On the Spanning and Routing Ratios of the Directed Θ_6-Graph

Hugo A. Akitaya[1], Ahmad Biniaz[2], and Prosenjit Bose[3](✉)

[1] Department of Computer Science, University of Massachusetts Lowell, Lowell, USA
Hugo_Akitaya@uml.edu

[2] School of Computer Science, University of Windsor, Windsor, Canada
abiniaz@uwindsor.ca

[3] School of Computer Science, Carleton University, Ottawa, Canada
jit@scs.carleton.ca

Abstract. The family of Θ_k-graphs is an important class of sparse geometric spanners with a small spanning ratio. Although they are a well-studied class of geometric graphs, no bound is known on the spanning and routing ratios of the directed Θ_6-graph. We show that the directed Θ_6-graph of a point set P, denoted $\overrightarrow{\Theta}_6(P)$, is a 7-spanner and there exist point sets where the spanning ratio is at least $4-\varepsilon$, for any $\varepsilon > 0$. It is known that the standard greedy Θ-routing algorithm may have an unbounded routing ratio on $\overrightarrow{\Theta}_6(P)$. We design a simple, online, local, memoryless routing algorithm on $\overrightarrow{\Theta}_6(P)$ whose routing ratio is at most 14 and show that no algorithm can have a routing ratio better than $6-\varepsilon$.

Keywords: Spanners · Theta graphs · Routing algorithms

1 Introduction

A geometric graph $G = (V, E)$ is a graph whose vertex set V is a set of points in the plane and whose edge set E is a set of segments joining vertices. Typically, the edges are weighted with the Euclidean distance between their endpoints and we refer to such graphs as Euclidean geometric graphs. A spanning subgraph H of a weighted graph G is a t-spanner of G provided that the weight of the shortest path in H between any pair of vertices is at most t times the weight of the shortest path in G. The smallest constant t for which H is a t-spanner of G is known as the **spanning ratio** or the **stretch factor** of H.

There is a vast literature outlining different algorithms for constructing various geometric $(1+\varepsilon)$-spanners of the complete Euclidean geometric graph (see [13,18] for a survey of the field). One can view a t-spanner H of a graph G as an approximation of G. From this perspective, there are many parameters that can be used to measure how good the approximation is. The obvious parameter is the spanning ratio, however, many other parameters have been studied in addition

Research supported in part by NSERC.

© Springer Nature Switzerland AG 2021
A. Lubiw and M. Salavatipour (Eds.): WADS 2021, LNCS 12808, pp. 1–14, 2021.
https://doi.org/10.1007/978-3-030-83508-8_1

to the spanning ratio such as the size, the weight, the maximum degree, connectivity, diameter to name a few. The study of spanners is a rich subfield and many of the challenges stem from the fact that these parameters are sometimes opposed to each other. For example, a spanner with high connectivity cannot have low maximum degree. As such, many different construction methods have been proposed which outline trade-offs between the various parameters.

A geometric graph H being a $(1+\varepsilon)$-spanner of the complete Euclidean geometric graph certifies the existence of a short path in H between every pair of vertices. Finding such a short path is as fundamental a problem as constructing a good spanner. Typically, most path-planning or routing algorithms are assumed to have access to the whole graph when computing a short path [12,15,17]. However, in many settings, the routing must be performed in an **online** manner. This presents different challenges since the whole graph is not available to the algorithm but the routing algorithm must explore the graph as it attempts to find a path. By providing the routing algorithm with a sufficient amount of memory or a large enough stream of random bits, one can successfully route online using a random walk [14,19] or Depth-First Search [15]. The situation is more challenging if the online routing algorithm is to be **memoryless** and **local**, i.e. the only information available to the algorithm, prior to deciding which edge to follow out of the current vertex, consists of the coordinates of the current vertex, the coordinates of the vertices adjacent to the current vertex and the coordinates of the destination vertex. The routing ratio of such an algorithm is analogous to the spanning ratio except that the ratio is with the weight of the path followed by the routing algorithm as opposed to the shortest path in the spanner. Thus, the routing ratio, by definition, is an upper bound on the spanning ratio. The main difficulty in designing these types of algorithms is that deterministic routing algorithms that are memoryless and local often fail by cycling [9].

Introduced independently by Clarkson [11] and Keil and Gutwin [16], Θ_k-graphs are an important class of $(1+\varepsilon)$-spanners of the complete Euclidean geometric graph for $\varepsilon > 0$. Θ_k-graphs have bounded spanning ratio [2,6,7,11,16,20] for all $k > 3$ and unbounded spanning ratio [1] for $k = 2, 3$. Informally, a Θ_k-graph is constructed in the following way: the plane around each vertex v is partitioned into k cones with apex v and cone angle $\theta = 2\pi/k$. In each cone, v is joined to the point whose projection on the bisector of the cone is closest to v. Although this naturally gives rise to a directed graph (where the previously described edges are directed away from v), much of the literature on Θ_k-graphs has focused on the underlying undirected graph. For example, the tightest upper and lower bounds on the spanning ratio for Θ_k-graphs are proven on the underlying undirected graphs (see [7] for a survey). Given a planar point set P, to avoid any confusion, we will denote the directed version of the Θ_k-graph as $\overrightarrow{\Theta}_k(P)$ and the underlying undirected graph as $\Theta_k(P)$. While it is harder to obtain routing algorithms for $\overrightarrow{\Theta}_k(P)$ because of the extra constraint imposed by the directed edges, $\overrightarrow{\Theta}_k(P)$ has the advantage of maximum out-degree bounded by k, which allows for local routing algorithms in ad-hoc networks where each node's storage is limited by a constant. In contrast, the $\Theta_k(P)$ graph can have maximum degree linear in $|P|$.

Note that the definition of $\overrightarrow{\Theta}_k$-graphs gives rise to a simple, online, local routing algorithm often referred to as **greedy** Θ**-routing**: when searching for a path from a vertex s to a vertex d, follow the edge from s in the cone that contains d. Repeat this procedure until the destination is reached. At each step, the only information used to make the routing decision is the location of the destination and the edge out of the current vertex that contains the destination. Thus, greedy Θ-routing is online, local and memoryless. Ruppert and Seidel [20] showed that greedy Θ-routing has a routing ratio of $1/(1-2\sin(\pi/k))$ for $k \geq 7$. For $3 < k < 7$, it was shown that the routing ratio is unbounded [5]. Intuitively, it seems that the routing ratio should be worse than the spanning ratio for all values of k, since an online routing algorithm must explore the graph while searching for a short path. Indeed, this is true for all values of $k \geq 7$, except when $k \equiv 0 \bmod 4$, in which case the upper bound on the routing ratio and the spanning ratio is $(\cos(\pi/k) + \sin(\pi/k))/(\cos(\pi/k) - \sin(\pi/k))$.

Recently, it was shown that $\overrightarrow{\Theta}_4$ has bounded routing ratio [6]. Although this is not claimed explicitly by the authors, a careful analysis of their proof shows that their result actually carries over to the directed setting. It was shown that the Half-Θ_6 graph – a subgraph of the Θ_6 graph whose edges only consist of those defined in even cones – has an optimal spanning ratio of 2 and an optimal routing ratio of $5/\sqrt{3}$ [4,8,10]. This is the first result we are aware of that shows a strict separation between the optimal spanning and routing ratios. However, the routing algorithm is defined on the undirected graph and the algorithm explicitly follows edges in the *wrong* direction. No tight bounds are known for the spanning ratios in $\overrightarrow{\Theta}_k$, except when $k \geq 7$ and $k \equiv 2 \bmod 4$, for which it is known that the spanning ratio is $1 + 2\sin(\pi/k)$, and this bound is tight in the worst case [7]. For a comprehensive overview of the current best known spanning and routing ratios for Θ_k, for $k \geq 7$, we refer the reader to [7] (Table 1).

1.1 Our Contributions

We focus on fundamental questions related to $\overrightarrow{\Theta}_6(P)$. All that is known is that it is strongly-connected [5]. We show that $\overrightarrow{\Theta}_6(P)$ is a 7-spanner (Sect. 3). Although

Table 1. Partial summary of the best known upper bounds for spanning and routing ratios. Bold numbers indicate results from this paper. Results followed by * have a known matching lower bound. See Bose et al. [7] for other results on general Θ_k when $k \bmod 4 \neq 0$, or the full version of the paper for a more complete table including lower bounds.

	$\Theta_4/\overrightarrow{\Theta}_4$	$\Theta_6/\overrightarrow{\Theta}_6$	$\Theta_{4k}/\overrightarrow{\Theta}_{4k}$, $k > 1$
Spanning	17/17 [6]	2* [4]/**7**	$\frac{cos(\theta/2)+\sin(\theta/2)}{cos(\theta/2)-\sin(\theta/2)}$ [7]/$\frac{cos(\theta/2)+\sin(\theta/2)}{cos(\theta/2)-\sin(\theta/2)}$ [7]
Routing	17/17 [6]	$\frac{5}{\sqrt{3}}$* [8]/**14**	$\frac{cos(\theta/2)+\sin(\theta/2)}{cos(\theta/2)-\sin(\theta/2)}$ [7]/$\frac{cos(\theta/2)+\sin(\theta/2)}{cos(\theta/2)-\sin(\theta/2)}$ [7]

our proof is constructive, it cannot be converted into a local routing algorithm since the construction of the routing path between given points requires knowledge of the whole graph. However, we are able to successfully design an online, local, memoryless routing algorithm on $\overrightarrow{\Theta}_6(P)$ whose routing ratio is at most 14 (Sect. 4). Our algorithm is simple but different from greedy Θ-routing, and, the analysis of the routing ratio is non-trivial since our algorithm makes some decisions that are counter-intuitive. For example, even if there exists a greedy edge whose endpoint is close to the destination, under certain circumstances, our algorithm chooses to go to a vertex that is farther away in a cone that does not contain the destination. In essence, greed is not always good. We complement these upper bounds with the following lower bounds in the full version of this paper. We note that our lower bounds are proven on the strongest model (for any online local algorithm even with arbitrary memory) of online routing and our upper bound is designed on the weakest model (online, local, and memoryless). We summarize our main results below.

Theorem 1. *The spanning ratio of $\overrightarrow{\Theta}_6$ is at most 7 and there exists a point set P such that the spanning ratio of $\overrightarrow{\Theta}_6(P)$ is at least $4 - \varepsilon$ for any $\varepsilon > 0$.*

Theorem 2. *There exists an online, local, memoryless routing algorithm whose routing ratio on $\overrightarrow{\Theta}_6$ is at most 14. For any $\varepsilon > 0$ and any local routing algorithm A in $\overrightarrow{\Theta}_6(P)$, the routing ratio of A is at least $6 - \varepsilon$.*

2 Preliminaries

In this section, we outline some notation and definitions. Given two points a, b in the plane, $\|ab\|$ refers to their Euclidean distance. A convex polygon C is **regular** if all its edges are of the same length. By $\|C\|$, we refer to the side length of C. The boundary of C is denoted as $bd(C)$ and the interior of C is denoted as $int(C)$. We call a triangle (resp. hexagon) **aligned** if each of its edges is parallel to a line of slope $\sqrt{3}$, slope 0 or slope $-\sqrt{3}$. Given two distinct points u, v in the plane, the canonical triangle of u with respect to v, denoted $\triangledown_u^v$ is the regular aligned triangle where u is one of the vertices and v is on the edge of the triangle opposite u. Note that $\triangledown_u^v$ is congruent to $\triangledown_v^u$. Let $\hexagon_u^v$ be the regular aligned hexagon centered at u that has v on its boundary. The lines through u having slope $\sqrt{3}$, slope 0 and slope $-\sqrt{3}$, respectively, partition the hexagon into 6 regular aligned triangles. Label these triangles $\triangle_{uv}^0, \ldots, \triangle_{uv}^5$ in counter-clockwise order with the convention that $\triangle_{uv}^0$ is the triangle below u with a horizontal base. When referring to these triangles or sets related to these triangles, indices are manipulated modulo 6. When it is clear from the context, to make notation a little less cumbersome, we drop the subscript uv (see Fig. 1). Note that $\triangle^i$ for the $i \in \{0, \ldots, 5\}$ that has v on its base is identical to $\triangledown_u^v$. This implies that $\left\|\triangle^i\right\| = \|\triangledown_u^v\| = \left\|\hexagon_u^v\right\|$. Finally, we note that a regular aligned hexagon defines a distance metric. Given two points u, v in the plane, the hexagonal distance between u and v, $d_{\hexagon}(u, v) = \left\|\hexagon_u^v\right\| = \left\|\hexagon_v^u\right\|$.

A directed edge (u, v) in a graph is an ordered pair and represents an edge directed from vertex u to vertex v. We refer to u as the **tail** of the edge and v as the **head** of the edge. To simplify the discussion and avoid situations where points are bordering on two cones, we make the following general position assumption on a point set P: no two points lie on a line of slope $\sqrt{3}$, slope 0 or slope $-\sqrt{3}$. Note that a slight rotation of the point set removes this, as such, this assumption does not take away from the generality of our results. Given a set of points P in the plane, the directed Θ_6-graph whose vertex set is P is denoted $\overrightarrow{\Theta_6}(P)$. A directed edge (a, b) exists in $\overrightarrow{\Theta_6}(P)$ provided that $\bigtriangledown_a^b$ does not contain any point of $P \setminus \{a, b\}$. An equivalent way to construct $\overrightarrow{\Theta_6}(P)$ is the following. For each $u \in P$, the lines through u with slopes $-\sqrt{3}, 0, \sqrt{3}$ partition the plane into 6 cones. We label these cones $\wedge_u^i$, $i \in \{0, \ldots, 5\}$ counterclockwise with $\wedge_u^0$ being the cone directly below u. For each cone $\wedge_u^i$, add edge (u, v) if $v \in \wedge_u^i$ is the closest point to u in the $d_{\hexagon}$ metric. This makes explicit the fact that the maximum out-degree of $\overrightarrow{\Theta_6}(P)$ is 6.

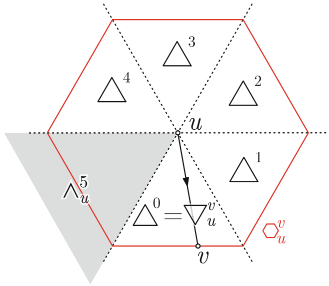

Fig. 1. Illustrations of our definitions.

2.1 The Routing Model

Given a graph $G = (V, E)$, with vertex set V and edge set E, an online, ℓ-local routing algorithm can be expressed as a function $f : V \times V \times H \times M \to V \times M$, where $M = \{0, 1\}^*$. The parameters of $f(u, d, G_\ell(u), m)$ are: u the current vertex, d the destination vertex, $G_\ell(u)$ the subgraph of G that consists of all paths rooted at u with length at most ℓ and m is a bit-string representing the memory. An invocation of the routing function $f(u, d, G_\ell(u), m)$ updates m and returns $v \in V$ such that the edge (u, v) should be followed out of u to reach destination d. This is the strongest model of online routing where the algorithm has infinite memory and is aware of the graph induced on the ℓ-neighborhood prior to making a routing decision. With this model, one can perform Depth-First Search on G. The algorithm is considered 1-local or **local** if $\ell = 1$. It is considered memoryless if $M = \emptyset$, that is, the algorithm has no memory or knowledge of where it started or where it has been. The weakest model is online, local and memoryless. For example, one cannot even perform Depth-First Search in this model. Although quite restrictive, our routing algorithm falls within the weakest model.

3 Upper Bound on the Spanning Ratio

In this section, we show that $\overrightarrow{\Theta}_6(P)$ is a 7-spanner. Given a destination vertex $d \in \overrightarrow{\Theta_6}(P)$, we define the **greedy edge** of vertex v with respect to d to be the outgoing edge of v in $\bigtriangledown_v^d$. Recall that the routing strategy of repeatedly following the greedy edge at every step until the destination is reached is called **greedy routing** or Θ-routing. The path found by the greedy routing algorithm is called the **greedy path**. Thus, the greedy path from s to d, denoted $\pi(s, d)$, is the path in $\overrightarrow{\Theta_6}(P)$ starting at s and where at every step, the greedy edge with respect to d is selected, until the destination d is reached.

Given a starting vertex s and a destination vertex d, by construction, we have that the canonical triangle, $\bigtriangledown_s^d$, is contained in the hexagon $\langle d \rangle^s$. Let (s, a) be the first greedy edge in $\pi(s, d)$. Then, since a is in $\bigtriangledown_s^d$ we have that $d_{\hexagon}(a, d) < d_{\hexagon}(s, d)$. The inequality is strict since by our general position assumption a is contained in $int(\bigtriangledown_s^d)$, or $a = d$. Therefore, at every step of the greedy routing algorithm, the hexagonal distance to the destination decreases. Since there are a finite number of points in P and the fact that the hexagonal distance to the destination is strictly decreasing at every step, the greedy algorithm terminates at d. We summarize this in the following lemma.

Lemma 1. *Given any pair of points $s, d \in P$, there always exists a greedy path from s to d in $\overrightarrow{\Theta_6}(P)$. Furthermore, let x be a vertex in $\pi(s, d)$ different from s and d. Then the following hold:*

- $\langle d \rangle^x$ *is contained in* $(\langle d \rangle^s)$
- $d_{\hexagon}(x, d) < d_{\hexagon}(s, d)$
- $\pi(x, d)$ *is contained in* $\langle d \rangle^x$

Although the greedy routing algorithm always reaches its destination, its spanning ratio is not bounded by a constant [5]. The issue is that $\pi(s, d)$, although getting closer to d with respect to the hexagonal distance, can spiral around d many times (see Fig. 2).

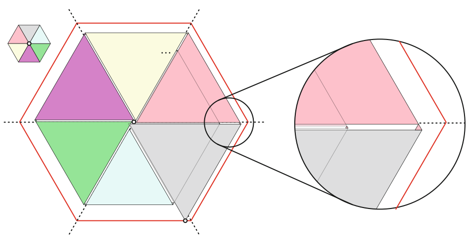

Fig. 2. From [5]. Colored triangles are interior-empty triangles $\bigtriangledown_u^v$ that define an edge (u, v) of $\overrightarrow{\Theta_6}(P)$. Different colors encode different canonical triangles in $\{\triangle_{uv}^0, \ldots, \triangle_{uv}^5\}$. The spanning ratio of the greedy path from the perimeter to the center of the red hexagon is not bounded by a constant. (Color figure online)

However, if there happens to be an edge from d to s, i.e. $(d, s) \in \overrightarrow{\Theta_6}(P)$, then $\pi(s, d)$ can no longer spiral around d since $\bigtriangledown_d^s$ is empty of points of P and acts as a barrier, as we shall prove in Lemma 2. This prevents the path from cutting across $\bigtriangledown_d^s$. We then prove that if (d, s) is an edge of $\overrightarrow{\Theta_6}(P)$ then the spanning ratio of $\pi(s, d)$ is at most $6 \|\bigtriangledown_d^s\|$ (Corollary 1).

For $i \in \{0, \ldots, 5\}$, let $T_i = \{(a, b) \in \pi(s, d) \mid a \in \triangle_{ds}^i\}$. T_i is the set of all edges of $\pi(s, d)$ whose tail is in $\triangle^i$. Define the weight of T_i, denoted $\|T_i\|$, to be $\sum_{(a,b) \in T_i} \left\|\bigtriangledown_a^b\right\|$. For ease of reference, label the sequence of vertices in $\pi(s, d)$ as $s = u_0, \ldots, u_k = d$ where k is the number of edges.

Lemma 2. *If (u_a, u_{a+1}) is an edge of $\pi(s,d)$ in T_i for $a \in \{0, \ldots, k-1\}$ and $i \in \{0, \ldots, 5\}$, then u_{a+1} can only be in one of $\triangle_{ds}^{i-1}, \triangle_{ds}^{i}$ or $\triangle_{ds}^{i+1}$.*

Proof. Without loss of generality, assume that u_a is in $\triangle_{ds}^0$. Let $h^+(d)$ be the half-plane above the horizontal line through d. Since the edge of $\triangledown_{u_a}^d$ that contains d is horizontal and the interior of the triangle lies below the horizontal line through d, we have that $int(\triangledown_{u_a}^d) \cap h^+(d) = \emptyset$. Therefore, u_{a+1} cannot be in $\triangle^2, \triangle^3$ or $\triangle^4$, since the interiors of all those triangles are in $h^+(d)$. The lemma follows. □

Note that Lemma 2 immediately implies that the greedy path cannot spiral around d since that would require $\pi(s,d)$ to contain a point of P in $int(\triangledown_d^s)$, contradicting the existence of edge (d,s). This lets us bound the length of $\pi(s,d)$.

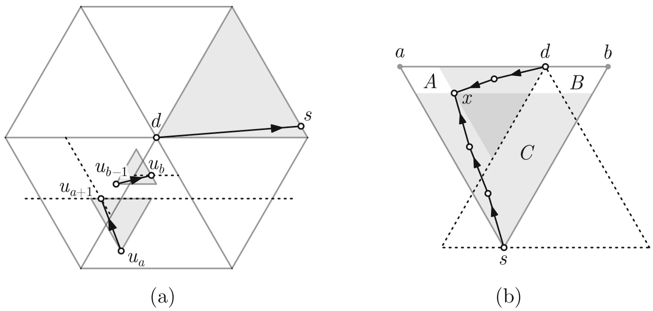

Fig. 3. (a) Illustration of Lemma 3. (b) Illustrations of Theorem 3.

Lemma 3. *Assume (d,s) is an edge of $\overrightarrow{\Theta_6}(P)$ and let u_a be a vertex of $\pi(s,d)$ in $\triangle_{ds}^i$. Let u_b be the next vertex in $\pi(s,d)$ after u_a that appears in $\triangle_{ds}^i$, i.e. $b > a$. Then, $int(\triangledown_{u_a}^{u_{a+1}}) \cap int(\triangledown_d^{u_b}) = \emptyset$.*

Proof. Without loss of generality, assume that u_a is in $\triangle_{ds}^0$. We have two cases: either $u_b = u_{a+1}$ or $u_b \neq u_{a+1}$. We begin with the former. If $u_b = u_{a+1}$ then the lemma holds trivially since $\triangledown_{u_a}^{u_{a+1}}$ and $\triangledown_d^{u_{a+1}}$ are separated by a horizontal line.

We now consider the case where $u_b \neq u_{a+1}$, i.e. $b > a+1$ (Fig. 3(a)). By Lemma 2 and u_b's definition, u_{a+1} must either be in $\triangle^1$ or $\triangle^5$. Without loss of generality, assume that u_{a+1} is in $\triangle^5$. Consider the edge (u_{b-1}, u_b) of $\pi(s,d)$. By Lemma 2, u_{b-1} must be in $\triangle^5$ since, by the existence of (d,s), the path cannot spiral around d and enter $\triangle^0$ from $\triangle^1$. By Lemma 1, u_{b-1} must be contained in $\triangledown_d^{u_{a+1}}$. Moreover, since (u_a, u_{a+1}) is an edge of the path, we have that $\triangledown_{u_a}^{u_{a+1}}$ is empty, which means that u_{b-1} lies above the horizontal line through u_{a+1}. This implies that u_b also lies above the horizontal line through u_{a+1} since the canonical triangle $\triangledown_{u_{b-1}}^{u_b}$ has a horizontal edge and lies above the horizontal line through u_{b-1}. Therefore, $int(\triangledown_{u_a}^{u_{a+1}}) \cap int(\triangledown_d^{u_b}) = \emptyset$. □

Lemma 4. *If $(d,s) \in \overrightarrow{\Theta_6}(P)$, then $\|T_i\| \leq \|\triangledown_d^s\|$, for $i \in \{0, \ldots, 5\}$.*

Proof. We show the bound for T_0. Let $(a, b) \in T_0$. Let a' (resp. b') be the intersection of a horizontal line through a (resp. b) with the left side of $\triangle_{ds}^0$. Since $\triangledown_a^b$ is equilateral, $\|a'b'\| \geq \|ab\|$. If (a_1, b_1) and (a_2, b_2) are two edges in T_0, by Lemma 3, $a_1'b_1'$ and $a_2'b_2'$ do not overlap. Therefore, $\|T_0\|$ is at most $\|\triangledown_d^s\|$. □

We are now able to bound the length of $\pi(s, d)$ when $(d, s) \in \overrightarrow{\Theta_6}(P)$. As each edge of $\pi(s, d)$ appears in only one T_i, the bound follows from Lemma 4.

Corollary 1. *If $(d, s) \in \overrightarrow{\Theta_6}(P)$, then $\|\pi(s, d)\| \leq 6 \|\triangledown_d^s\| = 6 \left\|\hexagon_d^{s}\right\|$.*

Corollary 1 implies that $\overrightarrow{\Theta_6}(P)$'s spanning ratio is upper bounded by $12\sqrt{3}$. This follows from the fact that $\Theta_6(P)$'s spanning ratio is 2 and for each edge e in $\Theta_6(P)$ there is a directed path of length at most $6\sqrt{3}\,\|e\|$ from one endpoint of e to the other in $\overrightarrow{\Theta_6}(P)$ (the $\sqrt{3}$ term comes from the hexagonal distance metric).

A more careful analysis lets us prove a better spanning ratio. In order to do this, we uncover a structural property of greedy paths in $\overrightarrow{\Theta_6}(P)$. We note that a weaker version of this claim is proven by Bonichon et al. [3] (proof of Theorem 1). Thus, we omit the proof here which is given in the full version.

Theorem 3. *Between any pair of points $s, d \in P$, there exists an $x \in P$ in $\triangledown_s^d$ such that the following hold (note that if the interior of $\triangledown_s^d$ is empty then $x = d$):*

1. *$\pi(s, x)$ and $\pi(d, x)$ are both in $\triangledown_s^d$,*
2. *$\|\pi(s, x)\| \leq \|\triangledown_s^x\|$,*
3. *$\|\pi(d, x)\| \leq \|\triangledown_d^x\|$.*

Proof Sketch. See Fig. 3(b) for an example. We prove the claim by induction on the rank of pairs of points (s, d) as sorted order by $\|\triangledown_s^d\|$. The induction step builds the required paths using the greedy edge from s in d's direction and the path obtained by applying a stronger induction hypothesis. □

We now prove the main result of this section.

Theorem 4. *Between any pair of points $s, d \in P$, there exists a directed path $\delta(s, d)$ in $\overrightarrow{\Theta_6}(P)$ such that the length of $\delta(s, d)$ is at most $7\,\|sd\|$.*

Proof. Given a greedy path $\pi(u, v)$, the **reverse** path, denoted $\rho(v, u)$, is a directed path from v to u where every edge (x, y) in $\pi(u, v)$ is replaced with the greedy path $\pi(y, x)$. By Theorem 3, between any pair of points $s, d \in P$, there exists an $x \in \triangledown_s^d$ such that $\pi(s, x)$ and $\pi(d, x)$ are both in $\triangledown_s^d$, $\|\pi(s, x)\| \leq \|\triangledown_s^x\| \leq \left\|\triangledown_s^d\right\|$, and $\|\pi(d, x)\| \leq \|\triangledown_d^x\|$. Let $\delta(s, d)$ be the path resulting from the concatenation of $\pi(s, x)$ and $\rho(x, d)$. By construction, $\delta(s, d)$ is a directed path from s to d. Let A be one of the two triangles obtained from $\triangledown_s^d \setminus \triangledown_d^s$. Let a, b and s be the vertices of $\triangledown_s^d$ with a being incident to A. Without loss of generality, assume the orientation shown in Fig. 3(b) and that $x \in A$. Consider the triangle defined by s, a, d and let γ be the angle at s. By elementary trigonometry, we have that the spanning ratio is $\|\delta(s, d)\| \,/\, \|sd\| \leq (\sin(2\pi/3 - \gamma) + 6 \sin\gamma) / \sin(\pi/3) \leq 7$, since the maximum is attained when $\gamma = \pi/3$. □

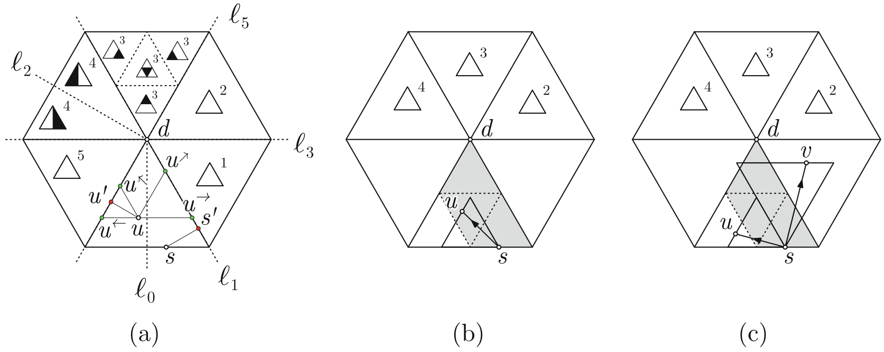

Fig. 4. (a) Examples of some of the notation used. (b) Example in which Algorithm 1 takes the non-greedy (s, u). (c) Example in which Algorithm 1 takes the greedy edge (s, v).

Although the proof of the spanning ratio of 7 for $\overrightarrow{\Theta_6}(P)$ is constructive, unfortunately, it does not provide an online routing algorithm. There are 3 main obstacles. First, in the proof, the path is constructed from both ends, where we build a greedy path from s to x and another from d to x. Second, the point x is not easily identifiable locally. And third, when finding the reverse path of an edge (a, b), one needs to know both a and b, which may not be information that is available if we are only aware of outgoing edges.

4 Routing Algorithm and Upper Bound on Routing Ratio

This section provides a routing algorithm in $\overrightarrow{\Theta}_6$. We first describe some notation used in this section. Similar to $\hexagon_u^v$, we denote by $\hexagon_u^v$ the axis aligned hexagon rotated by $\pi/6$ that is centered at u and contains v on its boundary. For an example, see the shaded hexagon in Fig. 5. The following definitions refer to a hexagon $\hexagon_d^s$. Refer to Fig. 4 (a). Let ℓ_0 be the vertical line through d and ℓ_1, ℓ_2, ℓ_3, ℓ_4, and ℓ_5 be the lines through d with slopes $-\sqrt{3}$, $-\frac{1}{\sqrt{3}}$, 0, $\frac{1}{\sqrt{3}}$, and $\sqrt{3}$ respectively. For a point $u \in \triangle_{ds}^0$, we define point u' as the orthogonal projection of u on ℓ_1 or ℓ_5, whichever is closest to u. We also define points $u^{\rightarrow}$, $u^{\nearrow}$, $u^{\nwarrow}$, and $u^{\leftarrow}$ as the intersections between ℓ_1 or ℓ_5 and the rays from u with angles 0, $\pi/3$, $2\pi/3$, and π from the positive x-direction. We define $\blacktriangle^0$ and $\blacktriangle^0$ to be the left and right triangles obtained from partitioning $\triangle_{ds}^0$ with ℓ_0. We also partition $\triangle_{ds}^0$ into four congruent triangles with the line segments through two of the three midpoints of sides of $\triangle_{ds}^0$. Denote by $\blacktriangle^0$, $\blacktriangle^0$, $\blacktriangle^0$, $\blacktriangle^0$ the top, left, right, and middle triangles respectively. We apply the appropriate rotations to obtain the analogous definitions for $\triangle_{ds}^i$, $i \in \{1, \ldots, 5\}$. For example, $\blacktriangle^3$ in Fig. 4 (a) is the bottom triangle in $\triangle_{ds}^3$.

Algorithm 1: DirectedRoute(s, d, $N(s)$)

s is the current vertex, d is the destination and $N(s)$ is the 1-neighborhood of s.
Assume that s is in $\triangle^0_{ds}$ otherwise apply the appropriate rotations and/or reflection;
Let (s, v) be the edge from s in $\wedge^3_s$;
// Greedy Edge
Let (s, u) be the edge from s in $\wedge^4_s$;
// Non-greedy Edge if it exists
if u **exists and** $u \in \blacktriangle^0_{ds} \cup \blacktriangle^0_{ds} \cup \triangle^0_{ds}$ **then**
  Return u; // Take Non-greedy Edge
else
  Return v; // Take Greedy Edge
end

We define the **potential** of a point $p \in P$ as $\Phi(p) = \frac{\sqrt{3}}{2} \left\| \hexagon_d^{\,p} \right\|$. Let $(u_0 = s, u_1, \ldots, u_k = d)$ be the sequence of vertices visited by Algorithm 1. The following lemma shows that the potential decreases with each step of the algorithm. We provide its proof in the full version.

Lemma 5. *Let (u_a, u_{a+1}) be an edge taken by Algorithm 1. Then $\Phi(u_{a+1}) < \Phi(u_a)$, and $\left\| \hexagon_d^{\,u_{a+1}} \right\| < \left\| \hexagon_d^{\,u_a} \right\|$*

Since the potential of a point is a function of its position, Lemma 5 implies that no point is visited twice and the destination is always reached. We now bound the routing ratio. We apply a charging scheme for each edge taken by the algorithm based on its type. We classify the edges (u_a, u_{a+1}) taken by the algorithm as follows. If u_{a+1} is in $\blacktriangle^1_{ds}$ we call the edge a **long step**. Otherwise, we call the edge a **short step**.

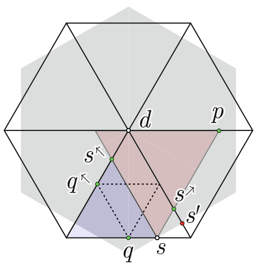

Fig. 5. The shaded hexagon $\hexagon_d^{\,s}$ contains points whose potentials are the same or lower than s. The red triangle is $\triangledown^d_s$ and the blue triangle is $\triangledown^{q^{\nwarrow}}_s$. (Color figure online)

Informal Overview of the Charging Scheme. Each step (u_a, u_{a+1}) will be associated with a decrease in potential $\Phi(u_a) - \Phi(u_{a+1})$ quantifying how much closer the current point u_{a+1} is to d than the previous point u_a. We show in Lemma 6 that for short steps, the decrease in potential is enough to pay for the size of the step $\|u_a u_{a+1}\|$. For long steps, the potential might decrease by an arbitrarily small amount. So in addition to the decrease in potential, we charge the step to a region of the hexagon $\hexagon_d^{\,s}$. The charged regions are axis-aligned trapezoids whose non-parallel edges are on ℓ_i, $i \in \{1, 2, 3\}$. Lemma 7 quantifies how the size of the charged trapezoids relates to the size of the step. We then show that the charged trapezoids have disjoint interiors, i.e., the same region cannot be double charged. This is what allows us to bound the cost of the path.

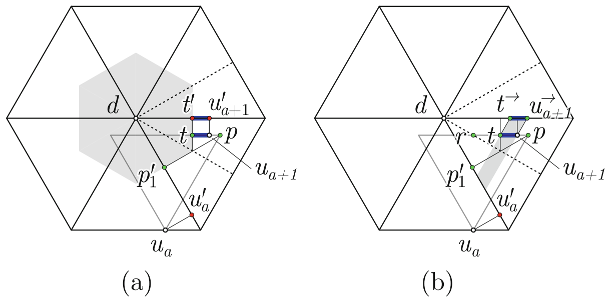

Fig. 6. Charging scheme for long steps.

For two points u and v in the interior of the same cone $\wedge_d^i$, define the **trapezoid** ⏢_u^v as $\bigtriangledown_d^u \setminus \bigtriangledown_d^v$. Note that $\text{⏢}_u^v = \emptyset$ if $\|\bigtriangledown_d^u\| \leq \|\bigtriangledown_d^v\|$, i.e. $\bigtriangledown_d^u \subset \bigtriangledown_d^v$.

Case 1 ((u_a, u_{a+1}) is short) Charge to the decrease in potential $\Phi(u_a) - \Phi(u_{a+1})$.
Case 2 ((u_a, u_{a+1}) is long) Charge to the decrease in potential $\Phi(u_a) - \Phi(u_{a+1})$ *and* to a region $\text{⏢}_{u_{a+1}}^t$ where t is defined as follows. Refer to Fig. 6 (b). Let p be the upper right corner of $\bigtriangledown_{u_a}^{u_{a+1}}$ and r be the intersection between the upper edge of $\bigtriangledown_{u_a}^{u_{a+1}}$ and ℓ_2. We define t to be the midpoint of rp.

Lemmas 6 and 7 formalize the charging scheme. Due to space restrictions, their proofs are in the full version.

Lemma 6. *In Case 1 where (u_a, u_{a+1}) is a short step, the decrease in potential is at least half the size of the step, i.e.,* $\frac{\|u_a u_{a+1}\|}{2} \leq \Phi(u_a) - \Phi(u_{a+1})$.

We define the **length** $\left\|\text{⏢}_{u_{a+1}}^t\right\|$ of a trapezoid $\text{⏢}_{u_{a+1}}^t$ to be the length of one of its non parallel sides. Note that in the context of Case 2, $\left\|\text{⏢}_{u_{a+1}}^t\right\| = \|tu_{a+1}\|$.

Lemma 7. *In Case 2 where (u_a, u_{a+1}) is a long step, the decrease in potential plus the length of the charged region is at least half the size of the step, i.e.,*

$$\frac{\|u_a u_{a+1}\|}{2} \leq \Phi(u_a) - \Phi(u_{a+1}) + \left\|\text{⏢}_{u_{a+1}}^t\right\|.$$

Let $\mathcal{T}$ be the set of all charged trapezoids, and $\|\mathcal{T}\|$ be the sum of lengths of all trapezoids in $\mathcal{T}$. We show a property that allows us to upper bound $\|\mathcal{T}\|$.

Lemma 8. *Let $(u_a, \ldots, u_b)$ be a subsequence of steps taken by Algorithm 1 where all visited points are in the same cone of d. Without loss of generality, let this cone be $\wedge_d^0$, and let $u_a \in \blacktriangle_{du_a}^0$. If u_{b+1} is in $\wedge_d^5$, then $u_{b+1} \in \text{⬡}_d^{\overset{\nwarrow}{u_a}}$. If u_{b+1} is in $\wedge_d^1$, then $u_{b+1} \in \text{⬡}_d^{\overset{\nearrow}{q}}$ where q is the midpoint of the bottom edge of $\triangle_{du_a}^0$.*

Proof. Refer to Fig. 7 (a). If u_{b+1} is not in $\wedge_d^0$, then by Algorithm 1, (u_b, u_{b+1}) is greedy. By Lemma 2, u_{b+1} is either in $\triangle^1_{du_a}$ or $\triangle^5_{du_a}$. Since all edges of the subsequence are short edges, by definition, we have that u_{a+1} is in the pentagon $qq^{\nearrow}du_a^{\nwarrow}u_a$. A simple inductive argument shows that this implies that u_b is in the region $qq^{\nearrow}du_a^{\nwarrow}u_a$. Hence, since (u_b, u_{b+1}) is a greedy edge, we have that $\triangledown^{u_{b+1}}_{u_b} \cap \triangle^1_{du_a} \subset \hexagon{d}^{u_a^{\nwarrow}}$ and $\triangledown^{u_{b+1}}_{u_b} \cap \triangle^5_{du_a} \subset \hexagon{d}^{q^{\nearrow}}$ The lemma follows. $\square$

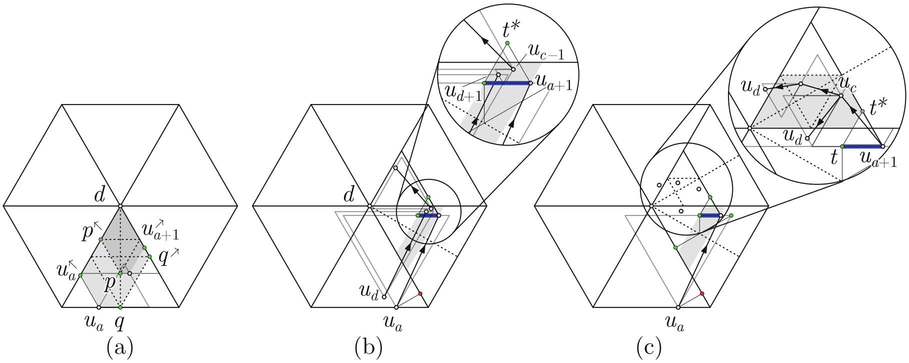

Fig. 7. (a) Illustration of Lemma 8. (b)–(c) Algorithm 1 cannot enter $⏢^{t}_{u_{a+1}}$ once it leaves triangle $tu_{a+1}t^*$.

Lemma 9. *Let $⏢^{t}_{u_{a+1}}$ be the trapezoid charged by a long step (u_a, u_{a+1}), and let $(u_{a+1}, \ldots, u_b)$, $a < b$ be the maximal subpath traversed by Algorithm 1 with $u_b \in ⏢^{t}_{u_{a+1}}$. Then, every step in the subpath is short, and every point in it ($u_i, i \in \{a+1, \ldots, b\}$) is in the equilateral triangle whose bottom edge is tu_{a+1}.*

Proof Sketch. The full proof is in the full version. Refer to Figs. 7 (b)–(c). Let $tu_{a+1}t^*$ be the equilateral triangle whose bottom edge is tu_{a+1}. From u_{a+1}, Algorithm 1 can only take short steps before leaving such triangle to a point u_c. We show that after u_c, the path output by the algorithm can never return to $tu_{a+1}t^*$. Then, no point visited after u_c can be in $⏢^{t}_{u_{a+1}}$. $\square$

Corollary 2. *The trapezoids in $\mathcal{T}$ are pairwise interior disjoint.*

Proof. For contradiction, assume that trapezoids $⏢^{t_a}_{u_{a+1}}$ $⏢^{t_b}_{u_{b+1}}$, charged by long steps (u_a, u_{a+1}) and (u_b, u_{b+1}) with $a < b$, intersect. Then, the larger base of $⏢^{t_b}_{u_{b+1}}$ is between the two bases of $⏢^{t_a}_{u_{a+1}}$. By construction of the trapezoids, u_b is in $⏢^{t_a}_{u_{a+1}}$ contradicting Lemma 9. $\square$

Theorem 5. *The routing ratio of Algorithm 1 is at most* 14.

Proof. By Lemmas 6 and 7, the length of the path returned by Algorithm 1 is at most

$$\sum_{i=1}^{k-1} \|u_i u_{i+1}\| \leq 2(\Phi(s) - \Phi(d) + \|\mathcal{T}\|).$$

By definition, $\Phi(s) < \|sd\|$ and $\Phi(d) = 0$. By Corollary 2, $\|\mathcal{T}\| \leq 6\,\|sd\|$ since the trapezoids can only fill the initial hexagon $⟨d⟩^s$. Then, $\sum_{i=1}^{k-1} \|u_i u_{i+1}\| \leq 14\,\|sd\|$ as required. □

5 Conclusions

We have provided upper and lower bounds for the spanning and routing ratios of $\overrightarrow{\Theta}_6(P)$. There are still gaps between the bounds as they are not matching. We believe that the actual bound is closer to the lower bounds, mainly because in the analysis of the upper bound of both the spanning and routing ratios, we account for the possibility that the path from source to destination goes around intermediate points and/or the destination. However, intuition seems to suggest that this does not actually happen and there is a shorter path that *cuts in* after going half-way around, which is the case in our lower bound constructions. We leave the closing of the gap between the upper and lower bounds as an open problem.

References

1. Aichholzer, O., et al.: Theta-3 is connected. Comput. Geom.: Theory Appl. **47**(9), 910–917 (2014)
2. Barba, L., Bose, P., De Carufel, J.L., van Renssen, A., Verdonschot, S.: On the stretch factor of the theta-4 graph. In: Dehne, F., Solis-Oba, R., Sack, J.R. (eds.) Algorithms and Data Structures. LNCS, vol. 8037, pp. 109–120. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40104-6_10
3. Bonichon, N., Bose, P., Carmi, P., Kostitsyna, I., Lubiw, A., Verdonschot, S.: Gabriel triangulations and angle-monotone graphs: local routing and recognition. In: International Symposium on Graph Drawing and Network Visualization, pp. 519–531. Springer, Heidelberg (2016)
4. Bonichon, N., Gavoille, C., Hanusse, N., Ilcinkas, D.: Connections between theta-graphs, delaunay triangulations, and orthogonal surfaces. In: Thilikos, D.M. (ed.) Graph Theoretic Concepts in Computer Science. LNCS, vol. 6410, pp. 266–278. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16926-7_25
5. Bose, P., De Carufel, J.-L., Devillers, O.: Expected complexity of routing in Θ_6 and half-Θ_6 graphs. arXiv preprint arXiv:1910.14289 (2019)
6. Bose, P., De Carufel, J.-L., Hill, D., Smid, M.H.M.: On the spanning and routing ratio of Theta-four. In: Symposium on Discrete Algorithms, pp. 2361–2370. SIAM (2019)
7. Bose, P., De Carufel, J.-L., Morin, P., van Renssen, A., Verdonschot, S.: Towards tight bounds on theta-graphs: more is not always better. Theoret. Comput. Sci. **616**, 70–93 (2016)

8. Bose, P., Fagerberg, R., van Renssen, A., Verdonschot, S.: Optimal local routing on delaunay triangulations defined by empty equilateral triangles. SIAM J. Comput. **44**(6), 1626–1649 (2015)
9. Bose, P., Morin, P.: Online routing in triangulations. SIAM J. Comput. **33**(4), 937–951 (2004)
10. Chew, P.: There are planar graphs almost as good as the complete graph. J. Comput. Syst. Sci. **39**(2), 205–219 (1989)
11. Clarkson, K.: Approximation algorithms for shortest path motion planning. In: Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC 1987), pp. 56–65 (1987)
12. Dijkstra, E.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959)
13. Eppstein, D.: Spanning trees and spanners. In: Handbook of Computational Geometry, pp. 425–461 (1999)
14. Grimmett, G.: Probability on Graphs: Random Processes on Graphs and Lattices, 2nd edn. Cambridge University Press, Cambridge (2018)
15. Hopcroft, J., Tarjan, R.: Algorithm 447: efficient algorithms for graph manipulation. Commun. ACM **16**(6), 372–378 (1973)
16. Keil, J.M., Gutwin, C.A.: Classes of graphs which approximate the complete Euclidean graph. Discret. Comput. Geom. **7**, 13–28 (1992). https://doi.org/10.1007/BF02187821
17. Lee, C.Y.: An algorithm for path connection and its applications. IRE Trans. Electron. Comput. **EC–10**(3), 346–365 (1961)
18. Narasimhan, G., Smid, M.: Geometric Spanner Networks. Cambridge University Press, Cambridge (2007)
19. Pearson, K.: The problem of the random walk. Nature **72**, 294 (1865)
20. Ruppert, J., Seidel, R.: Approximating the d-dimensional complete Euclidean graph. In: Proceedings of the 3rd Canadian Conference on Computational Geometry (CCCG 1991), pp. 207–210 (1991)