, \frac{\cos\left(\frac{\pi}{5}-\alpha\right)}{\cos\alpha}\cdot|T_{uw}|\right)}{\frac{\cos\frac{\pi}{5}}{\cos\alpha}\cdot|T_{uw}|} = \frac{c}{\cos\frac{\pi}{5}}\cdot\min\left(\cos\alpha, \cos\left(\frac{\pi}{5}-\alpha\right)\right)$$

To get an upper bound on the spanning ratio, we need to maximize the minimum of $\cos\alpha$ and $\cos\left(\frac{\pi}{5}-\alpha\right)$. Since for $\alpha \in [0, \pi/5]$, one is increasing and the other is decreasing, this maximum occurs at $\alpha = \pi/10$, where they are equal. Thus, our upper bound becomes

$$\frac{c}{\cos\frac{\pi}{5}}\cdot\cos\frac{\pi}{10} = \sqrt{50+22\sqrt{5}} \approx 9.960.$$

□

4 Lower bound

In this section, we derive a lower bound on the spanning ratio of the θ_5-graph.

Theorem 4 *The θ_5-graph has spanning ratio at least $\frac{1}{2}(11\sqrt{5}-17) \approx 3.798$.*

Proof. For the lower bound, we present and analyze a path between two vertices that has a large spanning ratio. The path has the following structure (illustrated in Figure 10).

The path can be thought of as being directed from w to u. First, we place w in the right corner of T_{uw}. Then we add a vertex v_1 in the bottom corner of T_{wu}. We repeat this two more times, each time adding a new vertex in the corner of T_{v_iu} furthest away from u. The final vertex v_4 is placed on the top boundary of $C_1^{v_3}$, such that u lies in $C_1^{v_4}$. Since we know all the angles involved, we can compute the length of each edge, taking $|uw| = 1$ as baseline.

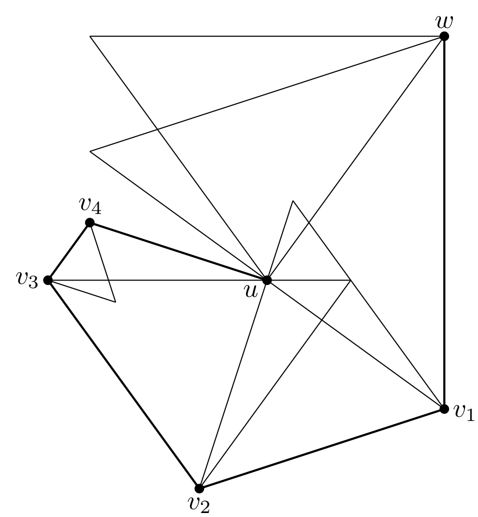


Figure 10: A path with a large spanning ratio.

$$|wv_1| = \frac{1}{\cos\frac{\pi}{5}} \qquad |v_1v_2| = |v_2v_3| = 2\sin\frac{\pi}{5}\tan\frac{\pi}{5} \qquad |v_3v_4| = \frac{\sin\frac{\pi}{10}}{\sin\frac{3\pi}{5}}\tan\frac{\pi}{5} \qquad |v_4u| = \frac{\sin\frac{3\pi}{10}}{\sin\frac{3\pi}{5}}\tan\frac{\pi}{5}$$

Since we set $|uw| = 1$, the spanning ratio is simply $|wv_1| + |v_1v_2| + |v_2v_3| + |v_3v_4| + |v_4u| = \frac{1}{2}(11\sqrt{5}-17) \approx 3.798$. Note that the θ_5-graph with just these 5 vertices would have a far smaller spanning ratio, as there would be a lot of shortcut edges. However, a graph where this path is the shortest path between two vertices can be found in Appendix A. □

5 Lower bound on θ-routing

In this section, we show that always following the edge to the closest vertex in the cone that contains the destination can generate very long paths in the θ_5-graph. More formally, we look at the *competitiveness* of this routing algorithm. A routing algorithm is c-competitive on a graph G if for each pair of vertices in the graph, the routing algorithm finds a path of length at most c times the Euclidean distance between the two vertices.

We look at the competitiveness of the *θ-routing* algorithm, the standard routing algorithm for θ-graphs with at least seven cones: From the current vertex u, follow the edge to the closest vertex in T_{uw}, where w is the destination. This step is repeated until the destination is reached. We construct a θ_5-graph for which the θ-routing algorithm returns a path with spanning ratio $\Omega(n)$.

Lemma 5 *The θ-routing algorithm is not $o(n)$-competitive on the θ_5-graph.*

Proof. We construct the lower bound example on the competitiveness of the θ-routing algorithm on the θ_5-graph by repeatedly extending the routing path from source u to destination w. First, we place w such that the angle between uw and the bisector of T_{uw} is $\theta/4$. To ensure that the θ-routing algorithm does not follow the edge between u and w, we place a vertex v_1 in the upper corner of T_{uw} that is furthest from w. Next, to ensure that the θ-routing algorithm does not follow the edge between v_1 and w, we place a vertex v_1' in the upper corner of $T_{v_1 w}$ that is furthest from w. We repeat this step until we have created a cycle around w (see Figure 11a).

To extend the routing path further, we again place a vertex v_2 in the corner of the current canonical triangle. To ensure that the routing algorithm still routes to v_1 from u, we place v_2 slightly outside of T_{uv_1}. However, another problem arises: vertex v_1' is no longer the vertex closest to v_1 in $T_{v_1 w}$, as v_2 is closer. To solve this problem, we also place a vertex x_1 in $T_{v_1 v_2}$ such that v_1' lies in $T_{x_1 w}$ (see Figure 11b). By repeating this process four times, we create a second cycle around w.

To add more cycles around w, we repeat the same process as described above: place a vertex in the corner of the current canonical triangle and place an auxiliary vertex to ensure that the previous cycle stays intact. Note that when placing x_i, we also need to ensure that it does not lie in $T_{x_{i-1} w}$, to prevent shortcuts from being formed (see Figure 11b). This means that in general x_i does not lie arbitrarily close to the corner of $T_{v_i v_{i+1}}$.

This way we need to add auxiliary vertices only to the $(k-1)$-th cycle, when adding the k-th cycle, hence we can add an additional cycle using only a constant number of vertices. Since we can place the vertices arbitrarily close to the corners of the canonical triangles, we ensure that the distance to w stays almost the same, regardless of the number of cycles. Hence, each 'step' of the form v_i to v_i' via x_i has length $\cos(\theta/4)/\cos(\theta/2) \cdot |v_i w|$. Since $\cos(\theta/4)/\cos(\theta/2)$ is greater than 1 for the θ_5-graph and $|v_i w|$ can be arbitrarily close to $|uw|$, every step has length greater than $|uw|$. Using n points, we can construct $n/2$ of these steps and the total length of the path followed by the θ-routing algorithm is greater than $n/2 \cdot |uw|$. Thus the θ-routing algorithm is not $o(n)$-competitive on the constructed graph. □

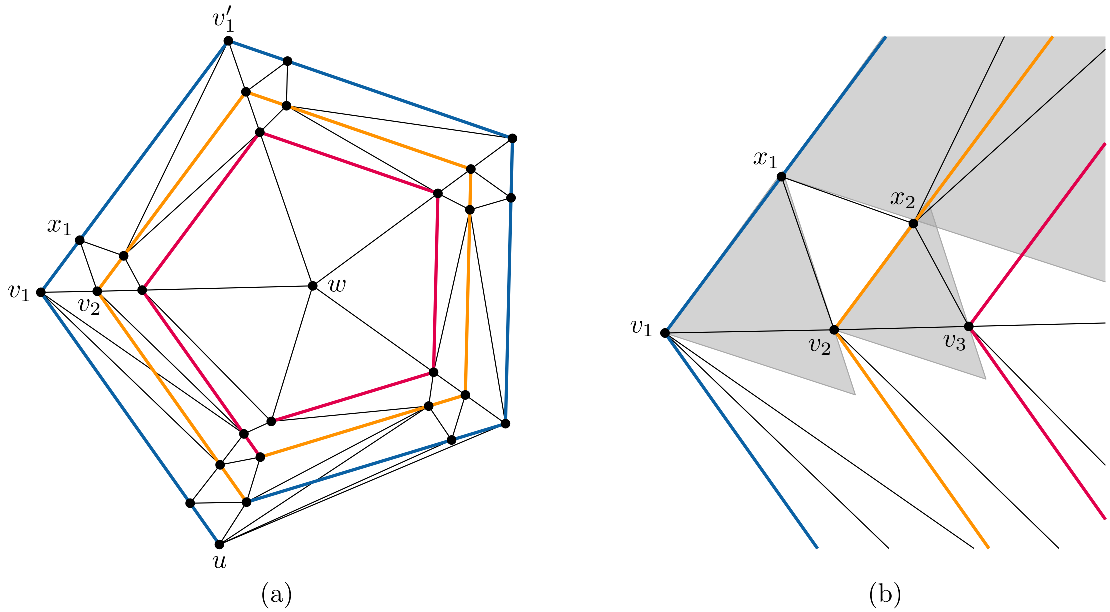


Figure 11: (a) A lower bound example for θ-routing on the θ_5-graph, consisting of three cycles: the first cycle is coloured blue, the second cycle is coloured orange, and the third cycle is coloured red. (b) The placement of vertices such that previous cycles stay intact when adding a new cycle.

6 Conclusions

We showed that there is a path between every pair of vertices in the θ_5-graph, of length at most $\sqrt{50+22\sqrt{5}} \approx 9.960$ times the straight-line distance between them. This is the first constant upper bound on the spanning ratio of the θ_5-graph, proving that it is a geometric spanner. We also presented a θ_5-graph with spanning ratio arbitrarily close to $\frac{1}{2}(11\sqrt{5}-17) \approx 3.798$, thereby giving a lower bound on the spanning ratio. There is still a significant gap between these bounds, which is caused by the upper bound proof mostly ignoring the main obstacle to improving the lower bound: that every edge requires at least one of its canonical triangles to be empty. Hence we believe that the true spanning ratio is closer to the lower bound.

While our proof for the upper bound on the spanning ratio returns a spanning path between the two vertices, it requires knowledge of the neighbours of both the current vertex and the destination vertex. This means that the proof does not lead to a local routing strategy that can be applied in, say, a wireless setting. This raises the question whether it is possible to route *competitively* on this graph, i.e. to discover a spanning path from one vertex to another by using only information local to the current vertices visited so far.

References

[1] Giri Narasimhan and Michiel Smid. *Geometric spanner networks*. Cambridge University Press, 2007.

[2] B. E. Flinchbaugh and L. K. Jones. Strong connectivity in directional nearest-neighbor graphs. *SIAM Journal on Algebraic and Discrete Methods*, 2(4):461–463, 1981.

[3] Andrew Chi Chih Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4):721–736, 1982.

[4] L. Paul Chew. There is a planar graph almost as good as the complete graph. In *Proceedings of the 2nd Annual Symposium on Computational Geometry (SoCG 1986)*, pages 169–177, 1986.

[5] Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9(1):81–100, 1993.

[6] Kenneth L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proceedings of the 19th Annual ACM Symposium on the Theory of Computing (STOC 1987)*, pages 56–65, 1987.

[7] Prosenjit Bose, Anil Maheshwari, Giri Narasimhan, Michiel Smid, and Norbert Zeh. Approximating geometric bottleneck shortest paths. *Computational Geometry: Theory and Applications*, 29(3):233–249, 2004.

[8] Prosenjit Bose, Mirela Damian, Karim Douïeb, Joseph O'Rourke, Ben Seamone, Michiel Smid, and Stefanie Wuhrer. $\pi/2$-angle Yao graphs are spanners. *ArXiv e-prints*, 2010. arXiv:1001.2913.

[9] J. Mark Keil. Approximating the complete Euclidean graph. In *Proceedings of the 1st Scandinavian Workshop on Algorithm Theory (SWAT 1988)*, volume 318 of *Lecture Notes in Computer Science*, pages 208–213, 1988.

[10] J. Mark Keil and Carl A. Gutwin. Classes of graphs which approximate the complete Euclidean graph. *Discrete & Computational Geometry*, 7(1):13–28, 1992.

[11] Jim Ruppert and Raimund Seidel. Approximating the d-dimensional complete Euclidean graph. In *Proceedings of the 3rd Canadian Conference on Computational Geometry (CCCG 1991)*, pages 207–210, 1991.

[12] Prosenjit Bose, André van Renssen, and Sander Verdonschot. On the spanning ratio of theta-graphs. In *Proceedings of the 13th Algorithms and Data Structures Symposium (WADS 2013)*, volume 8037 of *Lecture Notes in Computer Science*, pages 182–194, 2013.

[13] Prosenjit Bose, Jean-Lou De Carufel, Pat Morin, André van Renssen, and Sander Verdonschot. Optimal bounds on theta-graphs: More is not always better. In *Proceedings of the 24th Canadian Conference on Computational Geometry (CCCG 2012)*, pages 305–310, 2012.

[14] Peter Rysavy. Wireless broadband and other fixed-wireless systems. http://www.networkcomputing.com/netdesign/bb1.html. Accessed 30 June 2014.

[15] SATEL Oy. What is a radio modem? http://www.satel.com/products/what-is-a-radio-modem. Accessed 30 June 2014.

[16] Pat Morin and Sander Verdonschot. On the average number of edges in theta graphs. In *Proceedings of the 11th Meeting on Analytic Algorithmics and Combinatorics (ANALCO14)*, pages 121–132, 2014.

[17] Iyad Kanj. Geometric spanners: Recent results and open directions. In *Proceedings of the 3rd International Conference on Communications and Information Technology (ICCIT 2013)*, pages 78–82, 2013.

[18] Nawar M. El Molla. Yao spanners for wireless ad hoc networks. Master's thesis, Villanova University, 2009.

[19] Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and David Ilcinkas. Connections between theta-graphs, Delaunay triangulations, and orthogonal surfaces. In *Proceedings of the 36th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2010)*, volume 6410 of *Lecture Notes in Computer Science*, pages 266–278, 2010.

[20] Mirela Damian and Kristin Raudonis. Yao graphs span Theta graphs. *Discrete Mathematics, Algorithms and Applications*, 4(2):1250024, 2012.

[21] Prosenjit Bose, Mirela Damian, Karim Douïeb, Joseph O'Rourke, Ben Seamone, Michiel Smid, and Stefanie Wuhrer. $\pi/2$-angle Yao graphs are spanners. *International Journal of Computational Geometry & Applications*, 22(1):61–82, 2012.

[22] Luis Barba, Prosenjit Bose, Jean-Lou De Carufel, André van Renssen, and Sander Verdonschot. On the stretch factor of the Theta-4 graph. In *Proceedings of the 13th Algorithms and Data Structures Symposium (WADS 2013)*, volume 8037 of *Lecture Notes in Computer Science*, pages 109–120, 2013.

[23] 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. In *Proceedings of the 30th Annual Symposium on Computational Geometry (SoCG 2014)*, pages 30–39, 2014.

A Lower bound on the spanning ratio

#	Action	Shortest path
1	Start with a vertex v_1.	-
2	Add v_2 in C_0^u, such that v_2 is arbitrarily close to the top right corner of $T_{v_1v_2}$.	v_1v_2
3	Remove edge (v_1, v_2) by adding two vertices, v_3 and v_4, arbitrarily close to the counter-clockwise corners of $T_{v_1v_2}$ and $T_{v_2v_1}$.	$v_1v_4v_2$
4	Remove edge (v_1, v_4) by adding two vertices, v_5 and v_6, arbitrarily close to the clockwise corner of $T_{v_1v_4}$ and the counter-clockwise corner of $T_{v_4v_1}$.	$v_1v_3v_2$
5	Remove edge (v_2, v_3) by adding two vertices, v_7 and v_8, arbitrarily close to the clockwise corner of $T_{v_2v_3}$ and the counter-clockwise corner of $T_{v_3v_2}$.	$v_1v_6v_4v_2$
6	Remove edge (v_1, v_6) by adding two vertices, v_9 and v_{10}, arbitrarily close to the clockwise corner of $T_{v_1v_6}$ and the counter-clockwise corner of $T_{v_6v_1}$.	$v_1v_5v_4v_2$
7	Remove edge (v_4, v_5) by adding two vertices, v_{11} and v_{12}, arbitrarily close to the counter-clockwise corner of $T_{v_4v_5}$ and the clockwise corner of $T_{v_5v_4}$.	$v_1v_5v_6v_4v_2$
8	Remove edge (v_5, v_6) by adding two vertices, v_{13} and v_{14}, arbitrarily close to the counter-clockwise corner of $T_{v_5v_6}$ and the clockwise corner of $T_{v_6v_5}$.	$v_1v_5v_{14}v_6v_4v_2$
9	Remove edge (v_5, v_{14}) by adding two vertices, v_{15} and v_{16}, arbitrarily close to the counter-clockwise corner of $T_{v_5v_{14}}$ and the clockwise corner of $T_{v_{14}v_5}$.	$v_1v_5v_{13}v_6v_4v_2$
10	Remove edge (v_6, v_{13}) by adding two vertices, v_{17} and v_{18}, arbitrarily close to the clockwise corner of $T_{v_6v_{13}}$ and the counter-clockwise corner of $T_{v_{13}v_6}$.	$v_1v_3v_8v_2$
11	Remove edge (v_2, v_8) by adding a vertex v_{19} in the union of, and arbitrarily close to the intersection point of $T_{v_2v_8}$ and $T_{v_8v_2}$.	$v_1v_3v_7v_2$
12	Remove edge (v_3, v_7) by adding two vertices, v_{20} and v_{21}, arbitrarily close to the counter-clockwise corner of $T_{v_3v_7}$ and the clockwise corner of $T_{v_7v_3}$.	$v_1v_5v_{12}v_2$
13	Remove edge (v_2, v_{12}) by adding a vertex v_{22} arbitrarily close to the counter-clockwise corner of $T_{v_2v_{12}}$.	$v_1v_{10}v_6v_4v_2$
14	Remove edge (v_1, v_{10}) by adding a vertex v_{23} in the union of $T_{v_1v_{10}}$ and $T_{v_{10}v_1}$, arbitrarily close to the top boundary of $C_1^{v_{10}}$, and such that v_1 lies in $C_1^{v_{23}}$, arbitrarily close to the bottom boundary.	$v_1v_5v_{12}v_4v_2$
15	Remove edge (v_4, v_{12}) by adding two vertices, v_{24} and v_{25}, arbitrarily close to the counter-clockwise corner of $T_{v_4v_{12}}$ and the clockwise corner of $T_{v_{12}v_4}$.	$v_1v_5v_{13}v_{14}v_6v_4v_2$
16	Remove edge (v_{13}, v_{14}) by adding two vertices, v_{26} and v_{27}, arbitrarily close to the clockwise corner of $T_{v_{13}v_{14}}$ and the counter-clockwise corner of $T_{v_{14}v_{13}}$.	$v_1v_9v_{18}v_6v_4v_2$
17	Remove edge (v_9, v_{18}) by adding two vertices, v_{28} and v_{29}, arbitrarily close to the clockwise corner of $T_{v_9v_{18}}$ and the counter-clockwise corner of $T_{v_{18}v_9}$.	$v_1v_5v_{16}v_{11}v_4v_2$
18	Remove edge (v_{11}, v_{16}) by adding two vertices, v_{30} and v_{31}, arbitrarily close to the counter-clockwise corner of $T_{v_{11}v_{16}}$ and the clockwise corner of $T_{v_{16}v_{11}}$.	$v_1v_{23}v_{10}v_6v_4v_2$

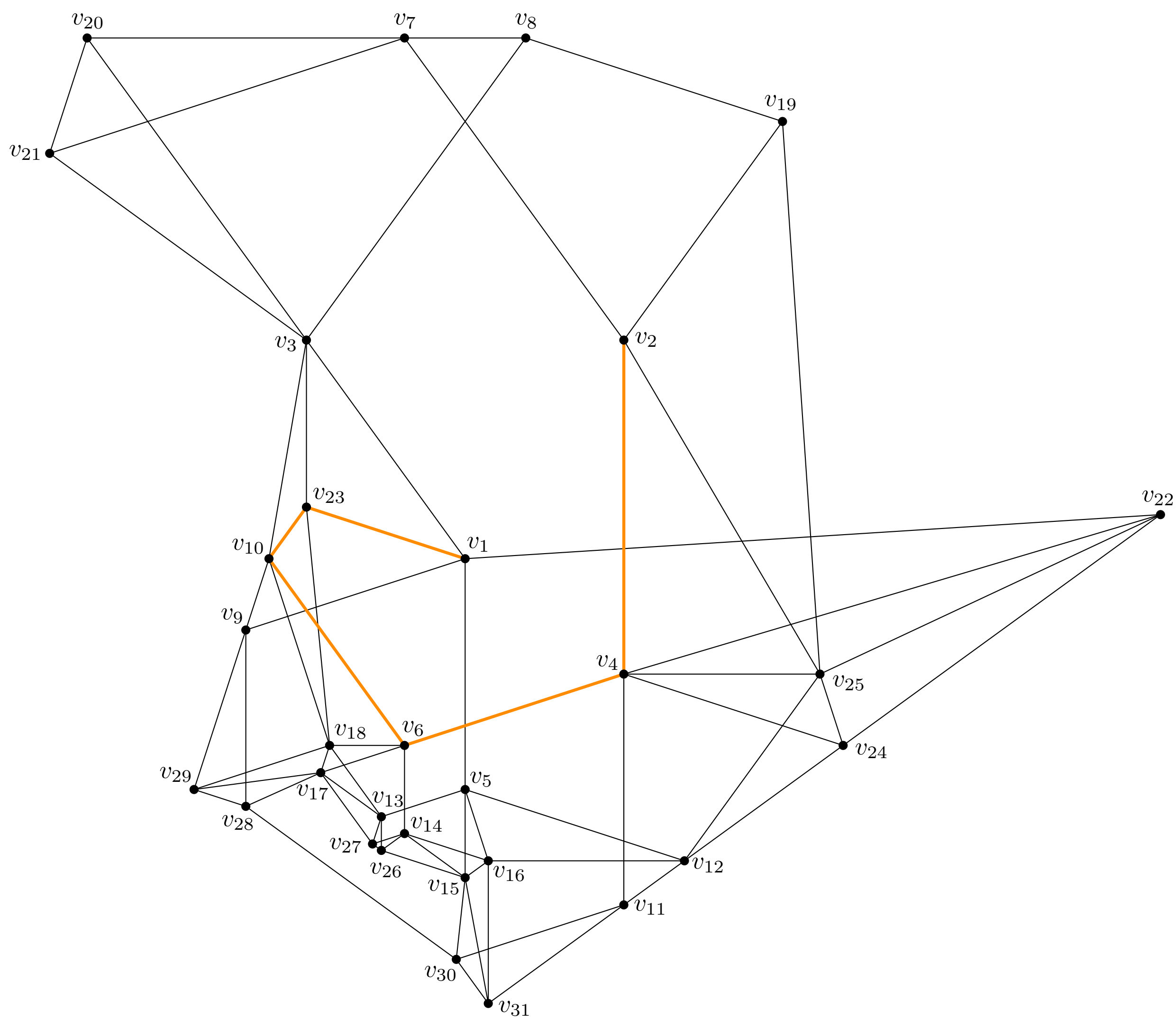


Figure 12: A θ_5-graph with a spanning ratio that matches the lower bound. The shortest path between v_1 and v_2 is indicated in orange.