Competitive Routing in the Half-θ_6-Graph*

Prosenjit Bose[‡] Rolf Fagerberg[§] André van Renssen[‡] Sander Verdonschot[‡]

Abstract

We present a deterministic local routing scheme that is guaranteed to find a path between any pair of vertices in a half-θ_6-graph whose length is at most $5/\sqrt{3} = 2.886...$ times the Euclidean distance between the pair of vertices. The half-θ_6-graph is identical to the Delaunay triangulation where the empty region is an equilateral triangle. Moreover, we show that no local routing scheme can achieve a better competitive spanning ratio thereby implying that our routing scheme is optimal. This is somewhat surprising because the spanning ratio of the half-θ_6-graph is 2. Since every triangulation can be embedded in the plane as a half-θ_6-graph using $O(\log n)$ bits per vertex coordinate via Schnyder's embedding scheme (SODA 1990), our result provides a competitive local routing scheme for every such embedded triangulation.

1 Introduction

A fundamental problem in networking is the routing of a message from one vertex to another in a graph. What makes routing more challenging is that often in a network the routing strategy must be *local*. Informally, a routing strategy is *local*, when the routing algorithm must decide which vertex to forward a message to based solely on knowledge of the source and destination vertex, the current vertex and all vertices directly connected to the current vertex. Routing algorithms are considered *geometric* when the underlying graph is embedded in the plane. Edges are segments connecting pairs of points and are weighted by the Euclidean distance between their endpoints. Geometric routing algorithms are important in wireless sensor networks (see [13] and [15] for surveys of the area) since they offer routing strategies that use the coordinates of the vertices to help guide the search as opposed to using the more traditional routing tables.

Papadimitriou and Ratajczak [14] posed a tantalizing question in this area that led to a flurry of activity: Does every 3-connected planar graph have a straight-line embedding in the plane that admits a local routing strategy such as greedy[1] routing? They provided a partial answer by showing that 3-connected planar graphs can always be embedded in $\mathbb{R}^3$ such that they admit a greedy routing strategy. They also showed that the class of complete bipartite graphs, $K_{k,6k+1}$ for all $k \geq 1$ cannot be embedded such that greedy routing always succeeds since every embedding has at least one vertex that is not connected to its nearest neighbor. Bose and Morin [4] showed that greedy routing always succeeds on Delaunay triangulations. In fact, a slightly restricted greedy routing strategy known as *greedy-compass* is the first local routing strategy shown to succeed on all triangulations [3]. Dhandapani [6] proved the existence of an embedding that admits greedy routing for every triangulation and Angelini et al.[1] provided a constructive proof. Leighton and Moitra [12] settled Papadimitriou and Ratajczak's question by showing that every 3-connected planar graph can be embedded in the plane such that greedy routing succeeds. One drawback of these embedding algorithms is that the coordinates require $\Omega(n \log n)$ bits per vertex. To address this, He and Zhang [9] and Goodrich and Strash [8] gave succinct embeddings using only $O(\log n)$ bits per vertex. Recently, He and Zhang [10] showed that every 3-connected plane graph admits a succinct convex embedding[2] on which a slightly modified greedy routing strategy always succeeds.

In light of these recent successes, it is surprising to note that the above routing strategies have solely concentrated on finding an embedding that guarantees a local routing strategy will succeed. No attention is paid to the quality of the resulting path relative to the shortest path. None of the above routing strategies have been shown to be competitive[3]. Bose and Morin

*Research supported in part by NSERC and the Danish Council for Independent Research, Natural Sciences.

‡School of Computer Science, Carleton University. Email: jit@scs.carleton.ca, avrensse@connect.carleton.ca, sverdons@connect.carleton.ca.

§Department of Mathematics and Computer Science, University of Southern Denmark. Email: rolf@imada.sdu.dk.

[1] A routing strategy is greedy when a message is always forwarded to the vertex whose distance to the destination is the smallest among all vertices in the neighborhood of the current vertex including the current vertex.

[2] An embedding of a planar graph is convex when every face is convex.

[3] A routing strategy is competitive if the path found by the routing strategy is not more than a constant (the competitive spanning ratio) times the shortest path. The competitive spanning ratio of a graph is defined as the minimum competitive spanning ratio over all routing strategies.

[4] show that many local routing strategies are not competitive but show how to route competitively on the Delaunay triangulation. However, Dillencourt [7] showed that not all triangulations can be embedded in the plane as Delaunay triangulations. This raises the following question: can **every** triangulation be embedded in the plane such that it admits a competitive local routing strategy? We answer this question in the affirmative.

The half-θ_6-graph was introduced by Bonichon et al. [2] who showed that it is identical to the Delaunay triangulation where the empty region is an equilateral triangle. Our main result is a deterministic local routing scheme that is guaranteed to find a path between any pair of vertices in a half-θ_6-graph whose length is at most $5/\sqrt{3} = 2.886...$ times the Euclidean distance between the pair of vertices. On the way to proving our main result, we uncover some local properties of spanning paths in the half-θ_6-graph. Since Schnyder [16] showed that every triangulation can be embedded in the plane as a half-θ_6-graph using $O(\log n)$ bits per vertex coordinate, our main result implies that a competitive local routing scheme exists for every triangulation embedded as such. Moreover, we show that no local routing scheme can achieve a better competitive spanning ratio on half-θ_6-graphs, implying that our routing scheme is optimal. This is somewhat surprising because Chew [5] showed that the spanning ratio of the half-θ_6-graph is at most 2. Thus, our lower bound provides a separation between the spanning ratio of the half-θ_6-graph and the competitive spanning ratio of any local routing scheme on the half-θ_6-graph. Finally, we conclude by highlighting some similarities and differences between the half-θ_6-graph and the full-θ_6-graph [11].

2 Preliminaries

In a weighted graph G, let the distance $d_G(u, v)$ between two vertices u and v be the length of the shortest path between u and v in G. A subgraph H of G is a *t-spanner* of G if for all pairs of vertices u and v, $d_H(u, v) \leq t \cdot d_G(u, v), t \geq 1$. Its *spanning ratio* is the smallest t for which it is a t-spanner. The graph G is referred to as the *underlying graph*.

We consider the situation where the underlying graph G is a straightline embedding of the complete graph on a set of n points in the plane denoted by K_n, with the weight of an edge (u, v) being the Euclidean distance $|uv|$ between u and v. A spanner of such a graph is called a *geometric spanner*. In this paper, we show how to route competitively on the geometric spanner called the half-θ_6-graph [2]. To define this graph, we need the following terminology.

Let a *cone* C be the region in the plane between two rays originating from the same point (referred to as the apex of the cone). For each vertex u of K_n consider the six rays originating from u with angles to the positive x-axis being multiples of $\pi/3$. Each pair of consecutive rays defines a cone. Let $\overline{C}_1, C_0, \overline{C}_2, C_1, \overline{C}_0, C_2$ be the sequence of cones in counterclockwise order starting from the positive x-axis, as depicted in Figure 1, and call C_0, C_1, and C_2 *positive* cones, and $\overline{C}_0$, $\overline{C}_1$, and $\overline{C}_2$ *negative* cones. The names are chosen such that using modulo 3 arithmetic on the indices, a positive cone C_i has the negative cone $\overline{C}_{i+1}$ ($\overline{C}_{i-1}$) as its clockwise (counterclockwise) neighbor. An analogous statement holds for a negative cone $\overline{C}_i$. When the apex is not clear from the context, we use C_i^u to denote cone C_i with apex u.

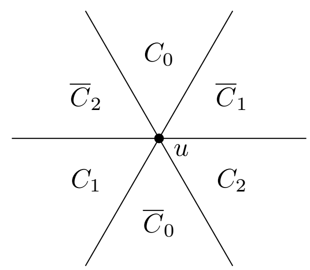


Figure 1: The cones having apex u.

The half-θ_6-graph [2] is constructed as follows: for each of the three positive cones of each vertex u, add an edge from u to the closest vertex in that cone, where distance is measured along the bisector of the cone. More formally, we add an edge between two vertices u and v if $v \in C_i^u$ and for all points $w \in C_i^u$ ($v \neq w$), $|uv'| \leq |uw'|$, where v' and w' denote the orthogonal projection of v and w, respectively, on the bisector of C_i^u.

For ease of exposition, we only consider point sets in general position: no two points lie on a line parallel to one of the rays that define the cones. This implies that each vertex adds at most one edge per positive cone to the graph, and hence there are at most $3n$ edges in total.

Given a vertex w in a positive cone C_i^u of vertex u, we define the *canonical equilateral triangle* T_{uw} to be the triangle defined by the borders of C_i^u and the line through w perpendicular to the bisector of C_i^u. Figure 2 gives an example with $w \in C_0^u$. Note that for any pair of vertices u and w, either w lies in a positive cone of u, or u lies in a positive cone of w, so there is exactly one canonical equilateral triangle (either T_{uw} or T_{wu}) for the pair.

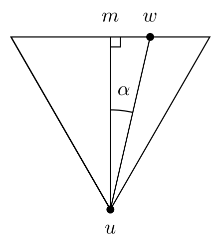


Figure 2: The canonical equilateral triangle T_{uw} of u and w, its bisector um, and the angle α between uw and the bisector.

3 Spanning Ratio of the Half-θ_6-Graph

Bonichon *et al.* [2] showed that the half-θ_6-graph is equivalent to the Delaunay triangulation based on empty equilateral triangles, which is known to have spanning ratio 2 [5]. In this section, we provide an alternative proof of the spanning ratio of the half-θ_6-graph. Our proof shows that between any pair of points u, w, there always exists a path with spanning ratio 2 that lies in the canonical triangle. This is a key property used by our routing algorithm.

For a pair of vertices u and w, our bound is expressed in terms of the angle α between the line from u to w and the bisector of their canonical equilateral triangle. See Figure 2.

Theorem 3.1. *Let u and w be vertices with w in a positive cone of u. Let m be the midpoint of the side of T_{uw} opposing u, and let α be the unsigned angle between the lines uw and um. There exists a path in the half-θ_6-graph of length at most*

$$(\sqrt{3} \cdot \cos\alpha + \sin\alpha) \cdot |uw|$$

where all vertices on this path lie in T_{uw}.

The expression $\sqrt{3} \cdot \cos\alpha + \sin\alpha$ is increasing for $\alpha \in [0, \pi/6]$. Inserting the extreme value $\pi/6$ for α, we arrive at the following.

Corollary 3.1. *The spanning ratio of the half-θ_6-graph is 2.*

We note that the bounds of Theorem 3.1 and Corollary 3.1 are tight: for all values of $\alpha \in [0, \pi/6]$ there exists a point set for which the shortest path in the half-θ_6-graph for some pair of vertices u and w has length arbitrarily close to $(\sqrt{3} \cdot \cos\alpha + \sin\alpha) \cdot |uw|$. A simple example appears later in the proof of Theorem 4.1.

Proof of Theorem 3.1. Given two vertices u and w, we assume w.l.o.g. that w lies in C_0^u. We prove the theorem by induction on the area of T_{uw} (formally, induction on the rank, when ordered by area, of the triangles T_{xy} for all pairs of points x and y). Let a and b be the upper left and right corner of T_{uw}, and let $A = T_{uw} \cap C_1^w$ and $B = T_{uw} \cap C_2^w$ (see Figure 3).

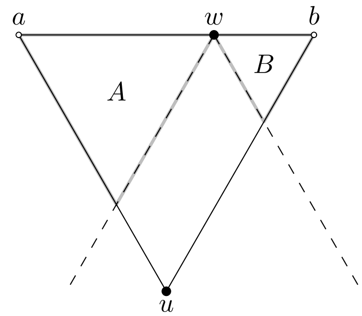


Figure 3: The corners a and b, and the areas A and B.

Our inductive hypothesis is the following, where $\delta(u, w)$ denotes the length of the shortest path from u to w in the half-θ_6-graph with all vertices lying in T_{uw}:

- If A is empty, then $\delta(u, w) \leq |ub| + |bw|$.
- If B is empty, then $\delta(u, w) \leq |ua| + |aw|$.
- If neither A nor B is empty, then $\delta(u, w) \leq \max\{|ua| + |aw|, |ub| + |bw|\}$.

We first note that this induction hypothesis implies Theorem 3.1: using the side of T_{uw} as the unit of length, we have $|wm| = |uw| \cdot \sin\alpha$ and $\sqrt{3}/2 = |um| = |uw| \cdot \cos\alpha$ (see Figure 2), hence the induction hypothesis gives that $\delta(u, w)$ is at most $1 + 1/2 + |wm| = \sqrt{3} \cdot (\sqrt{3}/2) + |wm| = (\sqrt{3} \cdot \cos\alpha + \sin\alpha) \cdot |uw|$.

Base case: T_{uw} has rank 1. Since the triangle is a smallest triangle, w is the closest vertex to u in a positive cone of u. Hence the edge (u, w) is in the half-θ_6-graph, and $\delta(u, w) = |uw|$. From the triangle inequality, we have $|uw| \leq \min\{|ua| + |aw|, |ub| + |bw|\}$, so the induction hypothesis holds.

Induction step: We assume that the induction hypothesis holds for all pairs of points with canonical triangles of rank up to k. Let T_{uw} be a canonical triangle of rank $k + 1$.

If (u, w) is an edge in the half-θ_6-graph, the induction hypothesis follows by the same argument as in the base case. If there is no edge between u and w, let v be the vertex closest to u in the positive cone C_0^u, and let a' and b' be the upper left and right corner of T_{uv}, respectively. See Figure 5. By definition, $\delta(u, w) \leq |uv| + \delta(v, w)$, and by the triangle inequality, $|uv| \leq \min\{|ua'| + |a'v|, |ub'| + |b'v|\}$.

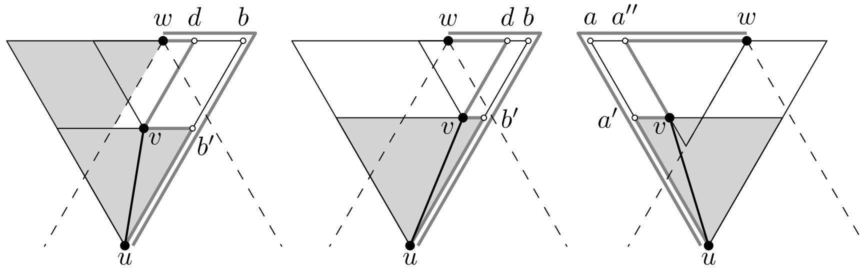
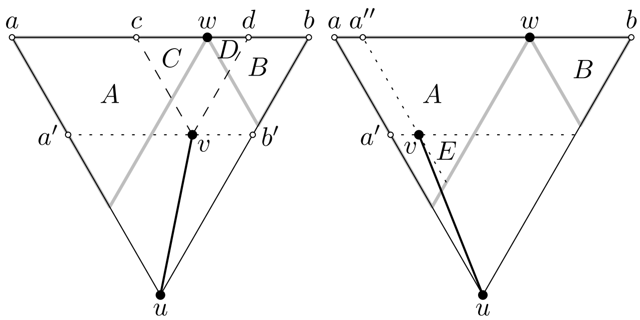


Figure 4: Visualization of the path inequalities. Thick, dark gray lines signify paths occurring in the inequalities, and light gray areas indicate emptiness.



(a) (b)

Figure 5: The two cases: (a) v lies in neither A nor B, (b) v lies in A.

We perform a case analysis based on the location of v: (a) v lies neither in A nor in B, (b) v lies inside A, and (c) v lies inside B. Case (c) is analogous to Case (b), so we only discuss the first two cases.

Case (a): Let c and d be the upper left and right corner of T_{vw}, respectively, and let $C = T_{vw} \cap C_1^w$ and $D = T_{vw} \cap C_2^w$. See Figure 5. Since T_{vw} has smaller area than T_{uw}, we apply the inductive hypothesis on T_{vw}. Our task is to prove all three statements of the inductive hypothesis for T_{uw}.

1. If A is empty, then C is also empty, so by induction $\delta(v, w) \leq |vd| + |dw|$. Since v, d, b, and b' form a parallelogram, we have:

$$\begin{aligned}
&(3.1) \quad \delta(u, w) &\leq\ & |uv| + \delta(v, w) \\
&(3.2) & \leq\ & |ub'| + |b'v| + |vd| + |dw| \\
&(3.3) & =\ & |ub| + |bw|,
\end{aligned}$$

which proves the first statement of the induction hypothesis. This argument is illustrated in Figure 4, left, where thick, dark grey lines signify paths occurring in the inequalities above, and light gray areas indicate emptiness.

2. If B is empty, an analogous argument proves the second statement of the induction hypothesis.

3. If neither A nor B is empty, by induction we have $\delta(v, w) \leq \max\{|vc| + |cw|, |vd| + |dw|\}$. Assume, without loss of generality, that the maximum of the right hand side is attained by its second argument $|vd| + |dw|$ (the other case is analogous). Since vertices v, d, b, and b' form a parallelogram, we have that:

$$\begin{aligned}
&(3.4) \quad \delta(u, w) &\leq\ & |uv| + \delta(v, w) \\
&(3.5) & \leq\ & |ub'| + |b'v| + |vd| + |dw| \\
&(3.6) & \leq\ & |ub| + |bw| \\
&(3.7) & \leq\ & \max\{|ua| + |aw|, |ub| + |bw|\},
\end{aligned}$$

which proves the third statement of the induction hypothesis. This argument is illustrated in Figure 4, middle.

Case (b): Let $E = T_{uv} \cap T_{wv}$, and let a'' be the upper left corner of T_{wv}. See Figure 5. Since v is the closest vertex to u in the positive cone C_0^u, T_{uv} is empty. Hence, E is empty. Since T_{wv} is smaller than T_{uw}, we can apply induction on it. As E is empty, the first statement of the induction hypothesis for T_{wv} gives $\delta(v, w) \leq |va''| + |a''w|$. Since $|uv| \leq |ua'| + |a'v|$ and v, a'', a, and a' form a parallelogram, we have that $\delta(u, w) \leq |ua| + |aw|$, proving the second and third statement in the induction hypothesis for T_{uw}. This argument is illustrated in Figure 4, right. Since v lies in A, the first statement in the induction hypothesis for T_{uw} is vacuously true. ☐

4 Routing in the Half-θ_6-Graph

In this section, we give matching upper and lower bounds for the competitive routing ratio on the half-θ_6-graph. We begin by defining our model. A deterministic k-local routing scheme defined on a graph G is a function $f(u, t, N_k(u))$ specifying the vertex the message

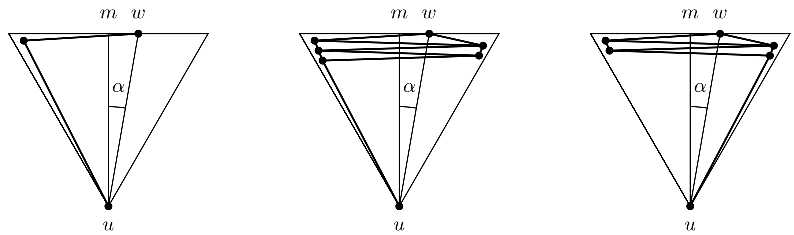


Figure 6: Lower bound instances.

should be forwarded to given that u is the current vertex, t is the target or destination vertex, and $N_k(u)$ is the k-neighborhood of u. The k-neighborhood of a vertex u is the set of vertices in the graph that can be reached from u by following at most k edges. When $k = 1$, we drop the value and refer to the scheme as a local routing scheme. We note that in the literature, many variants of this model have been studied where the routing scheme knows which node sent the message to u or the scheme has some memory. However, we study one of the weakest models and show that it is still possible to route competitively. Our upper bounds hold for $k = 1$ and our lower bounds hold for any fixed k. Since our graphs are geometric, the identifier for a vertex is its coordinates in the plane. The *competitive routing ratio* of a routing scheme is defined analogously to the spanning ratio as the smallest $t \geq 1$ for which no route computed by the routing scheme between any pair of vertices is longer than t times the Euclidean distance between that pair. Our bounds are expressed in terms of the angle α between the line from the source to the destination point and the bisector of their canonical equilateral triangle. See Figure 2.

THEOREM 4.1. *Let u and w be vertices with w in a positive cone of u. Let m be the midpoint of the side of T_{uw} opposing u, and let α be the unsigned angle between the lines uw and um. There is a local routing scheme on the half-θ_6-graph for which every path followed has length at most*

i) $(\sqrt{3} \cdot \cos\alpha + \sin\alpha) \cdot |uw|$ *when routing from u to w,*

ii) $(5/\sqrt{3} \cdot \cos\alpha - \sin\alpha) \cdot |uw|$ *when routing from w to u,*

and this is best possible for deterministic k-local routing schemes.

The first expression is increasing for $\alpha \in [0, \pi/6]$, while the second expression is decreasing. Inserting the extreme values $\pi/6$ and 0 for α, we get the following worst case version of Theorem 4.1.

COROLLARY 4.1. *Let u and w be two vertices with w in a positive cone of u. There is a local routing scheme on the half-θ_6-graph with routing ratio*

i) 2 *when routing from u to w,*

ii) $5/\sqrt{3} = 2.886\ldots$ *when routing from w to u,*

and this is best possible for deterministic k-local routing schemes.

Since the spanning ratio of the half-θ_6-graph is 2, the second lower bound shows a separation between the spanning ratio and the competitive routing ratio in the half-θ_6-graph.

Since every triangulation can be embedded in the plane as a half-θ_6-graph using $O(\log n)$ bits per vertex via Schnyder's embedding scheme [16], an important implication of Theorem 4.1 is the following.

COROLLARY 4.2. *Every n-vertex triangulation can be embedded in the plane using $O(\log n)$ bits per coordinate such that the embedded triangulation admits a deterministic local routing scheme with competitive routing ratio at most $5/\sqrt{3}$.*

In the remainder of this section, we prove Theorem 4.1. We first prove the lower bounds, then describe the routing scheme, and finally prove the stated upper bounds.

Lower bound. Let the side of T_{uw} be the unit of length. From Figure 2 we have $|wm| = |uw| \cdot \sin\alpha$ and $\sqrt{3}/2 = |um| = |uw| \cdot \cos\alpha$. From Instance 1 in Figure 6, the spanning ratio of the half-θ_6-graph is at least $1 + 1/2 + |wm| = \sqrt{3} \cdot (\sqrt{3}/2) + |wm| = (\sqrt{3} \cdot \cos\alpha + \sin\alpha) \cdot |uw|$ (since the point in the upper left corner of T_{uw} can be moved arbitrarily close to the corner). This is a lower bound for any routing scheme, and proves our statement on routing from u to w.

For routing from w to u, consider instances 2a and 2b in Figure 6. Any deterministic 1-local routing scheme only has information about direct neighbors, hence cannot distinguish between the two instances when routing out of w. So a deterministic algorithm must route to the same neighbor of w in both instances, and either choice of neighbor leads to a non-optimal route in one of the two instances. The smallest loss occurs when the choice is towards the closest corner of T_{uw} (making instance 2a the hard instance), which gives a lower bound of $(1/2 - |wm|) + 1 + 1 = 5/2 - |wm| = (5/\sqrt{3} \cdot \cos\alpha - \sin\alpha) \cdot |uw|$ (since the points in the corners of T_{uw} can be moved arbitrarily close to the corners while keeping their relative positions). In fact, this is precisely where the separation occurs between the spanning ratio of 2 and the competitive routing ratio. Note that by extending instances 2a and 2b of Figure 6 to have $\Omega(k)$ points close to the corners such that u is not in the k-neighborhood of w, the lower bound holds for any deterministic k-local routing scheme.

Routing scheme. We let s denote the current vertex, and t the fixed destination. The routing scheme needs to determine which edge (s, v) to follow next. We say we are currently *routing positively* (*negatively*) when t is in a positive (negative) cone of s. For ease of description, we assume w.l.o.g. that t is in cone C_0 of s when routing positively, and in cone $\overline{C}_0$ of s when routing negatively. When routing positively, T_{st} intersects only C_0 among the cones of s. When routing negatively, T_{ts} intersects $\overline{C}_0$ as well as the two positive cones C_1 and C_2 of s. We let $X_0 = \overline{C}_0 \cap T_{ts}$, $X_1 = C_1 \cap T_{ts}$, and $X_2 = C_2 \cap T_{ts}$. We let a be the corner of T_{ts} contained in X_1 and b the corner of T_{ts} contained in X_2. These definitions are illustrated in Figure 7.

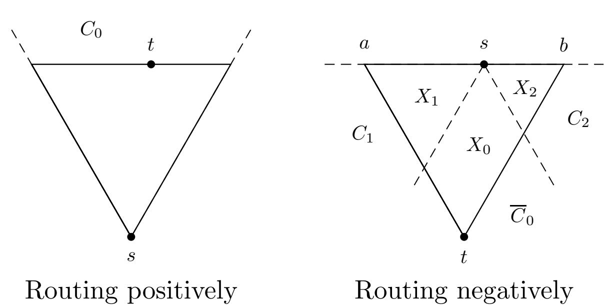


Figure 7: Routing terminology.

The routing scheme will only follow edges (s, v) where v lies in the canonical equilateral triangle of s and t. Routing positively is straightforward since there is exactly one edge (s, v) with $v \in T_{st}$, by the construction of the half-θ_6-graph. The challenge is to route negatively. When routing negatively, at least one edge (s, v) with $v \in T_{ts}$ exists, since s and t are connected by a path in T_{ts}, according to Theorem 3.1. The core of our routing scheme is how to choose which edge to follow when there are more than one. Intuitively, when routing negatively, our scheme tries to select an edge that makes measurable progress towards the destination. When no such edge exists, we are forced to take an edge that does not make measurable progress, however we are able to then deduce that certain regions within the canonical triangle are empty. This allows us to prove that we take such an edge at most once. In essence, we prove that we can go in the wrong direction only once. This can be seen in the instances 2a and 2b in Figure 6 where an adversary forces a routing scheme to go in the wrong direction once. We provide a formal description of our routing scheme below.

Our routing algorithm can be in one of four states. We call the situation when routing positively state A, and divide the situation when routing negatively into three states B, C, and D, as follows: By construction of the half-θ_6-graph, there is at most one edge (s, v) with $v \in X_1$ and the same applies to X_2 since they are both positive cones of s. Let state B be the case where both X_1 and X_2 contain an edge, state C the case where exactly one contains an edge, and state D the case where both are empty. At the start of a routing step, we are in exactly one of the states A, B, C, or D. Routing in state A is straightforward. We now describe routing in states B, C, and D.

In state B, the routing scheme first tries to follow an edge (s, v) with $v \in X_0$. If several such edges exist, an arbitrary one of these is followed. If no such edge exists, the routing scheme follows an edge in X_1 or X_2: if $|as| \le |sb|$, it follows the single edge (s, v) with $v \in X_1$; if $|as| > |sb|$, it follows the single edge (s, v) with $v \in X_2$. In short, the routing scheme favors moving towards the closest corner of T_{ts} when it is not able to move directly towards t. Note that this choice is made specifically to ensure we can bound the total distance travelled when faced with instances similar to 2a and 2b in Figure 6.

In states C and D, the routing scheme first tries to follow an edge (s, v) with $v \in X_0$. If several such edges exist, it chooses a specific one based on what we call the *projected length on a neighboring cone*. Let $\vec{e}_1$ ($\vec{e}_2$) be a unit vector in the direction of the ray from s constituting the border of $\overline{C}_0$ and C_1 (C_2). Since $\vec{e}_1$ and $\vec{e}_2$ are linearly independent, the vector $\vec{sv}$ can be written uniquely as $l_1 \cdot \vec{e}_1 + l_2 \cdot \vec{e}_2$. We define the projected length of the edge (s, v) on the neighboring cone C_1 (C_2) to be l_1 (l_2). Figure 9 illustrates the projected length on C_1.

In state C, exactly one of X_1 or X_2 is empty. If there exist edges (s, v) with $v \in X_0$, the routing scheme will follow one of these, choosing among them in the following way: If X_1 is empty, it chooses the edge with largest projected length on C_1. Else X_2 is empty, and it chooses the edge with largest projected length on C_2.

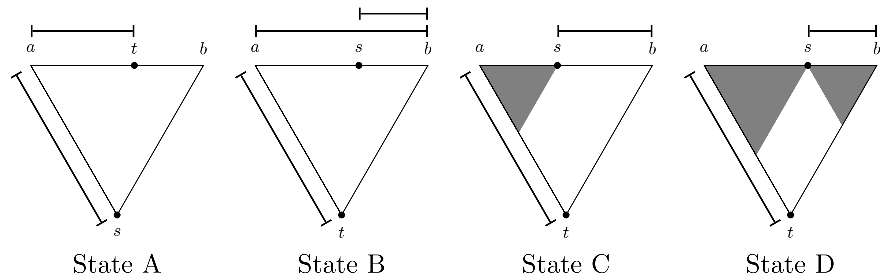


Figure 8: The potential ϕ in each of the states.

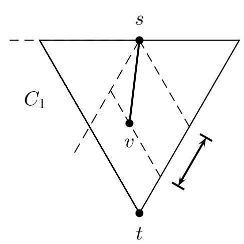


Figure 9: Projected length on C_1.

In short, the routing scheme favors staying close to the empty side of T_{ts}. If no edges (s, v) with $v \in X_0$ exist, the routing scheme follows the single edge (s, v) with v in $X_1 \cup X_2$.

In state D, both X_1 and X_2 are empty, so there must exist edges (s, v) with $v \in X_0$, as s and t are connected by a path in T_{ts}. If $|as| \geq |sb|$, the routing scheme follows the edge with largest projected length on C_1; if $|as| < |sb|$, it follows the edge with largest projected length on C_2. In short, when both sides of T_{ts} are empty, the routing scheme favors staying close to the largest empty side of T_{ts}.

Upper bound. The proof of the upper bound uses a potential function ϕ, defined as follows for each of the states A, B, C, and D, where a and b are the corners of T_{st} (T_{ts}) different from s (t), and $x \in \{a, b\}$ in state C is the corner contained in the non-empty one of the two areas X_1 and X_2.

State A: $\phi = |sa| + \max(|at|, |tb|)$
State B: $\phi = |ta| + |ab| + \min(|as|, |sb|)$
State C: $\phi = |ta| + |sx|$
State D: $\phi = |ta| + \min(|as|, |sb|)$

This definition is illustrated in Figure 8, where barlines designate potential and gray areas are empty. We will refer to the first term of ϕ (i.e., $|sa|$ in state A, $|ta|$ in states B, C, and D) as the *vertical part* of ϕ and to the rest as the *horizontal part*.

Our aim is to prove that for any routing step, the reduction in ϕ is at least as large as the length of the edge followed. Since ϕ is always non-negative, this will imply that no path followed can be longer than the initial value of ϕ. As all edges have strictly positive length, the routing scheme must terminate. Furthermore, the initial values of ϕ in state A and state B are exactly the path lengths appearing in the calculations leading to the lower bound of instances 1 and 2a/2b, respectively, so the same bounds apply, but now as upper bounds. The initial values of ϕ in states C and D are smaller than the initial value in state B, and hence can only lead to better routing bounds. This gives the upper bounds stated in Theorem 4.1.

What remains is to prove the statement that for any routing step, the reduction in ϕ is at least as large as the length of the edge followed. We do this by case analysis of the possible routing steps. One simple observation that is repeatedly used in our analysis is summarized in the fact below.

FACT 4.1. *In an equilateral triangle, the side length is the diameter, i.e. the longest distance defined by any two points in the triangle.*

State A. For a routing step starting in state A, v is either in a negative or a positive cone of t. The first situation leads to state A again. The second leads to state C or D, since the area of T_{st} between s and v must be empty by construction of the half-θ_6-graph. These situations are illustrated in Figure 10.

For the case ending in state A, the reduction of the vertical part of ϕ is at least as large as $|sv|$ by Fact 4.1. The horizontal part of ϕ can only decrease during the step. Hence the statement holds for this case. The same type of analysis proves the statement for the case ending in state C. For the case ending in state D, the final value of ϕ can only be smaller than for the case ending in state C, so again the statement holds.

State B. A routing step starting in state B cannot lead to state A, as the step stays within T_{ts}, but it may

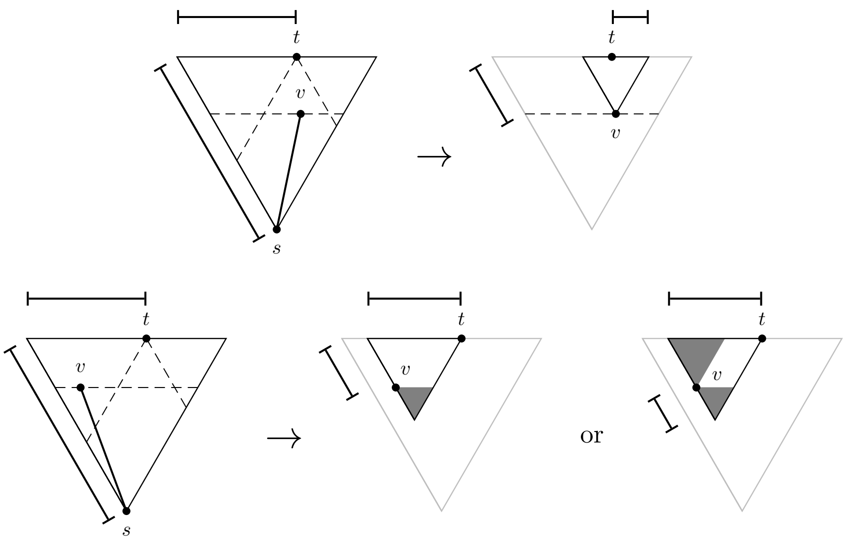


Figure 10: Routing in state A.

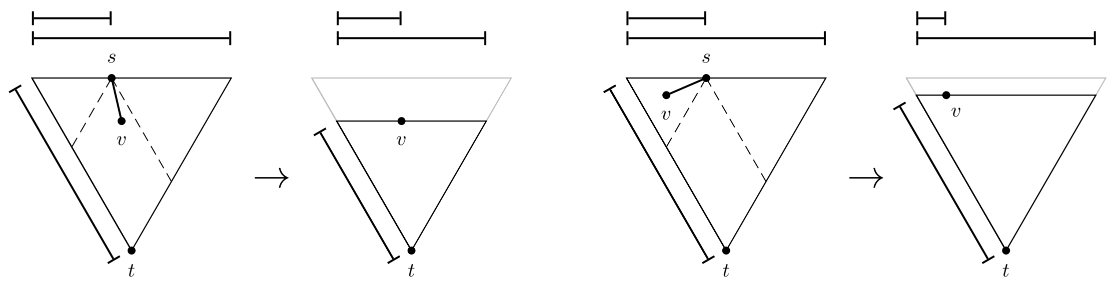


Figure 11: Routing in state B.

lead to states B, C, and D. There are two subcases, depending on whether edges (s, v) with $v \in X_0$ exist or not. These subcases are illustrated in Figure 11 for the situations leading to state B.

In the first subcase, the reduction of the vertical part of ϕ is at least as large as $|sv|$, by Fact 4.1. The horizontal part of ϕ can only decrease. In the second subcase, the reduction of the horizontal part of ϕ is at least as large as $|sv|$, by Fact 4.1. The vertical part of ϕ can only decrease. In both subcases, the statement is proven.

If the step instead leads to state C or D (not illustrated), the value of ϕ after the step can only be smaller than above, so the statement also holds there.

State C. A routing step starting in state C cannot lead to state A, as the step stays within T_{ts}. We first prove that it cannot lead to state B, either. There are two subcases, depending on whether edges (s, v) with $v \in X_0$ exist or not. For the subcase where edges (s, v) with $v \in X_0$ do exist, the situation at the start of the step is illustrated left of the arrow in the left half of Figure 12. By the construction of the half-θ_6-graph, the existence of the edge (s, v) implies that the horizontally hatched area must be empty. From this it follows that the vertically hatched area must also be empty: if not, the topmost point in it would have an edge to s, while having larger projected length on the neighboring cone, contradicting the choice of v in the routing algorithm.

For the subcase where edges (s, v) with $v \in X_0$ do not exist, the situation at the start of the step is illustrated left of the arrow in the right half of Figure 12. The horizontally hatched area must be empty by construction of the half-θ_6-graph. From this it follows that the vertically hatched area must also be empty: if not, the topmost point in it would have an edge to s, contradicting that edges (s, v) with $v \in X_0$

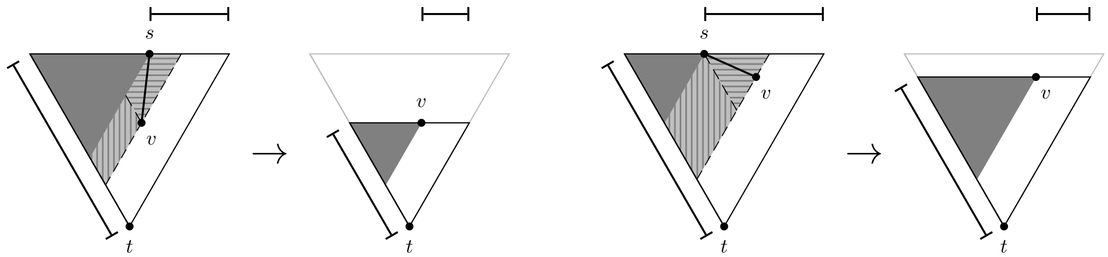


Figure 12: Routing in state C.

do not exist.

Thus, in both subcases the routing step can only lead to state C or D. The situations when ending up in state C are illustrated right of the arrows of Figure 12. In the left half of Figure 12, the reduction of the vertical part of ϕ is at least as large as $|sv|$ (by Fact 4.1) and the horizontal part can only decrease. In the right half of Figure 12, the reduction of the horizontal part of ϕ is at least as large as $|sv|$ (by Fact 4.1) and the vertical part can only decrease. In both situations, the statement is proven. If the step instead leads to state D (not illustrated), the value of ϕ after the step can only be smaller than when leading to state C, so the statement also holds there.

State D. A routing step starting in state D (not illustrated) has exactly the same analysis as the first subcase of a routing step starting in state C.

Thus, the statement is proven in all cases, which completes the proof of Theorem 4.1.

5 Concluding Remarks

The full-θ_6-graph, introduced by Keil and Gutwin [11], is similar to the half-θ_6-graph except that all 6 cones are positive cones. Thus, the full-θ_6-graph is the union of two copies of the half-θ_6-graph, where one half-θ_6-graph is rotated by $\pi/3$ radians. The half-θ_6-graph and the full-θ_6-graph both have a spanning ratio of 2, with simple lower bound examples showing that it is tight for both graphs. This is surprising since the full-θ_6-graph can have double the number of edges of the half-θ_6-graph. Our results show that although the spanning ratios are the same, the competitive routing ratios differ. Since the full-θ_6-graph has no negative cones, it has a competitive routing ratio of 2, which is equal to the spanning ratio and therefore is optimal.

Note that since the full-θ_6-graph consists of two rotated copies of the half-θ_6-graph, one question that comes to mind is what is the best spanning ratio if one is to construct a graph as two rotated copies of the half-θ_6-graph? Can one do better than a spanning ratio of 2? Consider the following construction. Build two half-θ_6-graphs as described in Section 2, but rotate each cone of the second graph by $\pi/6$ radians. By Theorem 3.1, the spanning ratio for each of these graphs is $\sqrt{3}\cos\alpha + \sin\alpha$, where α is the angle between the line connecting the vertices in question and the closest bisector. Since this function is increasing, the spanning ratio is defined by the maximum possible angle to the closest bisector, which in this construction is only $\pi/12$ radians, giving a spanning ratio of roughly 1.932.

By using more copies, we improve the spanning ratio even further: if each is rotated by $\pi/(3k)$ radians, we get a spanning ratio of $\sqrt{3}\cos\pi/(6k) + \sin\pi/(6k)$. This is better than the known upper bounds for the full θ_{3k}-graph [11] for $k \leq 4$ and better than the Yao$_{3k}$-graph [17] for $k \leq 5$.

References

[1] P. Angelini, F. Frati, and L. Grilli. An algorithm to construct greedy drawings of triangulations. *J. Graph Algorithms Appl.*, 14(1):19–51, 2010.

[2] N. Bonichon, C. Gavoille, N. Hanusse, and D. Ilcinkas. Connections between theta-graphs, Delaunay triangulations, and orthogonal surfaces. In *WG*, pages 266–278, 2010.

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

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

[5] P. Chew. There are planar graphs almost as good as the complete graph. *J. Comput. Syst. Sci.*, 39(2):205–219, 1989.

[6] R. Dhandapani. Greedy drawings of triangulations. *Discrete & Computational Geometry*, 43(2):375–392, 2010.

[7] M. B. Dillencourt. Realizability of Delaunay triangulations. *Inf. Process. Lett.*, 33(6):283–287, 1990.

[8] M. T. Goodrich and D. Strash. Succinct greedy

geometric routing in $\mathbb{R}^2$. In *ISAAC*, pages 781–791, 2009.

[9] X. He and H. Zhang. Schnyder greedy routing algorithm. In *TAMC*, pages 271–283, 2010.

[10] X. He and H. Zhang. On succinct convex greedy drawing of 3-connected plane graphs. In *SODA*, pages 1477–1486, 2011.

[11] J. M. Keil and C. A. Gutwin. Classes of graphs which approximate the complete Euclidean graph. *Discrete & Computational Geometry*, 7:13–28, 1992.

[12] T. Leighton and A. Moitra. Some results on greedy embeddings in metric spaces. *Discrete & Computational Geometry*, 44(3):686–705, 2010.

[13] S. Mishra, I. Woungang, and S. Chandra, editors. *Guide to Wireless Sensor Networks*. Springer, 2009.

[14] C. H. Papadimitriou and D. Ratajczak. On a conjecture related to geometric routing. *Theor. Comput. Sci.*, 344(1):3–14, 2005.

[15] H. Räcke. Survey on oblivious routing strategies. In *CiE*, pages 419–429, 2009.

[16] W. Schnyder. Embedding planar graphs on the grid. In *SODA*, pages 138–148, 1990.

[17] A. C.-C. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM J. Comput.*, 11(4):721–736, 1982.