HAL
open science

Improved Routing on the Delaunay Triangulation

Nicolas Bonichon, Prosenjit Bose, Jean-Lou de Carufel, Vincent Despré, Darryl Hill, Michiel Smid

To cite this version:

Nicolas Bonichon, Prosenjit Bose, Jean-Lou de Carufel, Vincent Despré, Darryl Hill, et al.. Improved Routing on the Delaunay Triangulation. ESA 2018 - 26th Annual European Symposium on Algorithms, Aug 2018, Helsinki, Finland. 10.4230/LIPIcs.ESA.2018.22 . hal-01881280

HAL Id: hal-01881280

https://hal.science/hal-01881280v1

Submitted on 12 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Improved Routing on the Delaunay Triangulation

Nicolas Bonichon
Université de Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France

Prosenjit Bose
Carleton University, 1125 Colonel By Dr, Ottawa, ON, Canada

Jean-Lou De Carufel
University of Ottawa, 800 King Edward Ave, Ottawa, ON, Canada

Vincent Despré
Team Gamble, INRIA Nancy, 54600 Villers-lès-Nancy, France

Darryl Hill
Carleton University, 1125 Colonel By Dr, Ottawa, ON, Canada

Michiel Smid
Carleton University, 1125 Colonel By Dr, Ottawa, ON, Canada

Abstract

A geometric graph $G = (P, E)$ is a set of points in the plane and edges between pairs of points, where the weight of an edge is equal to the Euclidean distance between its two endpoints. In local routing we find a path through G from a source vertex s to a destination vertex t, using only knowledge of the current vertex, its incident edges, and the locations of s and t. We present an algorithm for local routing on the Delaunay triangulation, and show that it finds a path between a source vertex s and a target vertex t that is not longer than $3.56|st|$, improving the previous bound of $5.9|st|$.

2012 ACM Subject Classification Mathematics of computing → Paths and connectivity problems, Mathematics of computing → Graph algorithms

Keywords and phrases Delaunay, local routing, geometric, graph

Digital Object Identifier 10.4230/LIPIcs.ESA.2018.22

Funding This work was supported by the National Sciences and Engineering Research Council of Canada (NSERC) and by the French ANR grant ASPAG (ANR-17-CE40-0017)

1 Introduction

A Euclidean geometric graph $G = (P, E)$ is a set P of points embedded in the plane, and a set E of edges, where each $e \in E$ is a pair of points (u, v) in P, and the weight of e is the Euclidean distance $|uv|$.

A *local routing algorithm* A is an algorithm that routes a packet through the geometric graph G from a source vertex s to a target vertex t using only knowledge of the locations of s and t, as well as the location of the current vertex and its adjacent vertices. Let $\mathcal{P}\langle s, t\rangle$ be the path found in G from s to t using A. The *routing ratio* of A for any two points s and t in the geometric graph G is the ratio of the length of $\mathcal{P}\langle s, t\rangle$ to the Euclidean distance from s to t. An algorithm A has a routing ratio c for a class of geometric graphs $\mathcal{G}$, if, for any two vertices s and t in $G \in \mathcal{G}$, $|\mathcal{P}\langle s, t\rangle| \leq c \cdot |st|$.

A graph $G = (P, E)$ is a c-spanner if for any pair of points u and v in P the shortest path in G is not longer than $c|uv|$. The value c is referred to as the *stretch factor* or *spanning*

© Nicolas Bonichon, Prosenjit Bose, Jean-Lou De Carufel, Vincent Despré, Darryl Hill, Michiel Smid; licensed under Creative Commons License CC-BY
26th Annual European Symposium on Algorithms (ESA 2018).
Editors: Yossi Azar, Hannah Bast, and Grzegorz Herman; Article No. 22; pp. 22:1–22:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

ratio of G. The stretch factor of G is thus a lower bound on the routing ratio of G for any routing algorithm A, and the routing ratio is an upper bound on the spanning ratio of G. Geometric spanners are described in detail in the book by Narasimhan and Smid [12].

A notable geometric graph is the *Delaunay triangulation*. Given a set P of points in the plane, we construct the Delaunay triangulation of P as follows. For each triple (p, q, r) of points in P, let C be the circle through p, q, and r. If there are no points of P in the interior of C, then we connect p, q, and r by edges to form a triangle. In this paper we assume that P is in general position: no 3 points are colinar and no 4 points are cocircular.

The Delaunay triangulation was first proven to be a spanner by Dobkin et al. [10], who showed an upper bound of 5.08 on the spanning ratio. This was subsequently improved to 2.42 by Keil and Gutwin [11], and then to 1.998 by Xia [13]. Xia and Zhang proved later that there exist Delaunay triangulations with spanning ratio greater than 1.59 [14].

Bose and Morin [6] explored some of the theoretical limitations of routing, and provided some of the first deterministic routing algorithms with constant routing ratio on the Delaunay triangulation. They denoted the spanning ratio found by Dobkin et al. [10] as $c_{dfs} \approx 5.08$. They showed that it is possible to locally route on the Delaunay triangulation with a routing ratio of $9 \cdot c_{dfs} \approx 45.749$. Bose et al. [4] further improved this bound to ≈ 15.479. Then, Bonichon et al. [2] showed that we can locally route on the Delaunay triangulation with a routing ratio of at most 5.9. In the same paper it was shown that the routing ratio of any deterministic local algorithm is at least 1.70 for the Delaunay triangulation.

Efforts to evaluate the spanning ratio and routing ratio have been made for Delaunay triangulations defined on other metrics. We can define these metrics by taking a convex shape and translating and scaling it until it intersects three vertices but contains no points of P in its interior. When we use a circle we obtain the L_2, or classical Delaunay triangulation. When the metric is not specified (as in the rest of this paper), then we are referring to the L_2-Delaunay triangulation. The L_1-Delaunay triangulation uses an axis aligned square, while the L_∞-Delaunay triangulation uses a square tipped at 45 degrees. By rotating the point set 45 degrees, it is easy to show that the L_1 and L_∞ triangulations are equivalent. Bonichon et al. [3] showed that the L_1 and L_∞ Delaunay triangulations are $\sqrt{4 + 2\sqrt{2}} \approx 2.61$-spanners, and they showed that this bound is tight. On this triangulation, Chew [7] proposed a routing algorithm with routing ratio $\sqrt{10}$. Moreover, the routing ratio of any deterministic local algorithm is at least 2.70 for this class of graph [1]. The TD-Delaunay triangulation is constructed using an equilateral triangle. Chew [8] showed that they are 2-spanners. Bose et al. [5] proposed a routing algorithm of routing ratio $\sqrt{5/3} \approx 2.89$ and they show that this ratio is the best possible. Recently Dennis, Perkovic and Duru [9] showed that the stretch factor of *Hexagon*-Delaunay triangulation is 2 and this is tight.

Table 1 Spanning and Routing Ratios of Delaunay Triangulations. Tight results are shown in bold.

Graph	Spanning Ratio	Routing Ratio
TD-Delaunay	**2** [8]	$\mathbf{5/\sqrt{3} \approx 2.89}$ [5]
L_1 and L_∞-Delaunay	$\mathbf{\sqrt{4 + 2\sqrt{2}} \approx 2.61}$ [3]	$\sqrt{10} \approx 3.16$ [7]
Hexagon-Delaunay	**2** [9]	
L_2-Delaunay	1.998 [13]	3.56 (this paper)

In this paper we present a local routing algorithm, called *MixedChordArc*, for the L_2-Delaunay triangulation, with a routing ratio of 3.56. This improves the current best routing ratio of 5.9 [1]. Table 1 shows our result in the context of spanning and routing ratios of

other Delaunay triangulations.

In Section 2 we define a local algorithm that achieves this routing ratio. In Section 3 we prove the result for a special case, called *balanced configurations*. In Section 4 we extend the technique presented in Section 3 to prove the main result in the general case. In Section 5 we present our conclusions and our ideas for future directions for this line of research.

2 The MixedChordArc Algorithm

Let P be a finite set of points in the plane, and let $DT(P)$ be the Delaunay triangulation of P. We want to route a packet between two vertices of P along edges of $DT(P)$ using only local knowledge and knowledge of our start and destination vertices.

Let s and t be the start and terminal vertices respectively, and assume, without loss of generality, that s and t are on the x-axis with s to the left of t. Our general position assumption ensures that no other vertex lies on st. Consider two triangles T and T' whose interior is cut by st. We say that T is to the left of T', and T' is to the right of T, if, by following st starting at s we intersect T before T'. If uv is the edge shared by T and T', then our general position assumption ensures that u and v are on opposite sides of st.

Let C be a circle that intersects st. We denote by t_C the rightmost point of C on st. Let u and v be two points on C. We denote by $\mathcal{A}_C(u,v)$ the clockwise arc of C from u to v, and by $\mathcal{B}_C(u,v)$ the counter-clockwise arc of C from u to v. We denote the length of a continuous curve S by $|S|$.

Let $p \neq t$ be the vertex representing the current location of the packet. We assume s to be above st, and we assume t to be on the opposite side of st from the current vertex. Let T be the rightmost triangle with p as a vertex whose interior is cut by st. Let $a \neq p$ be the vertex of T that is above st, and let $b \neq p$ be the vertex of T that is below st. Let C be the circumcircle of T.

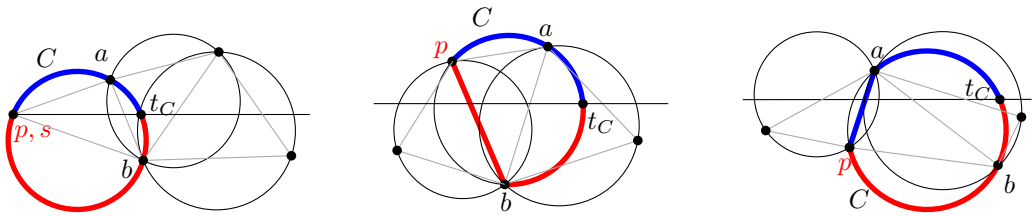
Here is the algorithm MixedChordArc. First assume that $p = s$. If $|\mathcal{A}_C(s,t_C)| \leq |\mathcal{B}_C(s,t_C)|$, set $p = a$, otherwise set $p = b$. See Fig. 1a. If $p \neq s$, we repeat the following until $p = t$.

1. If p is above st:
   - **a.** If $|\mathcal{A}_C(p,t_C)| \leq |pb| + |\mathcal{B}_C(b,t_C)|$, set $p = a$
   - **b.** Else set $p = b$.
2. If p is below st:
   - **a.** If $|\mathcal{B}_C(p,t_C)| \leq |pa| + |\mathcal{A}_C(a,t_C)|$, set $p = b$
   - **b.** Else set $p = a$.

Note that assuming that t is on the opposite side of st from p ensures that when t is a neighbour of the current vertex, the algorithm will forward the packet directly to t.

The possible choices are illustrated in Fig. 1. Let $\mathcal{P}\langle s,t\rangle = (s = p_0, p_1, ..., p_n = t)$ be the sequence of vertices produced by the algorithm. In this paper we prove the following theorem.

▶ **Theorem 1.** *The MixedChordArc Algorithm finds a path $\mathcal{P}\langle s,t\rangle$ from s to t whose length $|\mathcal{P}\langle s,t\rangle|$ is not more than $\mu|st|$, where $\mu = \sqrt{\frac{2}{1-\sin(1)}} < 3.56$.*



(a) From $p = s$, the blue arc is shorter than the red arc, so we forward to a.

(b) From p, the blue path is shorter than the red path, so we forward to a.

(c) From p, the blue path is shorter than the red path, so we forward to a.

Figure 1 Illustrating one step of the algorithm.

In some cases, the path produced by our algorithm is a *balanced configuration*. In such cases, the analysis of the length of $\mathcal{P}\langle s,t\rangle$ is much easier. In Section 3 we define what a balanced configuration is, and analyze the length of $\mathcal{P}\langle s,t\rangle$ for this specific case. Then, in Section 4, we analyze the length of $\mathcal{P}\langle s,t\rangle$ for the general case.

3 Bounding $|\mathcal{P}\langle s,t\rangle|$ in a Balanced Configuration

Let us consider a path $\mathcal{P}\langle s,t\rangle$ of vertices such that $p_0 = s, p_n = t$ and $p_{i-1}p_i$ is an edge of the rightmost triangle T_i of p_{i-1} that has a non-empty intersection with st. Let a_i and b_i be the other two vertices of T_i, where a_i is above st, and b_i is below st. Thus $p_i = a_i$ or $p_i = b_i$. Let $s = p_0 = a_0 = b_0$ and let $t = p_n = a_n = b_n$. Let C_i be the circumcircle of T_i, let r_i be its radius and let c_i be its center. Let C_0 be the circle centered at s with radius $r_0 = 0$. Let $\mathcal{T} = (T_1, T_2, ..., T_n)$, and let $\mathcal{C} = (C_0, C_1, ..., C_n)$ be the sequence of circles starting at C_0, followed by the circumcircles of $\mathcal{T}$. Note that the vertex of T_i that is on the opposite side of st to p_{i-1} may not be at the intersection of C_{i-1} and C_i. Thus we define a second intersection point of C_{i-1} and C_i as follows (p_{i-1} being one intersection point). If p_{i-1} is above st, then q_i is the lowest intersection of C_i and C_{i-1} (where "lowest" is defined by the point having the least y-coordinate). If p_{i-1} is below st, let q_i be the highest intersection of C_{i-1} and C_i (where "highest" is defined by the point having the greatest y-coordinate). Note that it is possible to have C_{i-1} and C_i intersect in two points, and still have $q_i = p_{i-1}$. See circle C_4 in Fig. 2. Observe that if T_i and T_{i-1} share an edge, then q_i is the vertex of T_i on the opposite side of st from p_{i-1}. See circles C_1, C_2, C_3, and C_5 in Fig. 2. To simplify the notation, we write t_i instead of t_{C_i}, and we write $\mathcal{A}_i(u,v)$ and $\mathcal{B}_i(u,v)$ instead of $\mathcal{A}_{C_i}(u,v)$ and $\mathcal{B}_{C_i}(u,v)$, respectively.

We say that a pair of consecutive circles C_{i-1} and C_i is *balanced* if $|\mathcal{A}_i(p_{i-1},t_i)| = |p_{i-1}q_i| + |\mathcal{B}_i(q_i,t_i)|$ when p_{i-1} is above st, and if $|\mathcal{B}_i(p_{i-1},t_i)| = |p_{i-1}q_i| + |\mathcal{A}_i(q_i,t_i)|$ when p_{i-1} is below st. A path $\mathcal{P}\langle s,t\rangle$ on a point set P is a *balanced configuration* when C_{i-1} and C_i are balanced for all $1 \le i \le n$.

3.1 Analysis Technique

▶ **Lemma 2.** *Let C_{i-1} and C_i be arbitrary circles of $\mathcal{C}$, where $1 \le i \le n$. Then*

1. $|p_{i-1}b_i| + |\mathcal{B}_i(b_i,t_i)| \le |p_{i-1}q_i| + |\mathcal{B}_i(q_i,t_i)|$ *when p_{i-1} is above st, and*
2. $|p_{i-1}a_i| + |\mathcal{A}_i(a_i,t_i)| \le |p_{i-1}q_i| + |\mathcal{A}_i(q_i,t_i)|$ *when p_{i-1} is below st.*

Proof. By the triangle inequality we have $|p_{i-1}b_i| \le |p_{i-1}q_i| + |\mathcal{B}_i(q_i,b_i)|$, from which 1 follows. Case 2 is symmetric. ◀

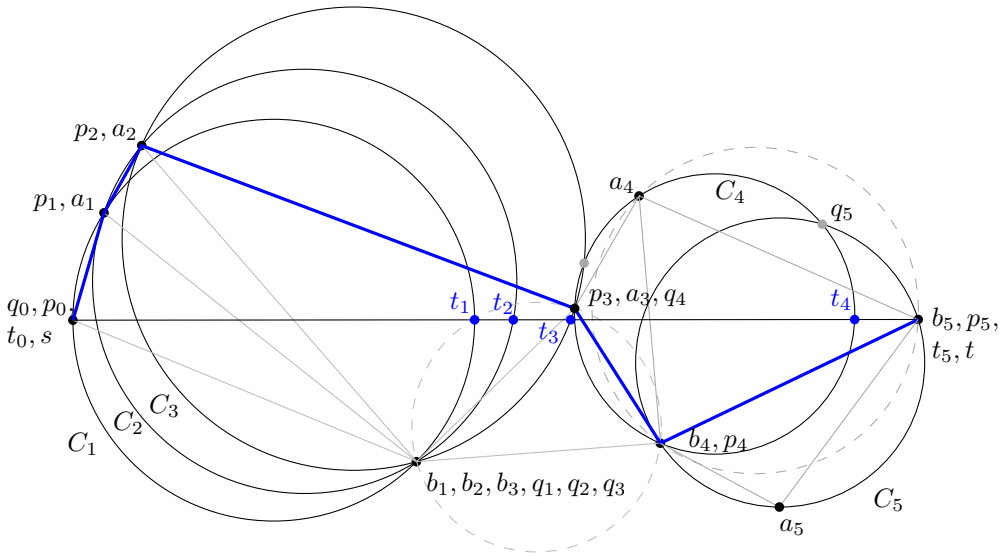


Figure 2 Sequence of circles in a balanced configuration and the path in blue. The dotted circles are circumcircles of triangles intersected by st but not in $\mathcal{T}$.

For the rest of this section, we assume that $\mathcal{P}\langle s,t\rangle$ is a balanced configuration. Consider the case when p_{i-1} is above st (the case when p_{i-1} is below st is symmetric). If $q_i = b_i$ then $|\mathcal{A}_i(p_{i-1},t_i)| = |p_{i-1}b_i| + |\mathcal{B}_i(b_i,t_i)|$, and the algorithm proceeds to a_i. If $q_i \neq b_i$, observe that $|p_{i-1}b_i| \leq |p_{i-1}q_i| + |\mathcal{B}_i(q_i,b_i)|$ by the triangle inequality (see circles C_4 and C_5 in Fig. 2). Thus we have $|p_{i-1}b_i| + |\mathcal{B}_i(b_i,t_i)| < |p_{i-1}q_i| + |\mathcal{B}_i(q_i,t_i)| = |\mathcal{A}_i(p_{i-1},t_i)|$, and the algorithm proceeds to b_i. Thus a balanced configuration allows for steps that cross st and steps that do not cross st. It also allows us to use $|\mathcal{A}_i(p_{i-1},t_i)|$ as an upper bound on $|p_{i-1}b_i| + |\mathcal{B}_i(b_i,t_i)|$ in the case where $p_{i-1}p_i$ crosses st.

Let $x(v)$ and $y(v)$ be the x and y-coordinates of a point v, respectively. Let s_i be a point on st such that $x(s_i) = x(t_i) - 2r_i$. We define the following potential function that we use to bound the length of $\mathcal{P}\langle s,t\rangle$.

▶ **Definition 3.** If p_{i-1} is above st, then

$$\Phi(C_{i-1},C_i) = |\mathcal{A}_i(p_{i-1},t_i)| - |\mathcal{A}_{i-1}(p_{i-1}t_{i-1})| - \lambda|s_{i-1}s_i| - (\mu-\lambda)|t_{i-1}t_i|.$$

Otherwise, if p_{i-1} is below st, then

$$\Phi(C_{i-1},C_i) = |\mathcal{B}_i(p_{i-1},t_i)| - |\mathcal{B}_{i-1}(p_{i-1}t_{i-1})| - \lambda|s_{i-1}s_i| - (\mu-\lambda)|t_{i-1}t_i|,$$

where $\lambda = \left(\frac{1+\sin(1)}{\cos(1)} - \pi/2 - 1\right)/2 \approx 0.42$ and $\mu = \sqrt{\frac{2}{1-\sin(1)}} < 3.56$.

See Fig. 2 and 3 for a complete example and an illustration of the potential functions. See Fig. 4 for an illustration of $\Phi(C_{i-1},C_i)$. Three lemmas are used to prove Theorem 1 for balanced configurations. The proof of Lemma 4 is found in Section 3.3 while the proof of Lemma 5 is in Section 3.2.

▶ **Lemma 4.** *Given a pair of balanced circles C_{i-1} and C_i,*

$$\Phi(C_{i-1},C_i) \leq 0.$$

▶ **Lemma 5.** *For any balanced configuration $\mathcal{P}\langle s,t\rangle$, $\sum_{i=1}^{n} |s_{i-1}s_i| \leq |st|$.*

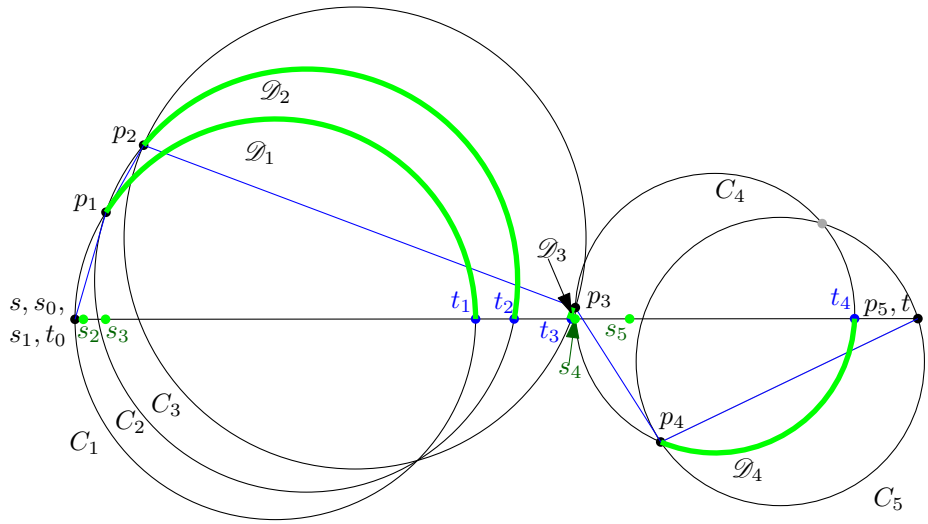


Figure 3 Illustrating the non-zero potential functions $\mathcal{D}_i, 1 \leq i \leq 4$ of a balanced configuration.

▶ **Lemma 6.** *For any $\mathcal{C}$, $x(t_{i-1}) < x(t_i)$ for all $1 \leq i \leq n$, and $\sum_{i=1}^{n} |t_{i-1}t_i| \leq |st|$.*

Proof. We prove that $x(t_{i-1}) < x(t_i)$, that is, t_i is right of t_{i-1} for all $1 \leq i \leq n$, by contradiction. Assume that $x(t_{i-1}) \geq x(t_i)$. If q_i is to the same side of st as p_{i-1}, then C_{i-1} must contain the vertex of T_i on the opposite side of st. If q_i is on the opposite side of st as p_{i-1}, then C_{i-1} contains the vertex of T_i on the same side of st as p_{i-1}. Both cases contradict the construction of a Delaunay triangulation. This, together with the fact that $t_0 = s$ and $t_n = t$ implies the second part of the lemma. ◀

▶ **Lemma 7.** *For $1 \leq i \leq n$, if p_{i-1} is above st, then*

1. **a.** $|\mathcal{A}_i(p_{i-1}, t_i)| > |p_{i-1}p_i| + |\mathcal{A}_i(p_i, t_i)|$ *if p_i is above st, and*
   b. $|\mathcal{A}_i(p_{i-1}, t_i)| > |p_{i-1}p_i| + |\mathcal{B}_i(p_i, t_i)|$ *if p_i is below st*
   otherwise p_{i-1} is below st and
2. **a.** $|\mathcal{B}_i(p_{i-1}, t_i)| > |p_{i-1}p_i| + |\mathcal{B}_i(p_i, t_i)|$ *if p_i is below st, and*
   b. $|\mathcal{B}_i(p_{i-1}, t_i)| > |p_{i-1}p_i| + |\mathcal{A}_i(p_i, t_i)|$ *if p_i is above st.*

Proof. Case 1a is because $|\mathcal{A}_i(p_{i-1}, p_i)| > |p_{i-1}p_i|$, and Case 1b is because if p_i is below st, then the algorithm chose to cross st, which implies 1b. Case 2 is symmetric. ◀

Theorem 1 follows from Lemmas 4, 5, 6, and 7:

Proof. We first analyze the case when p_{i-1} is above st. Recall that in this case, $\Phi(C_{i-1}, C_i)$ is defined as

$$\Phi(C_{i-1}, C_i) = |\mathcal{A}_i(p_{i-1}, t_i)| - |\mathcal{A}_{i-1}(p_{i-1}t_{i-1})| - \lambda|s_{i-1}s_i| - (\mu - \lambda)|t_{i-1}t_i|.$$

If p_i is above st (same side of st as p_{i-1}), then $|\mathcal{A}_i(p_{i-1}, t_i)| > |p_{i-1}p_i| + |\mathcal{A}_i(p_i, t_i)|$ by Lemma 7. In this case, let $\mathcal{D}_i = \mathcal{A}_i(p_i, t_i)$. If p_i is below st, then $|\mathcal{A}_i(p_{i-1}, t_i)| > |p_{i-1}p_i| + |\mathcal{B}_i(p_i, t_i)|$ by Lemma 7. In this case, let $\mathcal{D}_i = \mathcal{B}_i(p_i, t_i)$. In both cases we have $|\mathcal{A}_i(p_{i-1}, t_i)| > |p_{i-1}p_i| + |\mathcal{D}_i|$.

Let $\Phi'(C_{i-1}, C_i)$ be the function defined by

$$\Phi'(C_{i-1}, C_i) = |p_{i-1}p_i| + |\mathcal{D}_i| - |\mathcal{D}_{i-1}| - \lambda|s_{i-1}s_i| - (\mu - \lambda)|t_{i-1}t_i|.$$

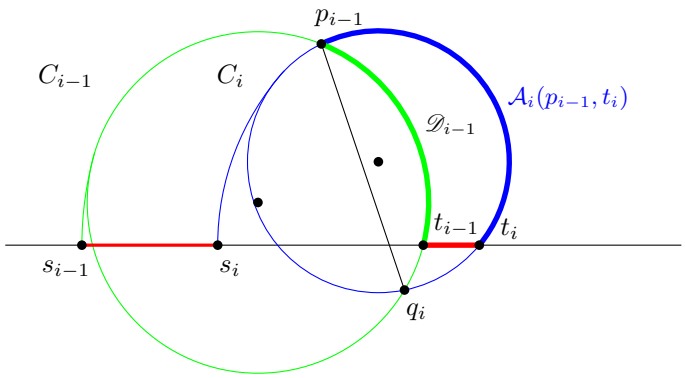


Figure 4 Illustrating the function $\Phi(C_{i-1}, C_i)$: blue minus green is charged to red to obtain an upper bound on the routing ratio.

Observe that $\Phi'(C_{i-1}, C_i) \leq \Phi(C_{i-1}, C_i)$. By Lemma 4, $\Phi(C_{i-1}, C_i) \leq 0$, thus $\Phi'(C_{i-1}, C_i) \leq 0$. When p_{i-1} is below st, a symmetric proof again shows us that $\Phi'(C_{i-1}, C_i) \leq 0$. Recall that $p_0 = t_0 = s$, and $p_n = t_n = t$, which means $|\mathcal{D}_0| = |\mathcal{D}_n| = 0$. Therefore we have

$$\sum_{i=1}^{n} \Phi'(C_{i-1}, C_i) \leq 0$$

from which we get:

$$\begin{aligned}
\sum_{i=1}^{n} (|p_{i-1}p_i| + |\mathcal{D}_i| - |\mathcal{D}_{i-1}|) &\leq \sum_{i=1}^{n} (\lambda |s_{i-1}s_i| + (\mu - \lambda)|t_{i-1}t_i|) \\
|\mathcal{P}\langle s, t\rangle| - |\mathcal{D}_0| + |\mathcal{D}_n| &\leq (\lambda + \mu - \lambda)|st| \qquad (1) \\
|\mathcal{P}\langle s, t\rangle| &\leq \mu |st|.
\end{aligned}$$

The right hand side of (1) is due to Lemmas 5 and 6. ◀

Lemma 5 is discussed in the next section. Lemma 4 is discussed in Section 3.3.

3.2 Proof of Lemma 5

Lemma 5 uses the following supporting result:

▶ **Lemma 8.** *Let C_{i-1} and C_i be balanced. Let s_{i-1} be the point on st where $x(s_{i-1}) = x(t_{i-1}) - 2r_{i-1}$ and let s_i be the point on st where $x(s_i) = x(t_i) - 2r_i$. Then $x(s_{i-1}) \leq x(s_i)$.*

Proof. See Fig. 5. Let u_{i-1} be the point on C_{i-1} that is diametrically opposed to t_{i-1} and let u_i be the point on C_i that is diametrically opposed to t_i. We will show the case when p_{i-1} is above st; the case when it is below st is symmetric. Since C_{i-1} and C_i are balanced, we have that $|\mathcal{A}_i(p_{i-1}, t_i)| = |p_{i-1}q_i| + |\mathcal{B}_i(q_i, t_i)|$ which implies that $|\mathcal{A}_i(p_{i-1}, t_i)| \leq \pi r_i$ and $|\mathcal{B}_i(q_i, t_i)| \leq \pi r_i$. Since $|\mathcal{A}_i(u_i, t_i)| = |\mathcal{B}_i(u_i, t_i)| = \pi r_i$, u_i is not on the open interval $\mathcal{A}_i(p_{i-1}, t_i)$ or $\mathcal{B}_i(q_i, t_i)$, which implies that either u_i is on the arc of C_i between p_{i-1} and q_i that does not contain t_i, or $u_i = p_{i-1} = q_i$. Lemma 6 implies that t_i is not inside C_{i-1}, which implies that u_i must be on or inside C_{i-1}. Let O_i be the circle centered at t_i with radius $|t_iu_i| = 2r_i$. Thus O_i and C_i are tangent at u_i, and O_i intersects st at s_i. Let O_{i-1} be the circle centered at t_{i-1} with radius $2r_{i-1}$. Thus O_{i-1} and C_{i-1} are tangent at u_{i-1}, and O_{i-1} intersects st at s_{i-1}. We prove the lemma by contradiction, thus assume that

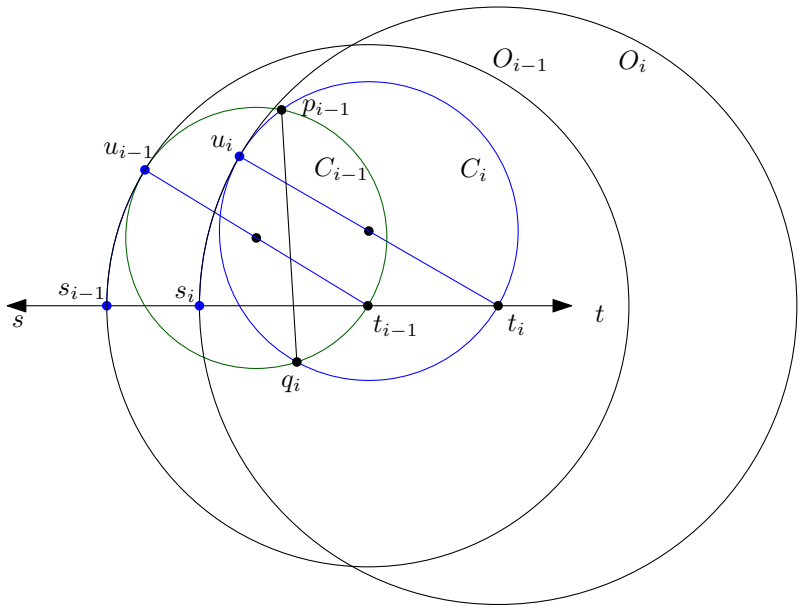


Figure 5 O_i must intersect O_{i-1} if C_{i-1} and C_i are path balanced, which implies that $x(s_{i-1}) \leq x(s_i)$.

$x(s_i) < x(s_{i-1})$. In the proof of Lemma 6, we showed that $x(t_i) > x(t_{i-1})$. Therefore, it must be that O_{i-1} is in the interior of O_i, and thus they do not intersect. Since u_i is on or inside C_{i-1}, and O_i intersects u_i, O_i must intersect C_{i-1}. But C_{i-1} is contained in O_{i-1} except for the point u_{i-1}, and O_{i-1} is contained in O_i, and thus O_i cannot intersect C_{i-1}, which is a contradiction. See Fig. 5. ◀

We can now prove Lemma 5:

Proof of Lemma 5. Follows from Lemma 8 and the fact that $x(s_0) = x(s)$ and $x(s_n) < x(t)$. ◀

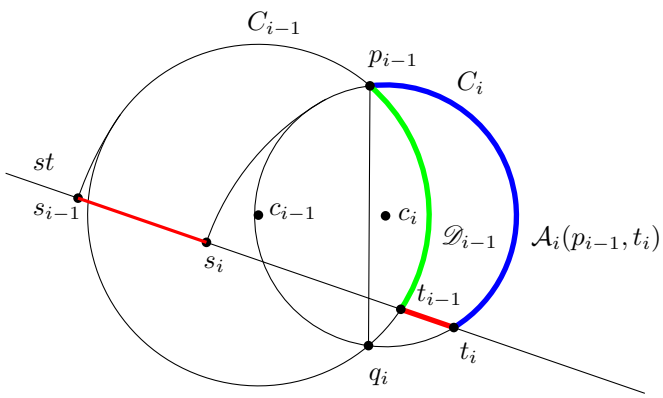


Figure 6 Coordinate system for analyzing $\Phi(C_{i-1}, C_i)$.

3.3 Proof of Lemma 4

To show that $\Phi(C_{i-1}, C_i) \leq 0$ when C_{i-1} and C_i are balanced, we set up the following coordinate system. We show the proof for the case when p_{i-1} is above st; the case when p_{i-1} is below st is symmetric. Let c_{i-1} and c_i lie along the x-axis, and let p_{i-1} and q_i lie along the y-axis. See Fig. 6. Lemma 4 follows from the following two lemmas:

▶ **Lemma 9.** *When C_{i-1} and C_i are balanced, if $y(t_{i-1}) \leq 0$, then $\Phi(C_{i-1}, C_i) \leq 0$.*

▶ **Lemma 10.** *When C_{i-1} and C_i are balanced, if $y(t_{i-1}) > 0$, then $\Phi(C_{i-1}, C_i) \leq 0$.*

The main tool to prove these two lemmas is the following transformation, which is similar to a transformation used by Xia [13].

▶ Transformation 11. Fix p_{i-1} and q_i, and translate c_i to the left along the x-axis until $c_i = c_{i-1}$. Moreover keep C_{i-1} unchanged and maintain C_i as the circle with center c_i with p_{i-1} on its boundary.

Observe that, after we have completed Transformation 11, we have $C_i = C_{i-1}$ and thus $\Phi(C_{i-1}, C_i) = 0$. If we can show that $\Phi(C_{i-1}, C_i)$ is increasing while $x(c_i)$ decreases, then it must be that $\Phi(C_{i-1}, C_i) \leq 0$ before Transformation 11. Thus we wish to find the change in $\Phi(C_{i-1}, C_i)$ with respect to the change in $x(c_i)$ during Transformation 11. Formally:

▶ **Lemma 12.** *If $\frac{d\Phi(C_{i-1},C_i)}{dx(c_i)} \leq 0$ during Transformation 11, then $\Phi(C_{i-1}, C_i) \leq 0$.*

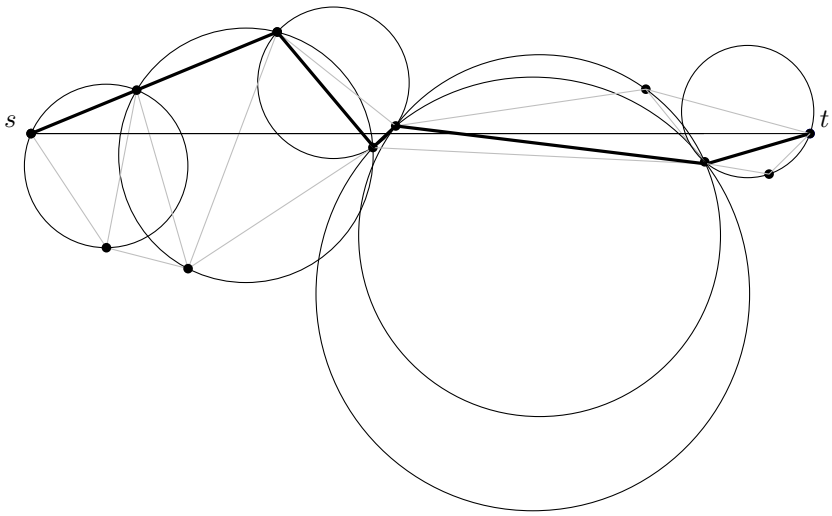
Proof. At the end of Transformation 11 we have that $\Phi(C_{i-1}, C_i) = 0$. If $\frac{d\Phi(C_{i-1},C_i)}{dx(c_i)} \leq 0$ then $\Phi(C_{i-1}, C_i)$ is not decreasing during Transformation 11, and thus $\Phi(C_{i-1}, C_i) \leq 0$ before Transformation 11. ◀

The analysis of this function is similar to Xia's approach [13]. To ensure that this transformation is well-defined, we require q_i to be below st. We observe that $\Phi(C_{i-1}, C_i)$ is maximized when st is on or above c_{i-1}, and this assumption implies q_i is below st (or on st, in the case where $p_{i-1} = q_i$). Full details of this analysis, the transformation analysis, and the proofs for Lemmas 9 and 10 have been left out due to space constraints.

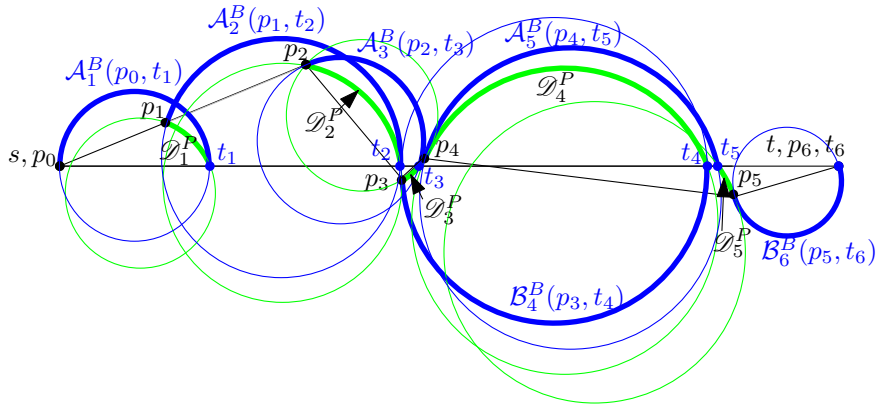
4 Bounding $\mathcal{P}\langle s, t\rangle$ in the General Case

In Section 3, we proved Theorem 1 for the case when the path produced by our algorithm results in a balanced configuration. In this section, we prove Theorem 1 for the general case. Given a sequence $\mathcal{C}$ of circles that intersect st, no series of transformations were found that could achieve a balanced configuration, while simultaneously providing a provable upper bound on the length of $|p_{i-1}, p_i|$. However, we were able to find *two* sequences of circles to substitute for $\mathcal{C}$. To represent each C_i in $\mathcal{C}$, we have a *potential circle* C_i^P and a *bounding circle* C_i^B. Like C_i, both C_i^P and C_i^B have t_i as their rightmost intersection with st. However, C_i intersects both p_i and p_{i-1}, while C_i^B is only required to intersect p_{i-1}, and C_i^P is only required to intersect p_i. If we look at a bounding circle C_i^B and the previous potential circle C_{i-1}^P, which intersect at p_{i-1}, they are balanced, and we can thus apply the function $\Phi(C_{i-1}^P, C_i^B)$ to relate the lengths of the arcs of these circles to $|st|$. Finally, when analyzed properly, they provide an upper bound on the length $|p_i p_{i-1}|$.

Formally, let C_0^P be the circle centered at $s = p_0$ with radius $r_0^P = 0$, and let C_n^P be the circle centered at t with radius $r_n^P = 0$. Assuming we have defined C_{i-1}^P, we will define C_i^B and C_i^P. If p_{i-1} is above st, let C_i^B be the circle through p_{i-1} and t_i for which $|\mathcal{A}_{C_i^B}(p_{i-1}, t_i)| = |p_{i-1}q_i'| + |\mathcal{B}_{C_i^B}(q_i', t_i)|$, where q_i' is the bottommost intersection



(a) The sequence of triangles $\mathcal{T}$ intersected by st, along with their circumcircles $\mathcal{C}$, and the path $\mathcal{P}\langle s,t\rangle$ found by the algorithm in bold.



(b) The complete set of bounding arcs and potential arcs used in the function $\Phi(C_{i-1}^P, C_i^B)$, used to bound the routing ratio in the general case.

Figure 7 The initial circumcircles in 7a, and the construction of the potential circles and bounding circles in the general case in 7b.

of C_{i-1}^P and C_i^B. If p_{i-1} is below st, let C_i^B be the circle through p_{i-1} and t_i for which $|\mathcal{B}_{C_i^B}(p_{i-1},t_i)| = |p_{i-1}q_i'| + |\mathcal{A}_{C_i^B}(q_i',t_i)|$, where q_i' is the topmost intersection of C_{i-1}^P and C_i^B. That is, C_{i-1}^P and C_i^B are balanced. Let r_i^B be the radius of C_i^B. The potential circle C_i^P is the circle through p_i, whose rightmost intersection with st is t_i, and whose radius is given by $r_i^P = \min\{r_i, r_i^B\}$ (with the exception of $r_n^P = 0$). Let s_i^P be the point on st with $x(s_i^P) = x(t_i) - 2r_i^P$, and let s_i^B be the point on st with $x(s_i^B) = x(t_i) - 2r_i^B$.

To simplify notation, for points u and v on C_i^P, instead of writing $\mathcal{A}_{C_i^P}(u,v)$ and $\mathcal{B}_{C_i^P}(u,v)$ to indicate clockwise and counter-clockwise arcs of C_i^P from u to v, respectively, we write $\mathcal{A}_i^P(u,v)$ and $\mathcal{B}_i^P(u,v)$. Likewise, for points u and v on C_i^B, instead of writing $\mathcal{A}_{C_i^B}(u,v)$ and $\mathcal{B}_{C_i^B}(u,v)$, we write $\mathcal{A}_i^B(u,v)$ and $\mathcal{B}_i^B(u,v)$.

See Figs. 7a and 7b for an example of the initial sequences $\mathcal{T}$ and $\mathcal{C}$ and the resulting bounding and potential arcs that we are interested in.

Since C^P_{i-1} and C^B_i are balanced, Φ can be extended to C^P_{i-1} and C^B_i, and thus we have

$$\Phi(C^P_{i-1}, C^B_i) = |\mathcal{A}^B_i(p_{i-1}, t_i)| - |\mathcal{A}^P_{i-1}(p_{i-1}, t_{i-1})| - \lambda|s^P_{i-1}s^B_i| - \mu|t_{i-1}t_i|$$

when p_{i-1} is above st and

$$\Phi(C^P_{i-1}, C^B_i) = |\mathcal{B}^B_i(p_{i-1}, t_i)| - |\mathcal{B}^P_{i-1}(p_{i-1}, t_{i-1})| - \lambda|s^P_{i-1}s^B_i| - \mu|t_{i-1}t_i|$$

when p_{i-1} is below st. Lemma 4 tells us that $\Phi(C^P_{i-1}, C^B_i) \leq 0$. To prove Theorem 1 in the general case, it is sufficient to prove the following two lemmas. Lemma 13 is a generalization of Lemma 5, whereas Lemma 14 is a generalization of Lemma 7.

▶ **Lemma 13.** $\sum_{i=1}^n |s^P_{i-1}s^B_i| \leq |st|$.

Proof. Since C^P_{i-1} and C^B_i are balanced, Lemma 8 tells us that $x(s^P_{i-1}) \leq x(s^B_i)$. We know that $x(s^P_i) = x(t_i) - 2r^P_i$ and $x(s^B_i) = x(t_i) - 2r^B_i$, thus the fact that $r^P_i = \min\{r_i, r^B_i\}$ implies that $x(s^B_i) \leq x(s^P_i)$. Thus $|s^P_{i-1}s^B_i| \leq |s^P_{i-1}s^P_i|$, and it is sufficient to show that $\sum_{i=1}^n |s^P_{i-1}s^P_i| \leq |st|$. The fact that $x(s^P_{i-1}) \leq x(s^B_i)$ implies that $x(s^P_{i-1}) \leq x(s^P_i)$, and C^P_0 is the circle centered at s with radius 0, and thus $s^P_0 = s$. Since $x(s^P_n) \leq x(t)$, this completes the proof. ◀

Due to space constraints, we omit the proof of the following lemma.

▶ **Lemma 14.** *For $1 \leq i \leq n$, if p_{i-1} is above st, then*

1. **a.** $|\mathcal{A}^B_i(p_{i-1}, t_i)| \geq |p_{i-1}p_i| + |\mathcal{A}^P_i(p_i, t_i)|$ *if p_i is above st, and*
   b. $|\mathcal{A}^B_i(p_{i-1}, t_i)| \geq |p_{i-1}p_i| + |\mathcal{B}^P_i(p_i, t_i)|$ *if p_i is below st*

   otherwise p_{i-1} is below st and
2. **a.** $|\mathcal{B}^B_i(p_{i-1}, t_i)| \geq |p_{i-1}p_i| + |\mathcal{B}^P_i(p_i, t_i)|$ *if p_i is below st, and*
   b. $|\mathcal{B}^B_i(p_{i-1}, t_i)| \geq |p_{i-1}p_i| + |\mathcal{A}^P_i(p_i, t_i)|$ *if p_i is above st.*

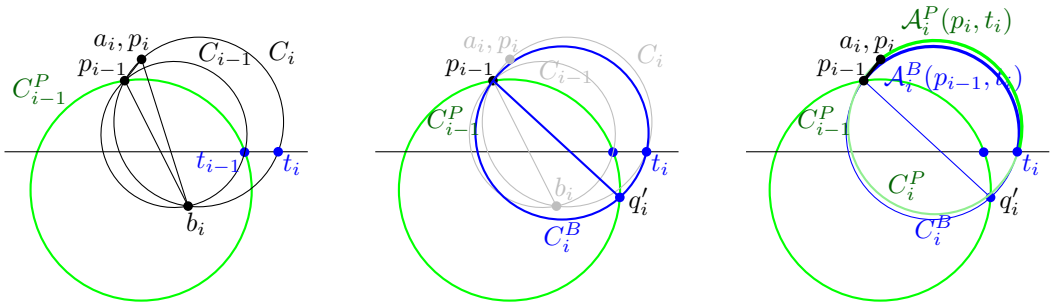
Theorem 1 follows from Lemmas 4, 6, 13, and 14.

Proof of Theorem 1. If p_i is above st, let $\mathcal{D}^P_i = \mathcal{A}^P_i(p_i, t_i)$. If p_i is below st, let $\mathcal{D}^P_i = \mathcal{B}^P_i(p_i, t_i)$. Let $\Phi'(C^P_{i-1}, C^B_i) = |p_{i-1}p_i| + |\mathcal{D}^P_i| - |\mathcal{D}^P_{i-1}| - \lambda|s^P_{i-1}s^B_i| - (\mu - \lambda)|t_{i-1}t_i|$. Lemmas 14 and 4 imply that $\Phi'(C^P_{i-1}, C^B_i) \leq \Phi(C^P_{i-1}, C^B_i) \leq 0$. Using $\Phi'(C^P_{i-1}, C^B_i)$ we get:

$$\begin{aligned}
\sum_{i=1}^n \Phi'(C_{i-1}, C_i) &\leq 0 \\
\sum_{i=1}^n \left(|p_{i-1}p_i| + |\mathcal{D}^P_i| - |\mathcal{D}^P_{i-1}|\right) &\leq \sum_{i=1}^n (\lambda|s^P_{i-1}s^B_i| + (\mu - \lambda)|t_{i-1}t_i|) \\
|\mathcal{P}\langle s, t\rangle| - |\mathcal{D}^P_0| + |\mathcal{D}^P_n| &\leq (\lambda + \mu - \lambda)|st| \qquad (2) \\
|\mathcal{P}\langle s, t\rangle| &\leq \mu|st|.
\end{aligned}$$

Line (2) follows from Lemmas 6 and 13. ◀

We give some insight into the selection of r^P_i. Assume that p_{i-1} is above st (when p_{i-1} is below st the explanation is symmetric). The purpose of $|\mathcal{A}^B_i(p_{i-1}, t_i)|$ is to bound $|p_{i-1}p_i| + |\mathcal{A}^P_i(p_i, t_i)|$, as expressed in Lemma 14. This lemma is also the reason for selecting the radius of C^P_i as $r^P_i = \min\{r_i, r^B_i\}$. It would be simpler to let $r^P_i = r^B_i$, since then we would have $s^P_i = s^B_i$. However, if we allow $r^P_i > r_i$, it can happen that the arc $|\mathcal{A}^B_{i+1}(p_i, t_{i+1})|$ on the next bounding circle is not large enough to cover $|p_ip_{i+1}| + |\mathcal{A}^P_{i+1}(p_{i+1}, t_{i+1})|$. See Fig. 8. Thus Lemma 14 would not hold. To account for this, we ensure that C^P_i has radius at most r_i.



(a) C_{i-1}, C_i, and C^P_{i-1}. Notice that $r^P_{i-1} > r_{i-1}$.

(b) C^B_i and its intersection with C^P_{i-1}.

(c) $|\mathcal{A}^B_i(p_{i-1}, t_i)| < |p_{i-1}, p_i| + |\mathcal{A}^P_i(p_i, t_i)|$.

Figure 8 The reasoning behind $r^P_i = \min\{r_i, r^B_i\}$. In this diagram, $r^P_i > r_i$, and we show why it is detrimental to our analysis. Notice that $|\mathcal{A}^B_i(p_{i-1}, t_i)| < |p_{i-1}, p_i| + |\mathcal{A}^P_i(p_i, t_i)|$. Thus the arc $\mathcal{A}^B_i(p_{i-1}, t_i)$ of the bounding circle is not long enough to pay for $|p_{i-1}, p_i| + |\mathcal{A}^P_i(p_i, t_i)|$.

5 Conclusion and Future Work

Consider the algorithm presented in Section 2, along with two variations. To keep the algorithms simple, assume we are at a vertex p above st. Otherwise all assumptions are the same as in Section 2.

A) **BestChord:** If $|pa| + |\mathcal{A}_C(a, t_C)| \leq |pb| + |\mathcal{A}_C(b, t_C)|$ then $p = a$ else $p = b$.
B) **MixedChordArc:** If $|\mathcal{A}_C(p, t_C)| \leq |pb| + |\mathcal{A}_C(b, t_C)|$ then $p = a$ else $p = b$.
C) **MinArc:** If $|\mathcal{A}_C(p, t_C)| \leq \pi r$ then $p = a$ else $p = b$.

The algorithm presented in this paper is *MixedChordArc*. Following the techniques used in [1] we are able to show that the routing ratio of *MinArc* is between 3.20 and 3.96. Since the routing ratio of 3.56 of *MixedChordArc* is better, we do not present the details of *MinArc*.

We suspect that *BestChord* is an improvement on *MixedChordArc*. It seems plausible that we can modify the proofs presented in this paper to obtain the same upper bound for *BestChord* as for *MixedChordArc*, but for now that remains unverified. Whether or not *BestChord* is asymptotically superior to *MixedChordArc*, or whether they are asymptotically the same is still unknown.

Although we have improved the upper bound of the routing ratio on the L_2-Delaunay triangulation, it is not clear how tight our analysis is. The upper bound on the analysis is where our potential function is the weakest. A more clever potential function could lower the routing ratio using a comparable analysis. Or perhaps one of the algorithms above would respond to a completely different style of analysis.

Furthermore, the lower bound on *MixedChordArc* is still the same as the lower bound on routing on the L_2-Delaunay triangulation in general, which is approximately 1.70 [1]. So it seems there is still much room for improvement. The question remains, what other algorithms or analysis can we use to improve the routing ratio of the Delaunay triangulation? And given that the upper and lower bounds on the spanning ratio of the L_2-Delaunay triangulation are 1.998 [13] and 1.5932 [14] respectively, is there a separation of the spanning and routing ratios of the Delaunay triangulation?

References

1. Nicolas Bonichon, Prosenjit Bose, Jean-Lou De Carufel, Ljubomir Perković, and André van Renssen. Upper and lower bounds for online routing on Delaunay triangulations. In Nikhil Bansal and Irene Finocchi, editors, *Algorithms - ESA 2015*, volume 9294 of *Lecture Notes in Computer Science*, pages 203–214. Springer Berlin Heidelberg, 2015. URL: http://dx.doi.org/10.1007/978-3-662-48350-3_18, doi:10.1007/978-3-662-48350-3_18.
2. Nicolas Bonichon, Prosenjit Bose, Jean-Lou De Carufel, Ljubomir Perković, and André van Renssen. Upper and lower bounds for online routing on Delaunay triangulations. *Discrete & Computational Geometry*, 58(2):482–504, Sep 2017. URL: https://doi.org/10.1007/s00454-016-9842-y, doi:10.1007/s00454-016-9842-y.
3. Nicolas Bonichon, Cyril Gavoille, Nicolas Hanusse, and Ljubomir Perković. Tight stretch factors for L_1 and L_∞ Delaunay triangulations. *Computational Geometry*, 48(3):237 – 250, 2015. URL: http://www.sciencedirect.com/science/article/pii/S0925772114001126, doi:https://doi.org/10.1016/j.comgeo.2014.10.005.
4. Prosenjit Bose, Jean-Lou De Carufel, Stephane Durocher, and Perouz Taslakian. Competitive online routing on Delaunay triangulations. In R. Ravi and Inge Li Gørtz, editors, *Algorithm Theory - SWAT 2014 - 14th Scandinavian Symposium and Workshops, Copenhagen, Denmark, July 2-4, 2014. Proceedings*, volume 8503 of *Lecture Notes in Computer Science*, pages 98–109. Springer, 2014. URL: http://dx.doi.org/10.1007/978-3-319-08404-6_9, doi:10.1007/978-3-319-08404-6_9.
5. Prosenjit Bose, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. Competitive routing in the half-theta-6-graph. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1319–1328. SIAM, 2012. URL: http://dl.acm.org/citation.cfm?id=2095116.2095220.
6. Prosenjit Bose and Pat Morin. Online routing in triangulations. In *Algorithms and Computation*, volume 1741 of *Lecture Notes in Computer Science*, pages 113–122. Springer Berlin Heidelberg, 1999. URL: http://dx.doi.org/10.1007/3-540-46632-0_12, doi:10.1007/3-540-46632-0_12.
7. L. Paul Chew. There is a planar graph almost as good as the complete graph. In *Proceedings of the Second Annual Symposium on Computational Geometry*, SCG '86, pages 169–177, New York, NY, USA, 1986. ACM. URL: http://doi.acm.org/10.1145/10515.10534, doi:10.1145/10515.10534.
8. L. Paul Chew. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2):205 – 219, 1989. URL: http://www.sciencedirect.com/science/article/pii/0022000089900445, doi:https://doi.org/10.1016/0022-0000(89)90044-5.
9. Michael Dennis, Ljubomir Perković, and Duru Türkoglu. The stretch factor of hexagon-Delaunay triangulations. *CoRR*, abs/1711.00068, 2017. URL: http://arxiv.org/abs/1711.00068, arXiv:1711.00068.
10. David P. Dobkin, Steven J. Friedman, and Kenneth J. Supowit. Delaunay graphs are almost as good as complete graphs. *Discrete & Computational Geometry*, 5(1):399–407, 1990. URL: http://dx.doi.org/10.1007/BF02187801, doi:10.1007/BF02187801.
11. J. Mark Keil and Carl A. Gutwin. Classes of graphs which approximate the complete euclidean graph. *Discrete & Computational Geometry*, 7(1):13–28, 1992. URL: http://dx.doi.org/10.1007/BF02187821, doi:10.1007/BF02187821.
12. Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks*. Cambridge University Press, New York, NY, USA, 2007.
13. Ge Xia. Improved upper bound on the stretch factor of Delaunay triangulations. In *Proceedings of the Twenty-seventh Annual Symposium on Computational Geometry*, SoCG

'11, pages 264–273, New York, NY, USA, 2011. ACM. URL: http://doi.acm.org/10.1145/1998196.1998235, doi:10.1145/1998196.1998235.

14 Ge Xia and Liang Zhang. Toward the tight bound of the stretch factor of Delaunay triangulations. In *Proceedings of the Canadian Conference on Computational Geometry*, CCCG '11, 2011.