Geometric Spanner Networks

Master Thesis

Chris Hoedemakers

Thesis committee:
prof.dr. Mark de Berg
prof.dr. Prosenjit Bose (supervisor)
dr.ir. Huub Van De Wetering

final version

Eindhoven, August 2015
Abstract

The topic of this thesis is geometric spanner networks. Geometric spanners are networks defined by a set of points $P$ and a set of straight line segments $E$, called edges, that connect the points in $P$. The aim of spanners is to balance the total number of edges in the network against the largest detour one has to make when taking the shortest path of edges from an arbitrary point in $P$ to any other. This largest detour defines the spanning ratio of the spanner. In this thesis we restrict our attention to the case where the points in $P$ are located in the 2-dimensional plane.

Within the area of geometric spanners we address two subtopics. First we propose a new type of spanner and prove that it has spanning ratio 1.998, and consists of at most $8n - 6$ edges. We furthermore show it has the desirable property that the shortest path between any two vertices consists of at most $53.2 \log n$ edges. This spanner is based on the Delaunay triangulation, and is constructed by means of the hierarchical data structure introduced by Kirkpatrick [18]. Using the same construction method, we show that it is possible to build a spanner with similar properties using any $\theta_k$-graph as a base. In particular, this spanner has spanning ratio $\frac{1}{1 - 2 \sin(\pi/k)}$, consists of $kn$ edges, and has a short path between any two vertices of $O(\log n)$ edges.

The second part of this thesis focuses on a specific spanner called the $\theta_5$-graph. A special version of the $\theta_5$-graph is the one where we introduce a set $S$ of line segments, called constraints. These constraints prevent us from drawing edges crossing them. Though (constrained) $\theta_k$-graphs have been studied extensively, and the spanning ratio of most of these graphs have been (tightly) proved, the spanning ratio of the constrained $\theta_5$-graph remains undetermined so far. In this thesis we make an attempt to bound it by generalizing the approach Bose et al. [7] used to bound the spanning ratio of the $\theta_5$-graph in the unconstrained setting. Unfortunately it turns out that their approach cannot be generalized easily to the constrained setting. We nonetheless present the proof to the point where it breaks down, and discuss alternative strategies that were considered.
# Contents

Abstract iii

Contents v

List of Figures vii

1 Introduction 1
   1.1 Geometric spanners 1
   1.2 Previous Work 3
   1.3 Contributions of the thesis 3

2 Preliminaries 5
   2.1 Geometric spanners 5
   2.2 Delaunay Triangulations 6

3 The Delaunay-Kirkpatrick Graph 7
   3.1 Kirkpatrick Hierarchy 7
   3.2 The Delaunay-Kirkpatrick Graph 9
   3.3 Spanner properties of the Delaunay-Kirkpatrick Graph 9

4 The $\theta_k$-Kirkpatrick Graph 17
   4.1 $\theta_k$-graphs 17
   4.2 The $\theta_k$-Kirkpatrick Graph 18
   4.3 Spanner properties of the $\theta_k$-Kirkpatrick Graph 19

5 The constrained $\theta_5$-graph 23
   5.1 Constrained $\theta_k$-graphs 23
   5.2 Spanning ratio of the constrained $\theta_5$-graph 24
   5.3 Geometric Lemmas 29

6 Conclusions 35

Bibliography 37

Appendix 39

A Trigonometric Identities 39

B Derivations of Lemma 5.3.4 40

Geometric Spanner Networks v
## List of Figures

2.1 Comparison of the complete graph, MST and Delaunay triangulation. ........... 5  
2.2 Example of a Delaunay triangulation. ............................................. 6  
3.1 Sequence of Delaunay triangulations. ............................................. 8  
3.2 Construction of the Delaunay-Kirkpatrick graph. ............................... 9  
3.3 Example scenario for Algorithm 2. ............................................... 11  
3.4 Chain of fans intersecting $uw$. .................................................... 11  
3.5 First base case of Lemma 3.3.1 ................................................... 12  
3.6 Second base case of Lemma 3.3.1 .................................................. 12  
3.7 Example scenarios for Lemma 3.3.2. ............................................. 14  
4.1 Explanation of the $\theta_k$-graph. ............................................... 17  
4.2 Construction of the $\theta_k$K-graph. .............................................. 18  
4.3 Example scenario for Algorithm 3 .................................................. 20  
5.1 The closest vertex in a constrained $\theta_k$-graph. .............................. 23  
5.2 The canonical triangle $\triangle_{uw}$ ............................................... 24  
5.3 Case distinction for Lemma 5.2.1. .............................................. 24  
5.4 Case 1 of Lemma 5.2.1. ............................................................... 25  
5.5 Worst-case scenario for case 1. .................................................... 25  
5.6 Worst-case scenario for case 3. .................................................... 28  
5.7 Unproven region of Lemma 5.2.1. ................................................... 28  
5.8 Considering the location of $v_u$. .................................................. 28  
5.9 Example scenario for Lemma 5.3.1. .............................................. 29  
5.10 Example scenario for Lemma 5.3.2. ............................................ 29  
5.11 Minimum smallest canonical triangle. .......................................... 30  
5.12 Maximum smallest canonical triangle. ........................................... 30  
5.13 Relation between $\text{slope}(uw)$ and $X_{uw}$. .................................. 30  
5.14 Example scenario for Lemma 5.3.4. ............................................. 31  
5.15 Example scenario for Lemma 5.3.5. ............................................. 34
Chapter 1

Introduction

A geometric network consists of a set of points in the plane, and a set of straight line segments connecting these points. We refer to these sets as the vertex set \(V\), respectively the edge set \(E\). In computer science, a geometric network is modeled as a graph \(G\), and denoted by \(G = (V, E)\). The edges of a graph often have a weight assigned to them, which is used in the analysis of the network. For a geometric network, the weight of an edge equals the Euclidean distance between the two endpoints of this edge.

Geometric networks are an important means of modeling relations or processes within a system. By analyzing the network, e.g., tracing paths from one point in the network to another, we are able to gain insight in the connectivity of the system. Connectivity issues that may come to light through this analysis, e.g., large detours between certain points in the network, can then be tackled by optimizing the shortest path between these points. Because of their generality, geometric networks are used in various application areas, for instance to model road networks, public utility networks, or communication networks.

This thesis focuses on a specific type of network called geometric spanners. In particular we propose two new graphs in Chapters 3 and 4, and prove that they are spanners with favorable properties. In Chapter 5, an attempt is made to prove that a specific graph called the constrained \( \theta \)-graph, is a spanner. The remainder of this chapter sketches a context for the most important topics addressed in this thesis. Further preliminary information is provided in Chapter 2.

1.1 Geometric Spanners

Recall that \(G = (V, E)\) is a weighted graph whose vertex set \(V\) is a set of points in the plane, and whose edge set \(E\) is a set of straight line segments connecting the vertices in \(V\). The weight of each edge in the graph is defined by the Euclidean distance between the two endpoints of this edge.

A graph \(G\) is called a geometric spanner if there exists a short path between any two vertices in \(G\). More specifically, a geometric graph \(G\) is a \(t\)-spanner if for any pair of vertices \(u, w \in V\) there exists a set of edges in \(E\) that together make up a path from \(u\) to \(w\) of length at most \(t \cdot d(u, w)\), for some fixed constant \(t \geq 1\). Here, \(d(u, w)\) denotes the Euclidean distance between \(u\) and \(w\). In other words, if we let \(\delta_G(u, w)\) be the shortest path from \(u\) to \(w\) in \(G\), then for every pair of vertices \(u, w \in V\) it must hold that \(\frac{\delta_G(u, w)}{d(u, w)} \leq t\). We refer to \(t\) as the spanning ratio.

Typically, geometric spanners are sparse graphs that attempt to balance the total number of edges in the graph against the largest detour one may have to take when traveling from one vertex to another. Geometric spanners are used in numerous application areas such as motion planning and telecommunication networks. Imagine we are to connect a set of cities through a network of roads (assume we are only allowed to build straight roads between cities). We want to minimize construction costs by not building too many roads. However, we also want to avoid that traffic has to make large detour when traveling the network.

Chew [11] was the first to study geometric spanners. Given a point set \(P\) of size \(n\), he presented
a \sqrt{10}-spanner for \( P \), consisting of \( O(n) \) edges.\(^1\) Over the years, different construction methods for spanners have been introduced and analyzed. For an elaborate overview on geometric spanners we refer to the survey by Bose and Smid [8], as well as the book by Narasimhan and Smid [19]. In this thesis we focus on two existing types of geometric spanners: Delaunay triangulations and \( \theta_k \)-graphs. We introduce these spanners in the upcoming section. Before we do so, we consider some other properties for geometric spanners to have, and that are desirable in certain application areas.

**Weight.** The first property we consider is the weight of a spanner. The weight of a graph \( G = (V,E) \) is defined as \( w(G) = \sum_{uv \in E} w(u,v) \), where \( w(u,v) \) denotes the weight of the edge \( uv \). Recall that in this thesis we define the weight of each edge in the graph by the Euclidean distance between the two endpoints of this edge. In the road construction example mentioned earlier, a low spanner weight is desirable since this bounds the total length of roads in the network, and thereby the corresponding construction costs. Since a geometric spanner must be connected, a lower bound on its weight is the weight of the minimum spanning tree (MST). The goal when minimizing the weight is therefore to construct a spanner whose weight is \( O(w(MST)) \).

**Maximum degree.** Keeping the road construction example in mind, we may also want to prevent cities from becoming a very large interchange. In terms of graph theory this means that we want to bound the maximum degree of the spanner. The maximum degree of a graph \( G \) is defined as the smallest integer \( \Delta \), such that every vertex \( v \in G \) has degree at most \( \Delta \), where the degree of \( v \) denotes the number of edges incident to \( v \).

**Size.** In this thesis however, we focus on two other spanner properties. The first of these is the size of the spanner. The size of a graph \( G \) is defined by the total number of edges in \( G \). Note that this seems similar, but is different from the weight of the graph, which is defined as the sum of the weights of the edges in \( G \). Typically we want a spanner to be of size linear in the number of points in the graph.

**Hop-diameter.** The second spanner property we address in this thesis is the hop-diameter. The hop-diameter of a \( t \)-spanner is defined as the smallest integer \( H \) such that for every pair of vertices \( u \) and \( w \) in the spanner, there exists a path of length at most \( t \cdot d(u,w) \), consisting of at most \( H \) edges. Throughout this thesis we refer to the hop-diameter by the diameter when it is clear by the context.

Having a network with small hop-diameter may be very beneficial in certain application areas. Consider for instance a wireless communication network, where the vertices represent a set of routers, and the edges represent the communication between these routers. The communication speed of this network strongly depends on the number of routers through which a message has to pass in order to get from its source to its destination. We therefore want to minimize this number, i.e., the hop-diameter. However, we neither want to connect every two routers in the network, since this increases the number of frequencies that are needed to prevent interference among the routers. In other words, it is desirable to have a network of small size. Combining these facts, we conclude that we like to define a spanner of small size and small hop-diameter.

Arya et al. [2] were the first to consider the hop-diameter of geometric spanners. They constructed a randomized \( t \)-spanner based on skip lists, as well as a deterministic \( t \)-spanner based on the well-separated pair decomposition of the point set. Both these spanners have size \( O\left(\frac{n}{t-1}\right) \) and hop-diameter \( O(\log n) \).\(^2\) In the former case, these bounds are expected. Narasimhan and Smid [19] introduced a data structure they call the dumbbell tree, and used it to construct a \( t \)-spanner of size \( O(n) \) and hop-diameter \( O(\alpha(n)) \), where \( \alpha \) stands for the inverse Ackermann function [1]. They derive this result from a spectrum of spanners: a \( t \)-spanner of size \( O(n \log n) \)

---

\(^1\)Throughout this thesis we use \( n \) to refer to the size of a point set \( P \).

\(^2\)Throughout this thesis we use \( \log n \) to denote \( \log_2 n \).
and hop-diameter 2, a $t$-spanner of size $O(n \log \log n)$ and hop-diameter 3, and a $t$-spanner of size $O(n \log^2 n)$ and hop-diameter 4. In all of these cases it holds that in the 2-dimensional space, the implicit constant for the size of the spanner is $\frac{\log(1/(t-1))}{(t-1)^2}$.

1.2 Previous Work

In order to evaluate the properties of the spanners we introduce in this thesis, we provide some context by addressing previous work on which we base our new spanners.

**Delaunay Triangulations.** The first type of graph on which we base our new spanner is the Delaunay triangulation. The Delaunay triangulation $DT(P)$ of some point set $P$ is defined as follows. A triple of points $(u,v,w)$ forms a triangle of $DT(P)$ if the unique disk with $u$, $v$ and $w$ on its boundary does not contain any points of $P$ in its interior.

Other types of Delaunay graphs have been defined where the disk is replaced by some other convex shape such as a square. Chew [11] showed that the $L_1$ metric Delaunay triangulation (having empty diamond region), as well as the Delaunay triangulation with empty equilateral triangle is a spanner. In this thesis we limit our attention to the traditional Delaunay triangulation with empty circle region. Dobkin, Friedman and Supowit [14] showed that this Delaunay triangulation with empty circle region is a spanner with spanning ratio $1 + \frac{\sqrt{5}}{2} \pi \approx 5.08$. Xia [22] later improved this bound to $1.998$. The best known lower bound on the spanning ratio for Delaunay triangulations is $1.5932$ as shown by Xia and Zhang [23].

**$\theta_k$-graphs.** $\theta_k$-graphs are the second type of spanners we work with in this thesis. A $\theta_k$-graph is constructed as follows. For each vertex $u$, we partition the space around it into $k$ cones of equal angle, i.e., $\frac{2\pi}{k}$. In each non-empty cone $C$ we then draw an edge from $u$ to the vertex $w$, whose orthogonal projection on the bisector of $C$ lies closest to $u$. We explain $\theta_k$-graphs in more detail in Section 4.1.

$\theta_k$-graphs were introduced independently by Clarkson [12] and Keil [17][16]. Both proved a spanning ratio of $\frac{1}{\cos \sigma - \sin \sigma}$, where $\sigma = \frac{2\pi}{k}$ and $k > 8$. Ruppert and Seidel [20] later improved the spanning ratio to $1 - 2\sin \left(\frac{\pi}{k}\right)$, where $k > 6$, which we use in this thesis to bound the spanning ratio of one of our new spanners. Better bounds on the spanning ratio of $\theta_k$-graphs for $k \geq 6$ have been presented by Bose et al. [10]. They distinguish between four categories of $\theta_k$-graphs; $k = 4i + 2$, $k = 4i + 3$, $k = 4i + 4$, and $k = 4i + 5$, where $i \in \mathbb{N}^+$. Most of these spanning ratios are tight and can be found in Table 1.1, where we compare them to the graphs introduced in this thesis.

1.3 Contributions of the thesis

**The Delaunay-Kirkpatrick graph.** In Chapter 3 of this thesis we present a new type of spanner with small hop-diameter, which we refer to as the Delaunay-Kirkpatrick graph. Throughout this thesis we abbreviate this to the DK-graph. The construction of this graph is based on the Delaunay triangulation. We show that the DK-graph has spanning ratio 1.998, size at most $8k - 6$, and hop-diameter at most $53.2 \log n$.

To put the properties of this new graph into perspective, we compare them to those of the existing graphs mentioned in this introduction. Table 1.1 gives an overview of this comparison. Note that the clear benefit of the DK-graph over the Delaunay triangulation is the fact that its hop-diameter is bounded, while its spanning ratio and the order of its size have not changed.

The spanners by Narasimhan and Smid [19], based on their dumbbell tree data structure, are more difficult to match. The first spanner in their spectrum has a diameter of merely 2. Its size is $3^{\log^* n} = \min\{s \geq 0 : \log \log \ldots \log n \leq 1\}$.
however a factor \( \log n \) larger than our result. Their spanner of diameter \( O(\alpha(n)) \) does have the same order of size as the DK-graph, and since \( \alpha(n) \) is less than 5 for any practical \( n \), its diameter is asymptotically smaller. The main advantage of the Delaunay-Kirkpatrick graph in this case is that it can be built using a much simpler construction method, and uses a simple algorithm for finding a short path between two vertices.

The \( \theta_k \)-Kirkpatrick graph. In Chapter 4 we propose an altered version of the DK-graph, whose construction is based on a \( \theta_k \)-graph rather than the Delaunay triangulation. We refer to this graph as the \( \theta_k \)-Kirkpatrick graph and abbreviate it to \( \theta_k \)-graph throughout this thesis. The \( \theta_k \)-graph has spanning ratio \( \leq \frac{1}{1-2\sin\left(\frac{\pi}{k}\right)} \), size at most \( kn \), and hop-diameter at most \( g(k) \cdot \log n \), where \( g(k) = \frac{2}{\log\left(\frac{k}{\sin(\pi/k)}\right)} \). When comparing these properties to those of \( \theta_k \)-graphs, we observe that its spanning ratio is slightly worse than the mostly tight spanning ratios of the \( \theta_k \)-graphs. We note however that the \( \theta_k \)-graph bounds the hop-diameter, while having the same order of size. This is a big improvement compared to the previous bound of \( O(n) \).

<table>
<thead>
<tr>
<th>graph</th>
<th>spanning ratio</th>
<th>size</th>
<th>hop-diameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delaunay Triangulation [22]</td>
<td>1.998</td>
<td>( 3n - 6 )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>dumbbell tree graph [19]</td>
<td>( t )</td>
<td>( O(f(t) \cdot n \log n) )</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>( t )</td>
<td>( O(f(t) \cdot n \log \log n) )</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>( t )</td>
<td>( O(f(t) \cdot n \log^* n) )</td>
<td>4</td>
</tr>
<tr>
<td>DK-graph [this thesis]</td>
<td>1.998</td>
<td>( 8n - 6 )</td>
<td>( 53.2 \log n )</td>
</tr>
<tr>
<td>( \theta_{4(2^i+1)} )-graph [4]</td>
<td>( 1 + 2\sin(\frac{\pi}{k}) )</td>
<td>( kn )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>( \theta_{4(2^i-1)} )-graph [4]</td>
<td>( \frac{\cos(\frac{\pi}{2k})}{\cos(\frac{\pi}{k}) - \sin(\frac{\pi}{2k})} )</td>
<td>( kn )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>( \theta_{4(2^{i+3})} )-graph [10]</td>
<td>( 1 + \frac{2\sin(\frac{\pi}{k})}{\cos(\frac{\pi}{k}) - \sin(\frac{\pi}{2k})} )</td>
<td>( kn )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>( \theta_{4(2^{i+1})} )-graph [10]</td>
<td>( \frac{\cos(\frac{\pi}{k})}{\cos(\frac{\pi}{2k}) - \sin(\frac{\pi}{2k})} )</td>
<td>( kn )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>( \theta_k )-K-graph [this thesis]</td>
<td>( 1 - 2\sin(\frac{\pi}{k}) )</td>
<td>( kn )</td>
<td>( g(k) \cdot \log n )</td>
</tr>
</tbody>
</table>

Table 1.1: Comparison of the spanning ratio, size and hop-diameter of several existing spanners, and those introduced in this thesis. Here it holds that \( t > 1 \), \( f(t) = \log(t/(t-1)) \), \( g(k) = \frac{2}{\log\left(\frac{k}{\sin(\pi/k)}\right)} \) and \( \log^* n = \min\{s \geq 0 : \log \log \ldots \log n \leq 1\} \).

The constrained \( \theta_k \)-graph. Regarding our attempt to bound the spanning ratio of the constrained \( \theta_k \)-graph, we note that since it is unknown whether they are spanners, it is not possible to compare our findings to any others. We do however conclude from our efforts that the approach used for the proof of the unconstrained \( \theta_k \)-graph cannot easily be generalized to the constrained setting. This makes it difficult to predict whether the spanning ratio in the constrained setting is equal to that in the unconstrained setting, or whether the graph is a spanner at all in the constrained setting. Bose et al. [9] however showed that for all \( \theta_k \)-graphs where \( k > 5 \) those graphs are indeed spanners. As such, we conjecture that the constrained \( \theta_k \)-graph is a spanner.
Chapter 2

Preliminaries

2.1 Geometric spanners

Though we briefly explained geometric spanners in Section 1.2, we once more state its formal definition. Recall that in this thesis we define the weight of each edge in a graph by the Euclidean distance between the two endpoints of this edge.

**Definition 1.** A $t$-spanner $G$ is a weighted graph on a set of points in the plane, such that for every pair of vertices $u, w \in G$, it holds that $\delta_G(u, w) \leq t$, where $\delta_G(u, w)$ denotes the weight of the shortest path from $u$ to $w$ in $G$, $d(u, w)$ denotes the Euclidean distance between $u$ and $w$, and $t$ is a constant of at least 1.

![Figure 2.1: Comparing different graphs on a point set $P$ of size 6. (a) The complete graph: a 1-spanner of size 15, (b) a minimum spanning tree: a 5-spanner of size 5, and (c) the Delaunay triangulation: a $\frac{3}{2}$-spanner of size 9.](image)

Note that $t$ is referred to as the **spanning ratio**. When a graph is a $t$-spanner for constant $t$, we refer to it as a spanner.

As mentioned before, spanners aim to balance the total number of edges in the graph against the largest detour one may have to take when traveling from one point to another. To get a better understanding of what makes a good spanner, we consider both ends of this spectrum. We do this for a set $P$ of 6 points that are evenly distributed over a circle. First imagine we connect every two vertices in $P$ with each other. The resulting graph is called the **complete graph** of $P$. The complete graph is a 1-spanner, and in this example has size 15 (see Figure 2.1a). Though 15 may not seem very large, the exact number of edges in any complete graph is $\frac{n(n-1)}{2}$, which is quadratic in the number of vertices. The other end of the spectrum is the minimum spanning tree of $P$. Recall that the MST of $P$ minimizes the weight, and as such consists of only $n - 1$ edges. It may however have an arbitrary large spanning ratio. The MST in our example has size 5, as well as spanning ratio 5 (see Figure 2.1b). Let us however examine what happens when we add...
more points to \( P \), while keeping them evenly distributed over the circle. The MST of this graph on \( n \) points always looks like an \( n \)-gon where one edge is missing. This missing edge determines the spanning ratio of the graph. Assume that the distance between two consecutive points on the circle is \( c \), then the spanning ratio of the \( n \)-gon is \( \frac{(n-1)c}{c} = n - 1 \). Hence the spanning ratio will keep growing with every point we add to the graph. In conclusion, we want to define a graph on \( P \) that finds a balance between these two extremes. An example of such a graph is the Delaunay triangulation, which we discuss in the next section.

2.2 Delaunay Triangulations

As mentioned in Section 1.2, Delaunay triangulations may be defined using different convex shapes. In this thesis we only concern ourselves with the traditional Delaunay triangulation, which is defined as follows.

**Definition 2.** The Delaunay triangulation \( DT(P) \) of some point set \( P \), is the unique triangulation of \( P \) such that the circle circumscribing \( u, v \) and \( w \), for every triangle \( uvw \in DT(P) \), does not contain any other point \( p \in P \) in the interior of the circle.

![Figure 2.2: Example of a Delaunay triangulation \( DT(P) \), including the empty circumcircles of each triangle in \( DT(P) \).](image)

In this thesis, we assume all points are in general position, i.e., no three points are collinear and no four points are co-circular. Furthermore we refer to the circle circumscribing \( u, v \) and \( w \) as the circumscribed \( \triangle uvw \). Figure 2.2 shows an example Delaunay triangulation including the empty circumcircles for each triangle in the graph.

The size of a Delaunay triangulation \( DT(P) \) is \( 3n - 3 - z \), where \( z \) denotes the number of points on the convex hull of \( P \). We derive this size from Euler’s formula for planar graphs. This formula states that for any planar graph of \( n \) vertices, \( e \) edges and \( f \) faces, it holds that \( n - e + f = 2 \). Note that every edge in \( DT(P) \) is incident to 2 faces, that every bounded face is incident to 3 edges, and that the unbounded face is incident to \( z \) edges. Together this means that \( 2e = 3(f - 1) + z \), and hence \( 3f = 2e - z + 3 \). Using Euler’s formula we then derive that the total number of edges \( e = 3n - 3 - z \), which is linear in \( n \).

Dobkin [14] first proved that the Delaunay triangulation has a constant spanning ratio of at most \( \frac{1 + \sqrt{2}}{\pi} \approx 5.08 \). The most recent upper bound however has been provided by Xia [22], who showed that it has a spanning ratio of less than 1.998. We refer to his paper for the full proof of this statement.

From this we conclude the Delaunay triangulation makes a fair trade-off between size and spanning ratio. If we consider the example of the previous section, we observe that the Delaunay triangulation on this point set gives us a \( \frac{3}{2} \)-spanner of size 9 (see Figure 2.1c).
Chapter 3

The Delaunay-Kirkpatrick Graph

In this chapter we propose a new kind of graph we call the Delaunay-Kirkpatrick graph. This graph is based on the Delaunay triangulation and constructed similarly the point location data structure presented by Kirkpatrick [18]. We prove that the DK-graph is a 1.998-spanner of size at most $8n - 6$ and hop-diameter at most $53.2 \log n$.

3.1 Kirkpatrick Hierarchy

Let us consider some arbitrary point set $P$ of size $n$. Kirkpatrick [18] showed that we can construct a hierarchical data structure that allows for fast point location in a given planar subdivision with vertex set $P$. A planar subdivision with vertex set $P$ is a partition of the plane into polygonal regions, whose vertices are the vertices in $P$. The levels of the hierarchical data structure are defined by a sequence of planar subdivisions. In this thesis we use the Delaunay triangulation of $P$ to partition the plane. To create the first subdivision in the sequence, we add three vertices $a$, $b$ and $c$ to $P$, such that $abc$ forms a triangle containing all vertices in $P$. We then define the first subdivision as the Delaunay triangulation on $P \cup \{a, b, c\}$. We denote this initial triangulation by $T(0)$. From $T(0)$, we create the sequence $T(0), T(1), ..., T(h)$ of Delaunay triangulations. In order to explain how we construct this sequence we define the independent set and $D$-independent set:

**Definition 3.** A set $S \subseteq V$ of vertices of a graph $G = (V, E)$ is an independent set if the graph induced on $S$ has no edges, that is, if no two vertices in $S$ are connected by an edge in $E$.

**Definition 4.** A $D$-independent set $I_D$ is an independent set of vertices where each vertex in the set has degree less than $D$, for some constant $D$.

To create the triangulation $T(i + 1)$ in the sequence from its predecessor $T(i)$, we remove a $D$-independent set $I_D$ from $T(i)$, and retriangulate the "holes" that are formed by the deletion of vertices.

**Algorithm 1** $D$-independent set selection

**Input:** Planar graph $G$

**Output:** $D$-independent set $I_D$ of vertices of degree less than $D$

1. $S \leftarrow$ vertices of $G$, excluding $a$, $b$ and $c$, with degree less than $D$
2. $I_D \leftarrow \emptyset$
3. while $S$ is not empty do
   4. $v \leftarrow$ some vertex in $S$
   5. $I_D \leftarrow I_D \cup \{v\}$
   6. $S \leftarrow S \setminus \{v \cup N(v)\}$, where $N(v)$ is the set of neighbors of $v$
4. end while
5. return $I_D$
these vertices. During this process we never select \( a, b \) and \( c \), and continue until only these three vertices are left. Together they form the last triangulation \( T(h) \) in the sequence (see Figure 3.1). The number of triangulations we need to reduce \( T(0) \) to \( T(h) \) depends on the value of \( D \) and the size of \( I_D \) at each level. Kirkpatrick showed that we can always select a 9-independent set of size at least \( \frac{n}{18} \). This is achieved by creating a set \( S \) of all the vertices in \( T(i) \) with degree at most 8, repeatedly selecting a vertex \( v \) from \( S \) (while \( S \) is not empty), and deleting \( v \) together with all its neighbors from \( S \). Pseudocode for this algorithm is presented in Algorithm 1. The proof for the lower bound on the size of \( I_9 \) using this approach is based on Euler’s formula for planar graphs.

Biedl and Wilkinson [3] however presented a more involved algorithm that increases this lower bound to \( \frac{n}{6} \). More generally they show that for any planar graph it is possible to select a \( D \)-independent set of size at least \( \min \left\{ \frac{D}{4D-18} n, \frac{5}{23} n \right\} \). To achieve this result they consider all vertices of degree less than \( D \), and carefully select a vertex \( v \) from this set. However, instead of deleting \( v \) and its neighbors, they perform vertex contraction on \( v \) in some cases. We refer to their paper for the full explanation of this method. Here we only state the result of their research [3].

**Lemma 3.1.1.** In any planar graph we can select a \( D \)-independent set of size at least
\[
\min \left\{ \frac{D}{4D-18} n, \frac{5}{23} n \right\}.
\]

From this we conclude that for any triangulation \( T(i) \) in the sequence, we can find a \( D \)-independent set of size a constant times the number of vertices in \( T(i) \). Here we choose to select a 9-independent set at every triangulation. We choose this number in order to obtain a fair trade-off between the size and hop-diameter of the graph we construct from this sequence. From Lemma 3.1.1 we conclude that we can select such a set in every triangulation of size at least \( \frac{n}{6} \). From this we derive the following theorem. Recall that every triangulation in the sequence represents a level of the Kirkpatrick hierarchical data structure.

**Theorem 3.1.2.** Assume we construct a Kirkpatrick hierarchy for the Delaunay triangulation on a point set \( P \) of size \( n \), removing a 9-independent set of vertices at every level. The depth of this hierarchy is \( h \leq \log_2 \left( \frac{n}{3} \right) \approx 3.8 \log n \).

**Proof.** Define \( n(i) \) to be the number of vertices in \( T(i) \). From Lemma 3.1.1 we derive that at every Delaunay triangulation \( T(i) \) in the hierarchy, we can select a 9-independent set of size at least \( \frac{n(i)}{6} \). This means \( n(i) \leq \frac{5}{6} \cdot n(i-1) \), from which we derive that:

\[
3 = n(0) \leq \left( \frac{5}{6} \right)^h \cdot n
\]

Hence,

\[
\left( \frac{6}{5} \right)^h \leq \frac{n}{3}
\]

and so

\[
h \leq \log_{\frac{6}{5}} \frac{n}{3} \leq \log_2 \frac{n}{3} \approx 3.8 \log n
\]

Hence we need at most \( \log_2 \frac{n}{3} \approx 3.8 \log n \) triangulations in order to reduce \( T(0) \) to \( T(h) \). \( \square \)
CHAPTER 3. THE DELAUNAY-KIRKPATRICK GRAPH

3.2 The Delaunay-Kirkpatrick Graph

From the sequence of Delaunay triangulations that we defined in the previous section, we construct a new type of graph that we refer to as the Delaunay-Kirkpatrick graph. This graph is created by merging the triangulations \( T(0), T(1), \ldots, T(h) \) into one graph \( DK(P) = \bigcup_{i=0}^{h} T(i) \) (see Figure 3.2). We mark each vertex \( v \in DK(P) \) with a level, denoted by \( \text{level}(v) \), that refers to the triangulation \( T(i) \) in which the vertex last occurs, i.e., if \( v \) last occurs in \( T(i) \), then \( \text{level}(v) = i \). Let us elucidate the DK-graph through a number of observations.

Observation 3.2.1. For any vertex \( v \) in triangulation \( T(i) \) it holds that \( i \leq \text{level}(v) \leq h \).

The lower bound on \( \text{level}(v) \) follows directly from the definition of \( \text{level}(v) \). The upper bound is easily determined by observing that there is only a total of \( h \) triangulations in the hierarchy. Recall that in Lemma 3.1.2 we showed \( h \approx 3.8 \log n \).

Observation 3.2.2. Consider some vertex \( v \) of level \( i \) in the Delaunay-Kirkpatrick graph. All vertices adjacent to \( v \) in \( T(i) \) have level \( j > i \).

Again by definition every vertex in \( T(i) \) has level at least \( i \). Vertices adjacent to a vertex \( v \) of level \( i \) cannot have the same level, since this contradicts the fact that the vertices of level \( i \) form an independent set at \( T(i) \). Hence any vertex adjacent to \( v \) in \( T(i) \) has level \( j > i \).

Finally we prove an upper bound on the size of the Delaunay-Kirkpatrick graph.

Lemma 3.2.3. A Delaunay-Kirkpatrick graph \( DK(P) \), defined on point set \( P \) of size \( n \), has at most \( 8n - 6 \) edges.

Proof. Consider some vertex \( v \) in a DK-graph \( DK(P) \), defined on the point set \( P \) of size \( n \). Let us assume \( \text{level}(v) = i \). This means that in \( T(i) \), \( v \) has degree at most 8, and hence at most 8 edges are deleted when \( v \) is removed from \( T(i) \). In order to retriangulate this hole for the construction of \( T(i + 1) \), we need at most 5 new edges. Since every vertex is only deleted once, at most 5 new edges may be introduced in total. From Euler’s formula we derive that \( T(0) \) contains \( 3n - 6 \) edges. Combining these facts, we conclude that \( DK(P) \) has a total of at most \( 3n - 6 + 5n = 8n - 6 = O(n) \) edges.

3.3 Spanner properties of the Delaunay-Kirkpatrick Graph

Given a Delaunay-Kirkpatrick graph \( DK(P) \), we prove that there exists a path between any two vertices \( u \) and \( w \) in the graph of length at most \( 1.998 \cdot d(u, w) \), consisting of at most 53.2 \( \log n \) edges.
CHAPTER 3. THE DELAUNAY-KIRKPATRICK GRAPH

Algorithm 2 DK-graph short path construction

Input: Delaunay-Kirkpatrick graph $DK(P)$, and two vertices $u, w \in DK(P)$

Output: Set of vertices containing a path between $u$ and $w$ of length at most $1.998 \cdot d(uw)$, that consists of $O(\log n)$ edges

1: $v_u \leftarrow u$ ; $v_w \leftarrow w$
2: $v \leftarrow \text{null}$ ; $c_u \leftarrow \text{null}$ ; $e_w \leftarrow \text{null}$
3: $S \leftarrow \emptyset$ ; $U \leftarrow \emptyset$ ; $W \leftarrow \emptyset$
4: while $v_u \neq v_w$ do
5: \hspace{1em} if $\text{level}(v_u) \leq \text{level}(v_w)$ then
6: \hspace{2em} $v \leftarrow v_u$
7: \hspace{1em} else
8: \hspace{2em} $v \leftarrow v_w$
9: \hspace{1em} end if
10: \hspace{1em} consider $T(\text{level}(v))$ of $DK(P)$
11: \hspace{1em} if $v == v_u$ then
12: \hspace{2em} $S \leftarrow$ Delaunay triangle(s) $uxy$, intersecting $uw$, lying to the side of $e_u$ that contains $w$
13: \hspace{2em} $U \leftarrow U \cup S$
14: \hspace{2em} $e_u \leftarrow$ edge closest to $w$, intersecting $uw$, and defining a face of a triangle in $S$
15: \hspace{2em} $v_u \leftarrow$ the vertex of $e_u$ with smallest level
16: \hspace{1em} else
17: \hspace{2em} $S \leftarrow$ Delaunay triangle(s) $uxy$, intersecting $uw$, lying to the side of $e_w$ that contains $u$
18: \hspace{2em} $W \leftarrow W \cup S$
19: \hspace{2em} $e_w \leftarrow$ edge closest to $u$, intersecting $uw$, and defining a face of a triangle in $S$
20: \hspace{2em} $v_u \leftarrow$ the vertex of $e_w$ with smallest level
21: \hspace{1em} end if
22: end while
23: if $e_w \neq e_u$ then
24: \hspace{1em} $U \leftarrow U \cup \text{triangles between } e_u \text{ and } e_w, \text{ intersecting } uw \text{ in } T(\text{level}(v_u))$
25: \hspace{1em} end if
26: return $U \cup W$

In order to prove this, we propose an algorithm that, given a DK-graph $DK(P)$ and two vertices $u, w \in DK(P)$, returns a chain of triangles in $DK(P)$ that contains such a short path between these two vertices. We provide a detailed description of the algorithm in the next paragraph, and refer to Algorithm 2 for the pseudocode of this method.

In essence, Algorithm 2 performs two walks along the straight line segment $uw$. One walk starts at $u$ and moves towards $w$, while the other walk starts at $w$ and moves towards $u$. On their way towards each other, these walks collect triangles intersecting $uw$ and store them in two separate sets $U$ and $W$. The key for finding a short path between $u$ and $w$ lies in the fact that the triangles along $uw$ are collected at a specific triangulation $T(i)$ of $DK(P)$. The algorithm terminates when both walks meet each other (we later prove that this always happens).

Let us consider the walk from $u$ towards $w$ in more detail (the walk from $w$ towards $u$ is analogous). Assume this is the walk to perform the first step after initialization. We start this first step by checking the level of $u$, and consider the triangulation $T(\text{level}(u))$ in which $u$ last occurs. From this triangulation we add the triangle $uxy$, intersecting $uw$ to $U$. Note that on the first step, this can only be one triangle. From here on we keep track of two variables. First we define $e_u$ to be the edge that is part of a triangle in $U$ that intersects $uw$ closest to $w$. We call this edge the frontier edge of $u$. Secondly we let $v_u$ be the endpoint of $e_u$ with the smallest level. We call this vertex the frontier vertex. At the initialization of the algorithm, $e_u$ is undefined and $v_u = u$. The frontier edge $e_w$ and frontier vertex $v_u$ for the walk from $w$ to $u$ are defined in a similar way. Consider the example in Figure 3.3. At the end of the first step from $u$ to $w$, $e_u$ is the edge between the vertices with level 13 and 42, and $v_u$ is the vertex with level 13.
CHAPTER 3. THE DELAUNAY-KIRKPATRICK GRAPH

Figure 3.3: Example scenario depicting the third step of Algorithm 2 at $T(13)$. Note that $u$ and $w$ are not present anymore at this level. The blue triangles on the left and right have already been added to the sets $U$, respectively $W$. The light blue triangles are about to be added to $U$. The numbers in the figure indicate the levels of the corresponding vertices.

To determine which of the two walks to proceed next, we check which of the two frontier vertices has the lowest level. Note that in the example in Figure 3.3 this was the walk from $w$ to $u$ (since $\text{level}(v_u) = 13 > \text{level}(v_w) = 5$). At the third iteration of the while-loop, the actual situation depicted in the figure, it holds again that $\text{level}(v_u) < \text{level}(v_w)$, and hence we proceed on the walk from $u$ to $w$. This time there may be more triangles adjacent to $v_u$ that intersect $uw$. These triangles are colored light blue in Figure 3.3. Note that we only consider the triangles to the side of $e_u$ that contains $w$. We add all of these triangles to $U$, and update the frontier edge and vertex accordingly. The new frontier edge and vertex in the example in Figure 3.3 are indicated by $e_u'$, respectively $v_u'$.

We continue this process until $v_u = v_w$. Once this holds, there may however be a number of triangles in between $e_u$ and $e_w$ that have not been added to either $U$ or $W$. To make sure we have a complete chain of triangles from $u$ to $w$, we add these triangles at $T(\text{level}(v_u))$ to $U$.

Figure 3.4: A chain of fans intersecting $uw$.

In order to prove that Algorithm 2 always terminates, we make some observations about the triangles in $U \cup W$ returned by the algorithm. Since we only add triangles to $U$ and $W$ that intersect $uw$, we may consider these triangles as a chain, where the triangles are connected by shared edges. Moreover we observe that for every one of these triangles (except the first and last), two edges intersect $uw$, and the third edge lies entirely to one side of $uw$. When this third edge lies above $uw$, we say that the triangle is oriented upwards. If the edge lies below $uw$, we say the triangle is oriented downwards. By clustering adjacent triangles with the same orientation, we can define the triangles as a chain of fans. Each fan $F$ consists of $j > 0$ triangles that share the apex $f$ of $F$ (see Figure 3.4). We say a fan is oriented upwards when its apex lies below $uw$, and is oriented downwards when its apex lies above $uw$. Note that the orientation of consecutive fans in the chain is alternating.

**Lemma 3.3.1.** Algorithm 2 is guaranteed to terminate.

**Proof.** We start by observing that after every iteration of the while loop in Algorithm 2, one of the frontier edges progresses towards the other. This can be concluded from the following two
facts. First recall that when adding triangles to either $U$ or $W$, we only consider the triangles that lie to the side of the frontier edge that contains the other frontier edge. Furthermore we observe that the new frontier edge of either $u$ or $w$ can never contain the current frontier vertex of $u$, respectively $w$. Hence at every iteration of the while loop in Algorithm 2, $e_u$ and $e_w$ move closer to each other.

To show that $v_u = v_w$ at some iteration of the while-loop, we recall that we can regard the triangles returned by Algorithm 2 as a chain of fans intersecting $uw$, and with alternating orientation. We perform induction on the number of fans $i$ in between the frontier edges $e_u$ and $e_w$.

**[BASE $i = 1$]** If there is only one fan in between $e_u$ and $e_w$, then there are three possible situations for the positions of $v_u$ and $v_w$ (see Figure 3.5). In each of these situations we consider what happens when the number of triangles $j$ in the fan is equal to 1, and what happens when $j > 1$.

In situation (a), if $j = 1$, $v_u$ becomes equal to $v_w$ and we are done. If $j > 1$, then either $j$ is reduced by one, and we keep ending up in the same situation until $j = 1$, or alternatively, $j$ is reduced by one and we end up in situation (b) of this base case. Situation (b) may seem less trivial. However, by observing that both $v_u$ and $v_w$ are endpoints of $e_w$, we conclude that $\text{level}(v_u) > \text{level}(v_w)$. This means that the walk of $v_w$ is the one to proceed next. If $j = 1$, then $v_w$ becomes equal to $v_u$, and the algorithm terminates. If $j > 1$, then either $j$ is reduced by one, and we keep ending up in the same situation until $j = 1$, or $j$ is reduced by one and we end up in situation (c) of this base case. In situation (c), it holds regardless of the value of $j$ that $v_u = v_w$, and hence we are done.

![Figure 3.5: The three possible situations for the first base case of Lemma 3.3.1.](image)

**[BASE $i = 2$]** For the second base case there are again three possible situations (see Figure 3.6). For situation (a) there are four possible scenarios. If $v = v_u$ and $j = 1$ for the downward oriented fan, then $v_u$ will become equal to $v_u$. If $v = v_u$ and $j > 1$, we end up in either situation (a) or (b) of the first base case. If $v = v_w$ and $j = 1$, then we end up in situation (b) or (c) of the first base case. If $v = v_w$ and $j > 1$, then either $j$ is reduced by one, and we keep ending up in the same situation until $j = 1$, or we end up in situation (b) of this base case. For situation (b) there are only two possible scenarios. For both $v = v_u$ and $v = v_w$ it holds that if $j = 1$ for the fan not having apex $v$, then $v_u$ becomes equal to $v_w$. If $j > 1$ for this fan, then we end up in situation (b) or (c) of the first base case. For situation (c) it holds that if $j = 1$ for the fan adjacent to the edge to proceed next, we end up in situation (a) or (b) of the first base case. If $j > 1$ for this fan, then
CHAPTER 3. THE DELAUNAY-KIRKPATRICK GRAPH

either $j$ is reduced by one, and we keep ending up in the same situation until $j = 1$, or we end up in situation (a) of this base case.

**[STEP i > 2]** When $i > 2$, the cases where $v = v_u$ and $v = v_w$ are analogous. Two scenarios may occur. If $v$ is the apex of some fan between $e_u$ and $e_w$, then the number of fans $i$ in between $e_u$ and $e_w$ is reduced by at least one and at most two. If $v$ is not the apex of such a fan, then $i$ is only reduced by one if $j = 1$. If $j > 1$, $j$ is reduced by one, and we keep ending up in the same situation until $j = 1$.

In all of the situations described above, either $i$ or $j$ decreases by at least one. We note that $j$ can only be reduced a finite number of times until $j = 1$. When this happens, either $i$ gets reduced, or $v_u = v_w$. This completes the proof that Algorithm 2 always terminates.

In order to prove that the DK-graph is a 1.998-spanner, we first prove the following lemma.

**Lemma 3.3.2.** Let $V$ be the set of distinct vertices defining the triangles in $U \cup W$, returned by Algorithm 2 after termination. The Delaunay triangulation $DT(V)$ on $V$ contains the same triangles intersecting $uw$ as those in $U \cup W$.

**Proof.** We prove this lemma by showing that for every triangle $abc \in U \cup W$ it holds that $O_{abc}$ does not contain any other point $x \in V$. If this holds, then we can conclude that $DT(V)$ contains the same triangles intersecting $uw$ as those in $U \cup W$. We apply induction on the number $m = i+j$ of iterations of the while-loop in Algorithm 2. Here, $i$ denotes the number of steps in the walk from $u$ to $w$, and $j$ denotes the number of steps in the walk from $w$ to $u$. We define $e_u^k$ as the $k$th frontier edge in the walk from $u$ to $w$, and define $e_w^k$ in a similar way for the walk from $w$ to $u$.

Assume that the walk from $u$ to $w$ moves to the right. The induction hypothesis then states the following: the circumcircle of every triangle in $U$ that intersects $uw$ to the left of frontier edge $e^1$, as well as the circumcircle of every triangle in $W$ that intersects $uw$ to the right of frontier edge $e^j$ is empty of points in $V$.

**[BASE m = 1]** Assume that Algorithm 2 terminates after one iteration of the while-loop. Since $u$ and $w$ are the initial frontier vertices, only one triangle is added to $U \cup W$. This means that $uw$ is an edge in the graph and hence, the algorithm would not have entered the while-loop at all. Since the triangle in $U \cup W$ was a Delaunay triangle when it was collected, the induction hypothesis holds for the base case.

**[STEP m > 1]** For the step case we assume that the induction hypothesis holds for $m$ iterations, and prove it holds for $m + 1$. We only prove this for the situation where the frontier edge $e^m = e_u^m$ and $e^{m+1} = e_w^m$. All other cases are analogous. Consider the example scenario in Figure 3.7b. By induction we know that the circumcircles of the triangles in $U$ intersecting $uw$ to the left of $e^m$, as well as the triangles intersecting $uw$ to the right of $e^m$, do not contain any points in $V$. This means we only need to concern ourselves with the triangles in between $e^m$ and $e^{m+1}$, indicated in blue in Figure 3.7b. Let $S$ be the set of these triangles, and let $C$ be the set of circumcircles of the triangles in $S$. To show that all circles in $C$ do not contain any vertices in $V$, we divide $V$ into four subsets.

[1] First we consider the vertices between $e^m$ and $e^{m+1}$. These are the vertices defining the triangles in $S$. Since these triangles are Delaunay triangles, it immediately follows that all circumcircles in $C$ are empty of other points in $V$.

[2] For the vertices of the triangles intersecting $uw$ between $e^{m+1}$ and $e^j$, we observe that they are present at level $(v_u^m)$, where $v_u^m$ is the frontier vertex of $e^m_u$ (see Figure 3.7b). If any of these points lie inside one of the circles in $C$ at $T$(level $(v_u^m)$), then this contradicts the fact that the triangles in $S$ are Delaunay triangles. Therefore the circles in $C$ do not contain any of the vertices between $e^{m+1}$ and $e^j$.

[3] Next we consider the vertices in $V$ defining the triangles intersecting $uw$ to the left of $e^m$. Recall that by induction the circumcircles of these triangles do not contain any points in $V$. This includes the triangle $abv$, directly to the left of $e^m$. Note that the circumcircle $O_{abv}$ of this triangle intersects $e^m$ at both its endpoints. Since the triangles in $S$ intersect $uw$ to the right of $e^m$, we know that the circles in $C$ intersect $e^m$ at the frontier vertex $v_u^m$, and somewhere between the two endpoints of $e^m$. From this we conclude that the parts of the circumcircles in $C$ that lie to
CHAPTER 3. THE DELAUNAY-KIRKPATRICK GRAPH

the left of $e^m$ (indicated in grey in Figure 3.7b), are subregions of $\bigcup_{abv_i}$, and are therefore empty of points in $V$ to the left of $e^m$. Finally we consider the vertices in $V$ defining the triangles intersecting $uw$ to the right of $e^j$. If the frontier edge $e^j$ does not exist however, then we are done. Hence assume $e^j$ exists, then the vertices closest to the circles in $C$ are the endpoints of $e^j$. We know that these two vertices are not contained in any of the circles in $C$ by similar reasoning as for the vertices in subset [2]. By similar reasoning as for subset [3], we conclude that the vertices of the triangles intersecting $uw$ to the right of $e^j$ cannot be contained in any of the circles in $C$. This concludes the step case and the proof of the lemma.

Figure 3.7: Example scenario for (a) the base case and (b) the step case of the inductive proof of Lemma 3.3.2.

The last statement we need in order to prove the spanning ratio of the Delaunay-Kirkpatrick graph, is the theorem presented by Xia [22] on the spanning ratio of the Delaunay triangulation. Xia shows that the short path between any two vertices $u$ and $w$ in a Delaunay triangulation $DT(P)$ is contained in the chain of triangles intersecting $uw$. We prove the spanning ratio of the DK-graph right after introducing the theorem by Xia [22]:

Theorem 3.3.3. The spanning ratio of a Delaunay triangulation in the plane is less than 1.998.

Theorem 3.3.4. The Delaunay-Kirkpatrick graph is a 1.998-spanner.

Proof. Let $V$ be the set of distinct vertices defining the triangles in $U \cup W$, returned by Algorithm 2 after termination. From Lemma 3.3.2 we know that the Delaunay triangulation $DT(V)$ contains the same triangles intersecting $uw$ as those in $U \cup W$. By Theorem 3.3.3 we know that this set of triangles contains a path from $u$ to $w$ of length at most $1.998 \cdot d(u,w)$. Hence Algorithm 2 finds a path from $u$ to $w$ of length at most $1.998 \cdot d(u,w)$, for any pair of vertices $u, w \in DK(P)$. Therefore the Delaunay-Kirkpatrick graph is a 1.998-spanner.

Finally we show that the Delaunay-Kirkpatrick graph is a spanner of hop-diameter at most $53.2 \log n$.

Theorem 3.3.5. The Delaunay-Kirkpatrick graph has hop-diameter at most $14 \log_2 n \approx 53.2 \log n$.

Proof. We prove this theorem by showing that the set of triangles $U \cup W$, returned by Algorithm 2 contains at most $14 \log_2 n \approx 53.2 \log n$ triangles.

By Theorem 3.1.2 we know that any DK-graph has at most $h = \log_2 n \approx 3.8 \log n$ levels. At any of these $h$ levels, Algorithm 2 may add at most 7 new triangles to both $U$ and $W$. This follows from the following two facts. First, the frontier vertex has degree at most 8, meaning it can define at most 7 triangles that may be added to either $U$ or $W$. Secondly it holds that every time we move one of the frontier edges, the level of its corresponding frontier vertex must increase. Therefore the total number of triangles in $U \cup W$ after termination of the algorithm is at most $14 \log_2 n \approx 53.2 \log n$.

Recall that by Theorem 3.3.4 the triangles in $U \cup W$ contain a path between any two vertices $u$ and $w$ in a DK-graph, of length at most $1.998 \cdot d(u,w)$. Combining this with the fact that this
path is contained in at most $53.2 \log n$ triangles, we conclude that the Delaunay-Kirkpatrick graph has hop-diameter at most $53.2 \log n$.

Note that the constant 53.2 depends on the value of $D$ of the $D$-independent set we remove at every level of the DK-graph. This means that we are able to decrease this constant in return for an increase in the size of the graph.

From Lemma 3.2.3, and Theorems 3.3.4 and 3.3.5 we derive the complete set of properties of the Delaunay-Kirkpatrick graph.

**Corollary 3.3.5.1.** The Delaunay-Kirkpatrick graph is a 1.998-spanner of size at most $8n - 6$ and hop-diameter at most $53.2 \log n$. 

Chapter 4

The $\theta_k$-Kirkpatrick Graph

In section 3.2 we introduced the Delaunay-Kirkpatrick graph. We constructed this graph by using the Delaunay Triangulation $T(0)$ over a point set $P$ as a basis. From $T(0)$ we created a sequence of Delaunay triangulations by subsequently deleting vertices. We then merged this sequence of triangulations into one graph: the DK-graph. In this chapter we show that when we use a $\theta_k$-graph, where $k \geq 7$, instead of the Delaunay triangulation, we again obtain a spanner of diameter $O(\log n)$. This spanner has size $kn$ and spanning ratio $\frac{1}{1-2\sin\left(\frac{\pi}{k}\right)}$. Before introducing this new graph we define $\theta_k$-graphs.

### 4.1 $\theta_k$-graphs

Section 1.2 shortly introduced $\theta_k$-graphs, and stated that Bose et al. [10] proved mostly tight bounds on the spanning ratio of $\theta_k$-graphs where $k \geq 6$. We construct $\theta_k$-graphs on a point set $P$, as follows. For each point $u \in P$ we partition the space around it into $k$ cones, each with angle $\frac{2\pi}{k}$. We number these cones from 0 to $k-1$ in clockwise order, and orient them in such a way that the bisector of cone 0 points straight up. We refer to cone $i$ of vertex $u$ as $C^u_i$ (see Figure 4.1a). We say that the boundary between two cones belongs to the cone located on the counter clockwise side of the boundary, i.e., the boundary between the cones $C_0$ and $C_1$ belongs to $C_0$. For every cone $C^u_i$ we draw an edge from $u$ to the vertex in this cone whose orthogonal projection onto the bisector of $C^u_i$ lies closest to $u$ (see Figure 4.1b). In other words, we take the line through $u$ perpendicular to the bisector of $C^u_i$, and sweep it along the bisector until we hit the first vertex in $C^u_i$. We then draw an edge from $u$ to this vertex. For simplicity we refer to this vertex as the closest vertex to $u$ in $C^u_i$. The set of edges from $u$ to its at most $k$ closest vertices is referred to as the set of outgoing edges of $u$. We also say that $u$ has outdegree at most $k$. For the remainder

![Figure 4.1](image.png)

Figure 4.1: Construction of the $\theta_k$-graph, where $k = 5$: (a) the cones of some vertex $u$; (b) the closest vertex in one of $u$ its cones; (c) the canonical triangle $\triangle_{uw}$.
of this thesis we assume that the points in $P$ are in general position, i.e., there is always a unique closest vertex in each cone $C^w_i$. Because of the way $\theta_k$-graphs are constructed, we can make the following observation about their size.

**Observation 4.1.1.** A $\theta_k$-graph on $n$ vertices has at most $kn$ edges.

We define the canonical triangle $\triangle_{uw}$ between two vertices $u$ and $w$ as the isosceles triangle region defined by the boundaries of the cone $C^w$ containing $w$, and the line through $w$ perpendicular to the bisector of $C^w$. Note that if $w$ is the vertex closest to $u$ in this cone, then this triangular region is empty. For every pair of vertices $(u, w)$ we can define two canonical triangles: $\triangle_{uw}$ and $\triangle_{wu}$. We denote the size of a canonical triangle $\triangle_{uw}$ by $|\triangle_{uw}|$, and define it as the length of the isosceles sides of the triangle (see Figure 4.1c).

### 4.2 The $\theta_k$-Kirkpatrick Graph

In this section we introduce the $\theta_k$-Kirkpatrick graph. Throughout this thesis we abbreviate this to $\theta_k$-K-graph. The $\theta_k$-K-graph is constructed in a similar way as the Delaunay-Kirkpatrick graph. Given some $\theta_k$-graph $\theta_k(0)$, where $k \geq 7$, we construct a sequence $\theta_k(1), \theta_k(2), \ldots, \theta_k(h)$ of $\theta_k$-graphs. Each graph $\theta_k(i)$ in this sequence is created by removing a $4k$-independent set from its predecessor and constructing the $\theta_k$-graph on the remaining vertices. We continue this process until only three vertices are left, and denote this graph by $\theta_k(h)$. Similar as with the DK-graph, we mark each vertex $v \in \theta_k K(P)$ with a level, denoted by $\text{level}(v)$. This level refers to the $\theta_k$-graph $\theta_k(i)$ in which the vertex last occurs, i.e., if $v$ last occurs in $\theta_k(i)$, then $\text{level}(v) = i$. Recall that for the construction of the DK-graph, we simply merged all of the graphs in the sequence. For the construction of the $\theta_k$-graph we only select the outgoing edges incident to a vertex $v$ in the graph $\theta_k(\text{level}(v))$. Since each vertex can only be deleted once, and has at most $k$ outgoing edges in $\theta_k(\text{level}(v))$, we can immediately derive the size of the $\theta_k$-K-graph.

**Lemma 4.2.1.** A $\theta_k$-Kirkpatrick graph on point set $P$ of size $n$, has at most $kn$ edges.

Furthermore we note that Observation 3.2.2 about the DK-graph in Section 3.2 holds in a similar way for the $\theta_k$-Kirkpatrick graph.

**Observation 4.2.2.** Consider some vertex $v$ of level $i$ in a $\theta_k$-Kirkpatrick graph. All vertices adjacent to $v$ at $\theta_k(i)$ have level $j > i$.

![Figure 4.2](image-url)  
**Figure 4.2:** Construction of a $\theta_k$-K-graph: the bold edges in $\theta_k(0), \ldots, \theta_k(4)$ define the edge set of $\theta_k K(P)$.

Let us consider some $\theta_k$-K-graph $\theta_k K(P)$. In Theorem 3.1.2 we showed that a DK-graph has at most $\log_2 n$ levels. We prove a similar bound on the number of levels in the $\theta_k$-K-graph. To select a $D$-independent set at every graph in the sequence, we use Algorithm 1, presented in Section 3.1. Note that we are not able to use the selection technique presented by Biedl and Wilkinson [3] since it only works for planar graphs. Using Algorithm 1 we are able to bound the minimum size of the $4k$-independent we can select in any $\theta_k$-graph. From this lemma we then derive the number of levels $h$ in the $\theta_k$-Kirkpatrick graph.
Lemma 4.2.3. In every \( \theta_k \)-graph we can select a 4k-independent set of size at least \( \frac{n}{8k} \).

Proof. Consider some \( \theta_k \)-graph \( G \). Recall that every vertex of \( G \) has outdegree at most \( k \), and hence the total number of edges \( E \) is at most \( kn \). From this we derive that the total degree of \( G \) is \( \sum_{v \in G} \deg(v) = 2E \leq 2kn \). We argue that this means there are at least \( \frac{n}{k} \) vertices of degree less than 4k in \( G \). Assume for the sake of contradiction that there are more than \( \frac{n}{k} \) vertices with degree at least 4k. Since \( G \) is connected, every vertex has degree at least 1. When we compute the total degree of \( G \) in this case, we find that it is at least \( 4k \frac{n}{k} + \frac{n}{k} = 2kn + \frac{n}{k} \). This contradicts the fact that the total degree of \( G \) is at most \( 2kn \). Hence there are at least \( \frac{n}{k} \) vertices of degree less than 4k.

Consider what happens when we run Algorithm 1 on \( G \). Set \( S \) has size at least \( \frac{n}{k} \) when entering the while-loop in line 2. At every iteration of the while-loop, we remove at most 4k vertices, i.e., the vertex \( v \) that is added to \( I_D \), and its at most 4k−1 neighbors. This means we can add at least \( \frac{k}{2k} = \frac{n}{8k} \) vertices to \( I_D \) before there are no vertices left in \( S \). Hence, using Algorithm 1 we can select a 4k-independent set of size at least \( \frac{n}{8k} \) from any \( \theta_k \)-graph. \( \square \)

Theorem 4.2.4. A \( \theta_k \)-Kirkpatrick graph on point set \( P \) of size \( n \) has at most \( h = \frac{1}{\log(\frac{8k}{8k-1})} \log n \) levels.

Proof. Lemma 4.2.3 states that we can always select a 4k-independent set of size at least \( \frac{n}{8k} \) in any \( \theta_k \)-graph. This means that the number of vertices \( n(i) \) in graph \( \theta_k(i) \), satisfies \( n(i) \leq \frac{8k-1}{8k} n(i-1) \). From this we derive that:

\[
3 = n(h) \leq \left( \frac{8k-1}{8k} \right)^h \cdot n
\]

Hence,

\[
\left( \frac{8k}{8k-1} \right)^h \geq \frac{n}{3}
\]

and so

\[
h \leq \log \frac{8k}{8k-1} \cdot \frac{n}{3} \leq \frac{1}{\log \left( \frac{8k}{8k-1} \right)} \log n
\]

Hence, we need at most \( \frac{1}{\log\left( \frac{8k}{8k-1} \right)} \log n \) levels in order to reduce \( \theta_k(0) \) to \( \theta_k(h) \). \( \square \)

To give an impression of the size of this constant we note that for \( k = 7 \) we have 38.5 log \( n \), for \( k = 8 \) we have 44 log \( n \), for \( k = 9 \) we have 49.6 log \( n \), for \( k = 10 \) we have 55.1 log \( n \), and so on.

4.3 Spanner properties of the \( \theta_k \)-Kirkpatrick Graph

In this section we determine the spanning ratio and hop-diameter of the \( \theta_k \)-Kirkpatrick graph. The algorithm for finding the short path between two vertices in a \( \theta_k \)K-graph is similar to the routing algorithm for \( \theta_k \)-graphs presented by Ruppert and Seidel [20]. We provide a detailed description of the \( \theta_k \)K-graph short path construction algorithm in the next paragraph, and refer to Algorithm 3 for the pseudocode of this method.

Similar as Algorithm 2, this algorithm can be described as two walks: one from \( u \) to \( w \), and one from \( w \) to \( u \). However, instead of collecting triangles intersecting \( uw \), the walks in Algorithm 3 collect one edge at each step and store it in the sets \( U \) and \( W \). We again collect the edges at a specific \( \theta_k \)-graph \( \theta_k(i) \), and are finished when the two walks meet. Together, the edges in \( U \cup W \) form a short path from \( u \) to \( w \).

Let us consider a step of the walk from \( u \) to \( w \) in more detail (a step of the walk from \( w \) to \( u \) is analogous). Throughout the execution of the algorithm we keep track of the vertex \( v_u \) that is the last vertex of the partial path \( u = v_0, v_1, ..., v_u \) in \( U \). We refer to \( v_u \) as the frontier vertex of the walk from \( u \) to \( w \). Initially \( v_u = u \). At every iteration of the while-loop in line 4 we determine
which of the two walks to proceed next, by checking which of the two frontier vertices $v_u$ and $v_w$ has the lowest level. Here we consider the walk from $u$ to $w$ at an arbitrary iteration of the while loop, and therefore assume $\text{level}(v_u) < \text{level}(v_w)$. Consider the example scenario in Figure 4.3. Here $\text{level}(v_u) = 56 < \text{level}(v_w) = 72$. We denote the frontier vertex of lowest level at a specific iteration of the while-loop by $v$. Hence, $v = v_u$ in Figure 4.3. To determine what edge to add to $U$, we look at the $\theta_k$-graph $\theta_k (\text{level}(v))$, i.e., the $\theta_k$-graph in which $v$ last occurs. We first check if $vv_w$ is an edge in this graph. If it is, we add $vv_w$ to $U$ and break out of the while loop, terminating the algorithm. Since we perform this check at the start of every iteration of the while-loop, we are certain that Algorithm 3 terminates. If $vv_w$ does not exist, we consider the cone $C^v$ of $v$ that contains $v_w$ (indicated in blue in Figure 4.3). We then find the vertex $z \in C^v$ closest to $v$, and add the edge $vz$ to $U$. Finally we update $v_u$ and start a new iteration of the while-loop. We continue this process until the path is complete.

To prove the spanning ratio of the $\theta_k$-Kirkpatrick graph, we provide a proof based on the theorem by Ruppert and Seidel [20] on the spanning ratio of $\theta_\pi$-graphs. This theorem shows that when we construct a path between two vertices $v$ and $w$ in a $\theta_\pi$-graph, using an approach similar to that of Algorithm 3, then this path has length at most $t \cdot d(v, w)$, where $t = \frac{1}{1 - 2 \sin (\theta)}$. In particular, they show that when we add an edge $uv$ to such a path from $v$ to $w$, it holds that $|uv| + t \cdot d(v, w) \leq t \cdot d(u, w)$, where $|uv|$ denotes the length of edge $uv$.

**Theorem 4.3.1.** The $\theta_k$-Kirkpatrick graph, where $k \geq 7$, is a $\frac{1}{1 - 2 \sin (\frac{\pi}{k})}$-spanner.

**Proof.** Suppose we run Algorithm 3 on a $\theta_k$-Kirkpatrick graph $\theta_k K(P)$, where $k \geq 7$, and two arbitrary vertices $u, w \in \theta_k K(P)$. Define $l(u, w)$ as the length of the path from $u$ to $w$ defined by the edges in $U \cup W$ after termination of the algorithm. We show that $l(u, w) \leq t \cdot d(u, w)$, where $t = \frac{1}{1 - 2 \sin (\frac{\pi}{k})}$, by applying induction on the number $i$ of edges in $U \cup W$.

**BASE $i = 1$** If $U \cup W$ contains only one edge, then $l(u, w) = |uv| = d(u, w)$, and the hypothesis is trivially true.

**STEP $i > 1$** For the step case we assume that when Algorithm 3 returns a path of $i > 1$ edges, then $l(u, w) \leq t \cdot d(u, w)$. Now suppose the algorithm returns a path of $i + 1$ edges, and assume without loss of generality that the first step that was taken, was from $u$ to vertex $v$. Then by the induction hypothesis, and the proof of Ruppert and Seidel [20] we derive:

$$l(u, w) = |uv| + l(v, w) \leq |uv| + t \cdot d(v, w) \leq t \cdot d(u, w)$$

thereby completing the step case.

Since the above holds for any two vertices $u, w \in \theta_k K(P)$, we conclude the $\theta_k$-Kirkpatrick graph, where $k \geq 7$, is a $\frac{1}{1 - 2 \sin (\frac{\pi}{k})}$-spanner.

It is left to show that $\theta_k$-Kirkpatrick graph has bounded hop-diameter. We do this by the following theorem.
CHAPTER 4. THE $\theta_K$-KIRKPATRICK GRAPH

Algorithm 3 $\theta_k$K-graph short path construction

Input: $\theta_k$-Kirpatrick graph $\theta_kK(P)$, and two vertices $u, w \in \theta_kK(P)$

Output: Set of $O(\log n)$ edges, defining a path between $u$ and $w$ of length at most $\frac{1}{1 - 2\sin(\frac{\pi}{k})}|uw|$. 

1: $v_u \leftarrow u$ ; $v_w \leftarrow w$
2: $v \leftarrow \text{null}$ ; $e \leftarrow \text{null}$
3: $U \leftarrow \emptyset$ ; $W \leftarrow \emptyset$
4: while $v_u \neq v_w$ do
5: if $\text{level}(v_u) \leq \text{level}(v_w)$ then
6: $v \leftarrow v_u$
7: else
8: $v \leftarrow v_w$
9: end if
10: consider $\theta_k(\text{level}(v))$ of $\theta_kK(P)$
11: if $v_u v_w$ is an edge in $\theta_k(\text{level}(v))$ then
12: $U \leftarrow U \cup v_u v_w$
13: break
14: end if
15: if $v == v_u$ then
16: $e \leftarrow \text{edge } vz$, where $z$ is the closest vertex in cone $C_v$, such that $v_w \in C_v$
17: $U \leftarrow U \cup e$
18: $v_u \leftarrow z$
19: else
20: $e \leftarrow \text{edge } vz$, where $z$ is the closest vertex in cone $C_v$, such that $v_u \in C_v$
21: $W \leftarrow W \cup e$
22: $v_w \leftarrow z$
23: end if
24: end while
25: return $U \cup W$

Theorem 4.3.2. The $\theta_k$-Kirpatrick graph, where $k \geq 7$, has hop-diameter $2h = \frac{2}{\log(\frac{2}{2\sin(\frac{\pi}{k})})} \log n$.

Proof. In Theorem 4.3.1 we showed that using Algorithm 3 we can construct a short path between any two vertices $u$ an $w$ in a $\theta_k$-K-graph $\theta_kK(P)$. Here we show that this short path consists of at most $2h$ edges. First observe that with every step of either one of the two walks in Algorithm 3, the level of the frontier vertex of that walk must increase. This means that at every level of $\theta_kK(P)$, both walks add at most one edge to the short path. By Theorem 4.2.4 we know that $\theta_kK(P)$ has at most $h = \frac{1}{\log(\frac{2}{2\sin(\frac{\pi}{k})})} \log n$ levels. Hence the short path constructed by Algorithm 3 consists of at most $2h$ edges. Since this holds for any two vertices $u, w \in \theta_kK(P)$, the $\theta_k$-Kirpatrick graph, where $k \geq 7$, has hop-diameter $2h = \frac{2}{\log(\frac{2}{2\sin(\frac{\pi}{k})})} \log n$. \hfill \qed

From Lemma 4.2.1, and Theorems 4.3.1 and 4.3.2, we finally derive the complete definition of the spanner properties of the $\theta_k$-Kirpatrick graph.

Corollary 4.3.2.1. The $\theta_k$-Kirpatrick graph $\theta_kK(P)$, where $k \geq 7$, is a $\frac{1}{1 - 2\sin(\frac{\pi}{k})}$-spanner of size $kn$ and hop-diameter at most $\frac{2}{\log(\frac{2}{2\sin(\frac{\pi}{k})})} \log n$. 

Geometric Spanner Networks 21
Chapter 5

The constrained $\theta_5$-graph

In this chapter, we attempt to bound the spanning ratio of the constrained $\theta_5$-graph. We try to do this by generalizing the approach used by Bose et al. [7] to bound the spanning ratio of the $\theta_5$-graph in the unconstrained setting. Unfortunately the inductive proof breaks down for the constrained $\theta_5$-graph, and no successful alternative method was found for proving or disproving that the constrained $\theta_5$-graph is a spanner. Before we discuss the attempted proof in Section 5.2, we introduce constrained $\theta_k$-graphs in Section 5.1. Section 5.3 provides several geometric lemmas that we refer to throughout Section 5.2.

5.1 Constrained $\theta_k$-graphs

Recall that in Section 4.1 we defined $\theta_k$-graphs. Here, we consider the scenario in which a set $S$ of non-intersecting line segments is introduced to this graph. The endpoints of all segments in $S$ must be in the point set $P$ on which the $\theta_k$-graph is defined. We say that two vertices $u, w \in P$ can see each other when there is no constraint intersecting the line $uw$, or when the line segment $uw$ itself is a constraint. Instead of saying that $u$ and $w$ can see each other, we also say that $w$ is visible to $u$, or that $uw$ is a visibility edge. Note that when $uw$ is a visibility edge, it does not imply that $uw$ is an edge in the $\theta_k$-graph. We let the visibility graph $\text{Vis}(P,S)$ be the graph defined by the point set $P$, and all visibility edges with respect to the set of constraints $S$. In other words, $\text{Vis}(P,S)$ is the complete graph of $P$ minus the edges that intersect one or more constraints in $S$.

With the introduction of constraints, we redefine the closest vertex in a cone $C^u_\theta$. We say that the closest vertex in cone $C^u_\theta$ of a constrained $\theta_k$-graph is the vertex whose orthogonal projection on the bisector of $C^u_\theta$ lies closest to the apex $u$, and is visible to $u$. Hence, even when the orthogonal projection of some vertex $v$ lies closest to $u$, $v$ is not the closest vertex to $u$ when some constraint $c$ prevents $u$ from seeing $v$ (see Figure 5.1).

One of the first to address $\theta_k$-graphs in the constrained setting was Clarkson [12]. He did this within the context of motion planning and showed how to construct a $(1 + \varepsilon)$-spanner of $\text{Vis}(P,S)$, of size linear in $n$. Das [13] later showed that the visibility graph contains a bounded-degree spanner with constant spanning ratio, and that this graph can be constructed in $O(n \log n)$.
5.2 Spanning ratio of the constrained $\theta_5$-graph

In this section we attempt to prove that the constrained $\theta_5$-graph is a spanner of the visibility graph, by finding a path between any pair of vertices in the graph that can see each other. Unfortunately the approach presented in this section breaks down at a certain point. We nevertheless start the proof, explain why it breaks down, and discuss some alternative methods.

Lemma 5.2.1. For every two vertices $u$ and $w$ in the constrained $\theta_5$-graph that can see each other, there exists a path connecting $u$ and $w$ of length at most $c \cdot |\triangle uw|$, for some constant $c > 1$.

Proof. Consider two arbitrary vertices $u$ and $w$ in a constrained $\theta_5$-graph $G$. To prove that there exists a path between $u$ and $w$ of length at most $c \cdot |\triangle uw|$, we perform induction on the rank of $|\triangle uw|$. The rank of $|\triangle uw|$ is determined by taking the canonical triangles of all pairs of vertices that can see each other in $G$, and ordering them by ascending size.

[BASE] If the rank of $|\triangle uw|$ is 1, then there is no pair of vertices that can see each other with a canonical triangle smaller than $|\triangle uw|$. This means there is an edge between $u$ and $w$ in the constrained $\theta_5$-graph, which completes the base case of the proof.

[STEP] If the rank of $|\triangle uw|$ is greater than 1, then we apply induction using the following hypothesis: between every pair of vertices that can see each other and whose canonical triangle $|\triangle|$ is smaller than $|\triangle uw|$, there exists a path of length at most $c \cdot |\triangle|$. For simplicity we only consider the case where $w$ lies in the right half of $C_0^w$. All other cases are analogues. We define $a$ and $b$ as the left, respectively right corner of $\triangle uw$, we define $m$ as the midpoint of the side of $\triangle uw$ opposite to $u$, and define $x$ as the intersection of $mb$ with the bisector of $\angle mab$ (see Figure 5.2). Since we only consider the case where $w$ lies to the right of $m$, we conclude that $u \in C_3^w$. Furthermore we note that if $w$ lies to the left of $x$, then by basic trigonometry we derive that $|\triangle uw| < |\triangle uw|$, which means we can apply induction on $|\triangle uw|$. Hence, for the remainder of this proof we concern ourselves with the case where $w$ lies between $x$ and $b$.

If $w$ would be the vertex closest to $u$ in $C_0^w$, or $u$ would be the vertex closest to $w$ in $C_3^w$, then $u$ and $w$ are connected by an edge and we obtain a case similar to that of the base case. Let us therefore assume that this is not the case, and denote $v_w$ to be the vertex that lies closest to $w$ in $C_3^w$. We distinguish four different cases based on the position of $v_w$: $v_w \in C_2^w$, $v_w \in C_1^w$, $v_w \in C_0^w$ and $v_w \in C_4^w$ (see Figure 5.3).
CHAPTER 5. THE CONSTRAINED $\theta_5$-GRAPH

For the remainder of this proof we use the following general approach to show that we can apply induction in each of the four cases. We aim to find a path of length $g \cdot |\Delta_{uw}|$, such that we are left with a convex chain of visibility edges of which the sum of the sizes of their canonical triangles is strictly smaller than $|\Delta_{uw}|$. Let us say that this sum is $h \cdot |\Delta_{uw}|$, where $h < 1$, then for the convex chain $v_0, v_1, ..., v_j$ it should hold that $\sum_{i=1}^{j} |\Delta_{v_{i-1}v_i}| < h \cdot |\Delta_{uw}|$. By applying induction, we derive to following expression we need to satisfy:

$$
\begin{align*}
g \cdot |\Delta_{uw}| + c \cdot h \cdot |\Delta_{uw}| &\leq c \cdot |\Delta_{uw}| \\
g + c \cdot h &\leq c \\
g \cdot \frac{h}{h-1} &\leq c
\end{align*}
$$

(1)

**Case 1** $v_w \in C^w_2$

Let us assume $v_w$ is in some general position in $C^w_2$. By definition, $\Delta_{uwv_w}$ is empty, and $uw$ and $wv_w$ are visibility edges. Recall that this means no constraints can cross $uw$ or $wv_w$. We consider the path from $u$ to $v_w$. The induction hypothesis states that there must be a path $u = v_0, v_1, ..., v_j = v_w$ of visibility edges, such that $\sum_{i=1}^{j} |\Delta_{v_{i-1}v_i}| < h \cdot |\Delta_{uw}|$. If there are no constraints intersecting $uv_w$, then $uv_w$ is a visibility edge, and the proof is similar to that of the unconstrained $\theta_5$-graph by Bose et al. [7]. Hence assume $uv_w$ is not a visibility edge, and define $y$ as the intersection of $uw$ with the side of $\Delta_{wv_w}$ opposite of $w$. By Lemma 5.3.2 in Section 5.3, and the fact that $\Delta_{wv_w}$ is empty, we know that the endpoints of the constraints intersecting $uv_w$ must lie within $uyv_w$ (see Figure 5.4).

![Figure 5.4: Case 1: $v_w \in C^w_2$. The convex chain from $u$ to $v_w$ must lie within the orange area.](image)

![Figure 5.5: We increase $|uy|$ and $|yv_w|$ by moving both $w$ and $v_w$ to the right.](image)

Let us rewrite expression (1). By Lemma 5.3.4 we derive that for the convex path $u = u_0, u_1, ..., u_j = v_w$ of visibility edges from $u$ to $w$, it holds that $\sum_{i=1}^{j} |\Delta_{u_{i-1}u_i}| \leq |\Delta_{uy}^*| + |\Delta_{yv_w}^*|$. Here $|\Delta_{uy}^*|$ and $|\Delta_{yv_w}^*|$ denote the size of the smallest canonical triangle between $u$ and $y$, respectively $y$ and $v_w$. By Lemma 5.3.3 it holds that $|\Delta_{uy}^*| = \frac{\cos(X_{uy})}{\cos(-\frac{\pi}{5})}|uy|$, where

$$X_{uy} = \frac{\pi}{5} - \left| (\text{slope}(uy) \mod \frac{\pi}{5}) - \frac{\pi}{10} \right|,$$

and $\text{slope}(uy)$ denotes the slope of the line through $u$ and $y$.^1 Finally we observe that $yv_w$ has a maximum slope (see Definition 5). This allows us to rewrite expression (1) as follows:

---

^1We use both $|\Delta_{uy}^*|$ and $\text{slope}(uw)$ throughout this thesis as described here.
CHAPTER 5. THE CONSTRAINED $\theta_5$-GRAPH

\[ g \cdot |\Delta_{uw}| + c \cdot h \cdot |\Delta_{uw}| \leq c \cdot |\Delta_{uw}| \]

\[ |wv_w| + c \cdot \sum_{i=1}^{j} |\Delta_{ui-1}u_i| \leq c \cdot |\Delta_{uw}| \]

\[ |wv_w| + c \cdot \left( |\Delta_{uy}| + |\Delta_{uy'}| \right) \leq c \cdot |\Delta_{uw}| \]

\[ |wv_w| + c \cdot \left( \frac{\cos(X_{uw})}{\cos(\frac{\pi}{5})} \cdot |uy| + \frac{\cos(\frac{\pi}{7})}{\cos(\frac{\pi}{5})} \cdot |yu_w| \right) \leq c \cdot |\Delta_{uw}| \]

(2)

To determine the situation in which this expression is maximized, we assume $v_w$ is in some general position in $C^w_2$. We observe that we can increase both $|wv_w|$ and $|yu_w|$, while not changing $|uy|$ and $X_{uy}$, by moving $v_w$ along the boundary of $\Delta_{wv_w}$ opposite of $w$, until it is positioned in the bottom corner of $\Delta_{wv_w}$.

Once $v_w$ lies in the bottom right corner of $\Delta_{wv_w}$, we can increase $|uy|$ and $|yu_w|$, while not changing $|wv_w|$ by moving both $w$ and $v_w$ to the right, i.e., we keep $v_w$ in the bottom corner of $\Delta_{wv_w}$ (see Figure 5.5). However, the slope of $uy$ changes in such a way that $\cos(X_{uw})$ decreases. When modeling this situation with the mathematics software package GeoGebra [15], we observe that this decrease does not outweigh the increase in $|uy|$ and $|yu_w|$. Unfortunately no formal proof was devised to show this. We do however reason that $|uy|$ and $|yu_w|$ increase. Consider the triangle $wv'u'$, and translate $w$ and $v_w$ to the right in order to obtain the new triangle $w'v'u'$ (see Figure 5.5). Note that $|w'v'u'| = |wv_w|$, $\angle v'u'w' = \angle yv_w$, and $\angle w'v'u'w > \angle uvw$. From this we conclude that $|yv'u'| > |yu_w|$. Furthermore note that $\text{slope}(yv'u') = \text{slope}(yu_w)$ and $\angle w'ub < \angle uvb$, and therefore $|uy'| > |uy|$.

From these observations we conclude that expression (2) is maximized when $w$ lies on $b$, and $v_w$ lies on the right boundary of $\Delta_{uw}$. Assume $v_w$ is positioned a distance $g \cdot |\Delta_{uw}|$ from $w$, where $g > 0$. Since $w$ lies on $b$, $uy$ has a minimum slope and therefore $|\Delta_{uy}| = |uy|$. For simplicity, we define $\phi = \frac{\pi}{10}$ for the remainder of this proof, and derive the following:

\[ |wv_w| + c \cdot (|\Delta_{uy}| + |\Delta_{yu_w}|) \leq c \cdot |\Delta_{uw}| \]

\[ |wv_w| + c \cdot (|uy| + \frac{\cos(\phi)}{\cos(2\phi)} \cdot |yu_w|) \leq c \cdot |\Delta_{uw}| \]

\[ g \cdot |\Delta_{uw}| + c \cdot \left( |\Delta_{uw}| - g \cdot \cos(2\phi) |\Delta_{uw}| + g \cdot \frac{\cos(\phi)}{\cos(2\phi)} \sin(2\phi) |\Delta_{uw}| \right) \leq c \cdot |\Delta_{uw}| \]

(3)

\[ g + c \cdot \left( 1 - g \cdot \cos(2\phi) + g \cdot \cos(\phi) \tan(2\phi) \right) \leq c \]

\[ c - c \cdot \left( 1 - g \cdot \cos(2\phi) + g \cdot \cos(\phi) \tan(2\phi) \right) \geq g \]

\[ c \cdot \left( 1 - \left( 1 - g \cdot \cos(2\phi) + g \cdot \cos(\phi) \tan(2\phi) \right) \right) \geq g \]

\[ \frac{g}{g \cdot \cos(2\phi) - g \cdot \cos(\phi) \tan(2\phi)} \leq c \]

\[ 2(2 + \sqrt{5}) \leq c \]

Hence we conclude that the induction hypothesis holds for case 1 for $c \geq 2(2 + \sqrt{5}) \approx 8.472$.

**Case 2** $v_w \in C^w_1$

For case 2 we follow the same line of reasoning as for case 1. $uw$ and $wv_w$ are again visibility edges, $\Delta_{wv_w}$ is empty by definition, and by Lemma 5.3.2 there must be a convex path of
Case 3 \( v_w \in C_0^u \)

Case 3 is the case in which the induction hypothesis breaks down. First note that if \( w \) lies on \( b \), this case is analogous to case 2. In case 3 however, expression (2) is maximized when \( w \) lies on \( wv \) and \(uw\) lies on the upper boundary of \( \triangle uwv \) (see Figure 5.6). Unfortunately no formal proof was found to show this. These worst case positions were again obtained by modeling the situation using the mathematics software package GeoGebra [15]. We observe that because \( w \) lies on \( x \), both \( yw \) and \( uy \) have a maximum slope. This allows us to rewrite expression (2) into:

\[
|uwv| + c \cdot \frac{\cos(\phi)}{\cos(2\phi)} \left( |uy| + |yw| \right) < c \cdot |\triangle uw|
\]

If we again assume that \( v_w \) is located a distance \( g \cdot |\triangle uw| \) from \( w \), where \( g > 0 \), then by basic trigonometry we have that \( |uy| = (1-g)\frac{\cos(2\phi)}{\cos(\phi)}|\triangle uw| \) and \( |yw| = g \left( \sin(2\phi) + \sin(\phi)\frac{\cos(2\phi)}{\cos(\phi)} \right) \cdot |\triangle uw| \). This allows us to derive the total path length from \( u \) to \( w \) as follows:

\[
g \cdot |\triangle uw| + c \cdot \frac{\cos(\phi)}{\cos(2\phi)} \left( 1 - g \right) \frac{\cos(2\phi)}{\cos(\phi)} |\triangle uw| + g \left( \sin(2\phi) + \sin(\phi)\frac{\cos(2\phi)}{\cos(\phi)} \right) \cdot |\triangle uw| < c \cdot |\triangle uw|
\]

\[
g + c \cdot \frac{\cos(\phi)}{\cos(2\phi)} \left( 1 - g \right) \frac{\cos(2\phi)}{\cos(\phi)} + g \left( \sin(2\phi) + \sin(\phi)\frac{\cos(2\phi)}{\cos(\phi)} \right) < c
\]

\[
c - c \left( 1 - g + g \cdot \frac{\cos(\phi)}{\cos(2\phi)} \sin(2\phi) + g \cdot \sin(\phi) \right) < c
\]

\[
c - c \left( g - g \cdot \frac{\cos(\phi)}{\cos(2\phi)} \sin(2\phi) - g \cdot \sin(\phi) \right) > g
\]

\[
c \cdot g \left( 1 - \frac{\cos(\phi)}{\cos(2\phi)} \sin(2\phi) - \sin(\phi) \right) > g
\]

Since we stated that \( g > 0 \), we conclude that the induction hypothesis breaks down for case 3 of our proof. Since the induction hypothesis breaks down in a similar way for case 4, we need to adopt a different strategy for these last two cases. We discuss some of the alternative approaches that were considered in the next paragraph.

Alternative methods. Since the induction hypothesis used for the attempted proof of Lemma 5.2.1 breaks down when \( v_w \in C_0^u \), alternative methods for proving the spanning ratio of the constrained \( \theta_5 \)-graph were considered. First note that since the induction hypothesis still holds
when \( w \) lies on \( b \), we only need to concern ourselves with the case in which \( v_w \) lies in the region highlighted in Figure 5.7. Bose et al. [7] encounter a similar problem in case 4 of their proof of the spanning ratio of the \( \theta_5 \)-graph in the unconstrained setting. They resolve this problem by considering the position of the vertex \( v_u \) that is closest to \( u \) in \( C_u^b \). They make a case distinction based on what cone of \( v_u \), \( w \) lies in. Unfortunately this approach does not work for the constrained \( \theta_5 \)-graph. Consider the situation depicted in Figure 5.8. Here it holds that \( |v_u y_u| = |u y_u| \), and \( X_{y y} = X_{y y y} = X_{v y w} \), all having a maximum slope. From this we conclude that we can rewrite expression (2) into

\[
|u v_u| + |w v_w| + c \cdot \frac{\cos(\phi)}{\cos(2\phi)} \cdot (|v_u y_u| + |y_u y_w| + |v_w y_w|) < c \cdot |\Delta_{uw}|
\]

Here the left-hand side of the inequality is even greater than in case 3, while the right-hand side remains unchanged. Therefore the inequality neither holds in this case.

An entirely different approach that was considered is applying induction on the number of constraints in the \( \theta_5 \)-graph. For the base case, where there are no constraints, Bose et al. [7] proved that the graph is indeed a spanner. For the step case we assume the constrained \( \theta_5 \)-graph is a spanner for \( i \) constraints, and prove it still is for either \( i + 1 \), or \( i - 1 \) constraints. Though this approach was not fully explored, we note that when we add a constraint, we may intersect short paths in the \( \theta_5 \)-graph with \( i \) constraints. This means we need to redirect edges, causing an increase in the path length. We are however not certain by how much such a path length may increase. We reason though that the shortest detour an edge \( uw \) intersected by the constraint \( xy \) may have to make, is either through the path \( u, x, w \) or \( u, y, w \). Hence there is still potential in applying this approach, but further investigation is necessary.
5.3 Geometric Lemmas

This section contains several geometric lemmas used in the attempted proof in Section 5.2.

**Lemma 5.3.1.** Let $u, v$ and $w$ be vertices in a $\theta_k$-graph. If $v \in C_{0}^u$, $w \in C_{1}^v$ and $w \in C_{0}^v$ for some constant $i$, then $|\triangle uw| = |\triangle uv| + |\triangle vw|$.

**Proof.** Consider the vertices $u, v$ and $w$ in some $\theta_k$-graph. Without loss of generality we assume $v \in C_{0}^u$, $w \in C_{1}^v$ and $w \in C_{0}^v$. Let $p$ be the top right corner of $|\triangle uw|$, $q$ be the top right corner of $|\triangle uv|$, and $r$ be the top right corner of $|\triangle vw|$ (see Figure 5.9). Because the vertices are positioned in cones of equal number, the canonical triangles $\triangle uw$, $\triangle vw$ and $\triangle uv$ all have the same orientation. Hence the polygon $vpqr$ is a parallelogram, and $pq = vr$. We conclude the proof by deriving:

$$|\triangle uw| = uq = up + pq = up + vr = |\triangle uv| + |\triangle vw|$$

\[\square\]

![Figure 5.9: Three vertices of a $\theta_k$-graph such that $v \in C_{0}^u$, $w \in C_{1}^v$ and $w \in C_{0}^v$.](image)

![Figure 5.10: The convex chain of visibility edges from $u$ to $w$.](image)

**Lemma 5.3.2.** Let $u, v, w$ be three arbitrary points in the plane such that $uw$ and $vw$ are visibility edges, and $w$ is not the endpoint of a constraint that intersects the interior of the triangle $uvw$. There exists a convex chain of visibility edges from $u$ to $v$ inside triangle $uvw$, such that the polygon defined by $uw$, $vw$ and the convex chain is empty, and does not contain any constraints.

**Proof.** Let $Q$ be the set of vertices inside triangle $uvw$. If $Q$ is empty, we define the convex chain from $u$ to $v$ as the edge $uv$. If $Q$ is not empty, we obtain the convex chain by taking the convex hull of $Q \cup \{u, v\}$, and removing the edge $uv$. We note that the edge $uw$ is always present in the convex hull of $Q \cup \{u, v\}$, because the vertices of $Q$ all lie within the triangle $uvw$, and thus all lie to one side of the line through $u$ and $v$.

We now prove that the obtained convex chain is indeed a chain of visibility edges. Let $x$ and $y$ be two arbitrary, consecutive vertices on the convex chain. We consider the line through $x$ and $y$, and its intersection points $u'$ and $v'$ with the edges $uw$ and $vw$ (see Figure 5.10). For sake of contradiction, assume that there is a constraint $e$ crossing $xy$. Then one of its endpoints lies to the same side of the line through $x$ and $y$, as $w$ is lying. This means there are three possible locations this endpoint $e$ may be: (1) $e$ lies within triangle $u'v'w$; (2) $e = w$; or (3) $e$ lies outside triangle $u'v'w$. In the first case we derive a contradiction, because during the construction of the convex hull, $e$ would have been included in the hull, and $x$ and $y$ would not have been consecutive vertices on the convex chain. The second case is not possible, since we specifically stated that $w$ is not the endpoint of a constraint that intersects the interior of triangle $uvw$. For the third location we derive a contradiction because $e$ would have to intersect either $u'w$ or $v'w$. This is not possible, since $uw$ and $vw$ are visibility edges, and hence by construction, $u'w$ and $v'w$ are visibility edges as well.

Since the above holds for every pair of consecutive vertices $x$ and $y$ on convex chain, the polygon defined by $uw$, $vw$ and the convex chain must be empty, and not contain any constraints. \[\square\]
Lemma 5.3.3. Let $u$ and $w$ be two vertices in a $\theta_5$-graph, and define $|\triangle_{uw}^\ast|$ as the size of the smallest canonical triangle between $u$ and $w$. Then $|\triangle_{uw}^\ast| = \frac{\cos(X_{uw})}{\cos(2\phi)} |uw|$, where $\phi = \frac{\pi}{10}$ and $X_{uw} = 2\phi - \left|\text{slope}(uw) \mod 2\phi\right| - \phi$.

Proof. Let $u$ and $w$ be two vertices in a $\theta_5$-graph, and consider the situation where the slope of $uw$, denoted $\text{slope}(uw)$, is $-\frac{\pi}{10}$. This means that $u \in C^4_w$ and $w \in C^1_u$ (see Figure 5.11). More precisely, $w$ lies on the border of $C^1_u$, and therefore $|\triangle_{uw}| = |uw|$. Since this is the smallest size a canonical triangle can have, we conclude that if $\text{slope}(uw) = -\frac{\pi}{10}$, then $|\triangle_{uw}^\ast|$ is minimized.

Next we consider the situation where $\text{slope}(uw) = 0$ (see Figure 5.12). In this case it holds that $|\triangle_{uw}| = |\triangle_{wu}|$, and $|\triangle_{uw}^\ast|$ is maximized.

We observe that the situations described above reoccur after every rotation of $uw$ of $\frac{\pi}{5}$. More formally, if $\text{slope}(uw) = \frac{\pi}{10} + c \cdot \frac{\pi}{5}$, where $c \in \{-2, -1, ..., 2\}$, then $|\triangle_{uw}| = |uw|$. Similarly, if $\text{slope}(uw) = 0 + c \cdot \frac{\pi}{5}$, where $c \in \{-2, -1, ..., 2\}$, then $|\triangle_{uw}| = |\triangle_{uw}|$. In the latter case we derive by basic trigonometry that $|\triangle_{uw}| = |\triangle_{wu}| = \frac{\cos(\phi)}{\cos(2\phi)} |uw|$, where $\phi = \frac{\pi}{10}$.

In order to express the behavior of $|\triangle_{uw}^\ast|$, we define the function $X_{uw}(\text{slope}(uw))$. We want this function to have a range from $\phi$ to $2\phi$, and linearly alternate between these two values for every change in $\text{slope}(uw)$ of $\phi$. Formally we can define this function as a triangle wave of amplitude $\frac{\pi}{10}$ and period $\frac{\pi}{5}$ (see Figure 5.13). This triangle wave can be described using either one of the following two expressions:

\[
2\phi - \left|\alpha \mod 2\phi\right| - \phi
\]

\[
\frac{1}{10} \left(\cos^{-1} \left(\cos \left(10\alpha \right)\right) + \pi\right)
\]

Using this function we define $|\triangle_{uw}^\ast| = \frac{\cos(X_{uw})}{\cos(2\phi)} |uw|$.

From here on forth we use the following definitions to state that the slope between two vertices maximizes, respectively minimizes the smallest canonical triangle between them.
**Definition 5.** Let $u$ and $w$ be two vertices in a $\theta_5$-graph. We say that the edge $uw$ has a maximum slope, or that its slope is maximized, when it has a value of $0 + c \cdot \frac{\pi}{5}$ for $c \in \{-2, -1, \ldots, 2\}$.

**Definition 6.** Let $u$ and $w$ be two vertices in a $\theta_5$-graph. We say that the edge $uw$ has a minimum slope, or that its slope is minimized, when it has a value of $\pi + c \cdot \frac{\pi}{5}$ for $c \in \{-2, -1, \ldots, 2\}$.

**Lemma 5.3.4.** Let $u$ and $w$ be two vertices in a $\theta_5$-graph. For any triangle $uvw$, of which either $uv$ or $vw$ has a maximum slope, it holds that $|\triangle^*_uw| \leq |\triangle^*_uv| + |\triangle^*_vw|$. 

**Proof.** We note that this proof is not fully complete, since the approach adopted here is rather cumbersome. Through analysis using the computational software program Mathematica [21], we are however positive that this proof can be completed.

Let $u$ and $w$ be two vertices in a $\theta_5$-graph. Let us consider some arbitrary point $v$, forming the triangle $uvw$. We assume without loss of generality that $vw$ is the side of triangle $uvw$ that has a maximum slope. We define $\alpha$ as $\angle uwv$ and define $\beta$ as the angle between $uv$ and the extension of $wv$ (see Figure 5.14). In order to prove that $|\triangle^*_uw| \leq |\triangle^*_uv| + |\triangle^*_vw|$, we express the size of the canonical triangles of $uw$, $uv$ and $vw$ in terms of the length of these edges. Let $\phi = \frac{\pi}{10}$. Since we assumed $vw$ has a maximum slope, we conclude that $|\triangle^*_vw| = \cos(\phi) \cdot \cos(2\phi) \cdot |vw|$. For the other two edges we use Lemma 5.3.3 to rewrite the inequality into:

$$\frac{\cos(X_\alpha)}{\cos(2\phi)} \cdot |uw| \leq \frac{\cos(X_\beta)}{\cos(2\phi)} \cdot |uw| + \frac{\cos(\phi)}{\cos(2\phi)} \cdot |vw|$$

where

$$X_\alpha = 2\phi - |(\alpha \mod 2\phi) - \phi| = \frac{1}{10} \left( \cos^{-1}(\cos(10\alpha)) + \pi \right)$$

![Figure 5.14: Triangle $uvw$ and the angles $\alpha$ and $\beta$.](image)

$X_\beta$ is defined in a similar way as $X_\alpha$. Note that compared to the expression in Lemma 5.3.3, we replaced the slope of the edge by the angle between the edge and $vw$. We are able to do this by the definition of $\alpha$ and $\beta$, the fact that $vw$ has a maximum slope, and the fact that a slope of 0 is a maximum slope. Next we use the law of sines, to derive that:

$$\frac{|uw|}{\sin(\alpha)} = \frac{|uv|}{\sin(\pi - \beta)} \left( = \frac{|uv|}{\sin(\beta)} \right) = \frac{|vw|}{\sin(\beta - \alpha)}$$

Note that since $0 \leq \beta \leq \pi$, the value of $\sin(\beta)$ is independent of whether this angle is acute or
CHAPTER 5. THE CONSTRAINED \( \theta_5 \)-GRAPH

obtuse. Using these observations, we are able to simplify the inequality as follows:

\[
\begin{align*}
|\Delta_{uw}^*| &\leq |\Delta_{uw}^*| + |\Delta_{uw}^*| \\
\cos(X_\alpha) &\cdot |uw| \leq \cos(X_\beta) \cdot |uw| + \frac{\cos(\phi)}{\cos(2\phi)} \cdot |uw| \\
\cos(X_\alpha) &\cdot |uw| \leq \cos(X_\beta) \cdot \sin(\alpha) + \frac{\cos(\phi)}{\cos(2\phi)} \cdot |uw| + \frac{\cos(\phi)}{\sin(\beta)} \cdot |uw| \\
\cos(X_\alpha) &\leq \cos(X_\beta) \cdot \sin(\alpha) + \frac{\cos(\phi)}{\sin(\beta)} \cdot |uw| \\
\cos(X_\alpha) &\leq \cos(X_\beta) \cdot \sin(\alpha) + \cos(\phi) \cdot \sin(\beta - \alpha) \\
\cos(X_\alpha) \cdot \sin(\beta) &\leq \cos(X_\beta) \cdot \sin(\alpha) + \cos(\phi) \cdot \sin(\beta - \alpha) \\
0 &\leq \cos(X_\beta) \cdot \sin(\alpha) + \cos(\phi) \cdot \sin(\beta - \alpha) - \cos(X_\alpha) \cdot \sin(\beta)
\end{align*}
\]

(4)

Note that in this derivation we are able to multiply both sides of the expression by \( \sin(\beta) \), because \( 0 \leq \beta \leq \frac{\pi}{2} \), ensuring that \( \sin(\beta) \geq 0 \). We now prove that inequality (4) holds for \( 0 \leq \alpha, \beta \leq \pi \), where \( \alpha < \beta \). Since the expression of \( X_\alpha \) and \( X_\beta \) is rather complex, we split the domains of \( \alpha \) and \( \beta \) into ten equal subdomains of size \( \phi = \frac{\pi}{10} \). Together this gives rise to 55 domains that we need to prove. We define \( \text{dom}(\alpha) \) and \( \text{dom}(\beta) \) as the subdomain that \( \alpha \), respectively \( \beta \), is in. We start with the ten domains where \( \text{dom}(\alpha) = \text{dom}(\beta) \). Since several of these derivations are rather extensive, they are collected in Appendix B. We use the trigonometric identities in Appendix A to prove the derivations in Appendix B. Table 5.1 gives an overview of the solutions of the right-hand side of inequality (4) for each of the ten domains. The second column shows the expression of \( X_\alpha \), similarly \( X_\beta \).

<table>
<thead>
<tr>
<th>Domain</th>
<th>( X_\alpha; X_\beta )</th>
<th>Solution of right-hand side of expression (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>([0, \phi])</td>
<td>( \alpha + \phi )</td>
<td>0</td>
</tr>
<tr>
<td>((\phi, 2\phi])</td>
<td>( -\alpha + 3\phi )</td>
<td>( 4 \cos(\phi) \sin^2(\phi) \sin(\beta - \alpha) )</td>
</tr>
<tr>
<td>((2\phi, 3\phi])</td>
<td>( \alpha - \phi )</td>
<td>0</td>
</tr>
<tr>
<td>((3\phi, 4\phi])</td>
<td>( -\alpha + 5\phi )</td>
<td>( 4 \cos(\phi) (1 + 2 \cos(2\phi)) \sin^2(\phi) \sin(\beta - \alpha) )</td>
</tr>
<tr>
<td>((4\phi, 5\phi])</td>
<td>( \alpha - 3\phi )</td>
<td>( 4 \cos(\phi) \sin^2(\phi) \sin(\beta - \alpha) )</td>
</tr>
<tr>
<td>((5\phi, 6\phi])</td>
<td>( -\alpha + 7\phi )</td>
<td>( 8 \cos(\phi) (1 + \cos(2\phi) + \cos(4\phi)) \sin^2(\phi) \sin(\beta - \alpha) )</td>
</tr>
<tr>
<td>((6\phi, 7\phi])</td>
<td>( \alpha - 5\phi )</td>
<td>( 4 \cos(\phi) (1 + 2 \cos(2\phi)) \sin^2(\phi) \sin(\beta - \alpha) )</td>
</tr>
<tr>
<td>((7\phi, 8\phi])</td>
<td>( -\alpha + 9\phi )</td>
<td>( 8 \cos(\phi) (1 + 2 \cos(2\phi)) \sin^2(\phi) \sin(\beta - \alpha) )</td>
</tr>
<tr>
<td>((8\phi, 9\phi])</td>
<td>( \alpha - 7\phi )</td>
<td>( 8 \cos(\phi) (1 + \cos(2\phi) + \cos(4\phi)) \sin^2(\phi) \sin(\beta - \alpha) )</td>
</tr>
<tr>
<td>((9\phi, \pi])</td>
<td>( -\alpha + 11\phi )</td>
<td>( 4 \cos(\phi) (3 + 2 \cos(2\phi) + 2 \cos(4\phi)) \sin^2(\phi) \sin(\beta - \alpha) )</td>
</tr>
</tbody>
</table>

Table 5.1: Overview of the solutions of the right-hand side of expression (4) for the ten domains where \( \text{dom}(\alpha) = \text{dom}(\beta) \).

From Table 5.1 we conclude that all of the solutions consist of a positive constant, multiplied by \( \sin(\beta - \alpha) \). This means that all of these expressions are at least zero for \( \alpha < \beta \). Since by definition it holds that \( \alpha < \beta \), we conclude that inequality (4) holds for these ten domains. To prove that the inequality holds for the domains where \( \text{dom}(\alpha) < \text{dom}(\beta) \), we calculate the partial derivative in the \( \beta \)-direction of the right-hand side of inequality (4). If we can show that this
derivative is positive for $\alpha < \beta$, then we have proved that the inequality holds for all domains.

\[
\frac{\partial \beta}{\partial \alpha} (\cos(X_\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(X_\alpha) \sin(\beta))
\]
\[
= \frac{\partial \beta}{\partial \alpha} \left( \cos(X_\beta) \sin(\alpha) + \cos(\phi) \left( \sin(\beta) \cos(\alpha) - \cos(\beta) \sin(\alpha) \right) - \cos(X_\alpha) \sin(\beta) \right)
\]
\[
= - \sin(X_\beta) \sin(\alpha) \frac{\partial \beta}{\partial \alpha} (X_\beta) + \cos(\phi) \left( \cos(\beta) \cos(\alpha) + \sin(\beta) \sin(\alpha) \right) - \cos(X_\alpha) \cos(\beta)
\]
\[
= - \sin(X_\beta) \sin(\alpha) \frac{\partial \beta}{\partial \alpha} \left( \frac{\pi}{10} + \frac{1}{10} \cos^{-1} \left( \cos(10\beta) \right) \right) + \cos(\phi) \cos(\beta - \alpha) - \cos(X_\alpha) \cos(\beta)
\]
\[
= - \sin(X_\beta) \sin(\alpha) \frac{\sin(10\beta)}{\sqrt{1 - \cos^2(10\beta)}} + \cos(\phi) \cos(\beta - \alpha) - \cos(X_\alpha) \cos(\beta)
\]
\[
= - \sin(X_\beta) \sin(\alpha) \frac{\sin(10\beta)}{\sqrt{1 - \cos^2(10\beta)}} + \cos(\phi) \cos(\beta - \alpha) - \cos(X_\alpha) \cos(\beta)
\]

We observe that this derivative is a summation of three addends. To determine whether its outcome is positive, we evaluate these three addends separately. For the first of these three addends we observe that the fraction $\frac{\sin(10\beta)}{\sqrt{1 - \cos^2(10\beta)}}$ defines a square wave that alternates between 1 and $-1$ for every interval of size $\frac{\pi}{10}$. Since $\sin(X_\beta)$ is a positive constant, the addend is positive only when $\sin(\alpha)$ has the opposite sign as the fraction. In the second addend $\cos(\phi)$ is a positive constant. Hence $\cos(\beta - \alpha)$ needs to be positive as well in order to make the entire addend positive. This certainly happens when $0 \leq \beta - \alpha \leq \frac{\pi}{2}$. For the third addend it holds that $\cos(X_\alpha)$ is always positive. Hence $\cos(\beta)$ has to have a negative sign in order to make the entire addend positive. This is true when $\frac{\pi}{2} < \beta \leq \pi$.

From these observations we are certain that the derivative is positive for some of the remaining domains. However, no conclusions can be drawn from this on whether the right-hand side of the inequality is positive as well. The approach for proving that inequality (4) holds for the remaining domains can either be done by proving the derivative is positive, or in a similar way used for the domains where $\text{dom}(\alpha) = \text{dom}(\beta)$. For now we assume that the following holds for all cases where $0 \leq \alpha, \beta \leq \pi$, and $\alpha < \beta$:

$$|\Delta^*_{u,v_i+1} - |\Delta^*_{u,v_i}| + |\Delta^*_{v_i,v_{i+1}}|$$

Lemma 5.3.5. Consider two vertices $u$ and $w$ in a constrained $\theta_5$-graph, and the convex path $u = u_0, u_1, ..., u_j = w$ of visibility edges between them. If all edges on the path lie within a triangle $uvw$, of which either $uw$ or $vw$ has a maximum slope, then $\sum_{i=1}^j |\Delta^*_{u_{i-1}u_i}| \leq |\Delta^*_{uw}| + |\Delta^*_{vw}|$

Proof. Let us consider some convex path $u = u_0, u_1, ..., u_j = w$ of visibility edges from $u$ to $w$ within a triangle $uvw$. We assume without loss of generality that $uw$ is the side of triangle $uvw$ that has a maximum slope. For some edge $u_{i-1}u_i$ on the convex path, we define $v_i$ as the intersection of the line extending this edge and $vw$ (see Figure 5.15). To prove that the lemma holds we apply induction on the number $m$ of edges on the convex path that lie strictly inside of $uvw$. We define the induction hypothesis as follows: $\sum_{i=1}^j |\Delta^*_{u_{i-1}u_i}| \leq |\Delta^*_{uw}| + |\Delta^*_{vw}|$.

[BASE $m = 0$] For the base case we assume that there are no edges that lie strictly inside of $uvw$. This means that all of the edges of the convex path from $u$ to $w$ either lie on $uw$ and $vw$, or on $uw$. In the first case, we assume $v_k = v$, and use Lemma 5.3.1 to derive that:

$$\sum_{i=1}^j |\Delta^*_{u_{i-1}u_i}| = \sum_{i=1}^k |\Delta^*_{u_{i-1}u_i}| + \sum_{i=k+1}^j |\Delta^*_{u_{i-1}u_i}| = |\Delta^*_{uw}| + |\Delta^*_{vw}|$$

In the latter case we use Lemmas 5.3.1 and 5.3.4 to derive that:

$$\sum_{i=1}^j |\Delta^*_{u_{i-1}u_i+1}| = |\Delta^*_{uw}| \leq |\Delta^*_{uw}| + |\Delta^*_{vw}|$$
CHAPTER 5. THE CONSTRAINED $\theta_5$-GRAPH

Figure 5.15: The convex path from $u$ to $w$ within triangle $uvw$. $u_m u_{m+1}$ is the last edge to lie strictly inside of $uvw$.

This proves the induction hypothesis for the base case.

[STEP $m > 0$] For the step case we assume that the induction hypothesis holds when there are $m$ edges on the convex path that lie strictly within $uvw$. We prove that this implies the hypothesis holds as well when there are $m + 1$ edges that strictly lie inside of $uvw$. Let us define the edge $u_m u_{m+1}$ to be the last edge that strictly lies inside of $uvw$ (see Figure 5.15). We derive that:

\[ \sum_{i=1}^{j} |\Delta^*_{u_{i-1}u_i}| = \sum_{i=1}^{m} |\Delta^*_{u_{i-1}u_i}| + |\Delta^*_{u_m u_{m+1}}| + \sum_{i=m+2}^{j} |\Delta^*_{u_{i-1}u_i}| \]

Using Lemma 5.3.4 we derive:

\[ \leq \sum_{i=1}^{m} |\Delta^*_{u_{i-1}u_i}| + |\Delta^*_{u_m v_m}| + |\Delta^*_{v_m u_{m+1}}| + \sum_{i=m+2}^{j} |\Delta^*_{u_{i-1}u_i}| \]

\[ \leq \sum_{i=1}^{m} |\Delta^*_{u_{i-1}u_i}| + |\Delta^*_{u_m v_m}| + \sum_{i=m+1}^{m-1} |\Delta^*_{u_{i-1}u_i}| \]

\[ = \sum_{i=1}^{m-1} |\Delta^*_{u_{i-1}u_i}| + |\Delta^*_{u_{m-1}u_m}| + |\Delta^*_{u_m v_m}| + \sum_{i=m+1}^{j} |\Delta^*_{u_{i-1}u_i}| \]

Using Lemma 5.3.1 we get:

\[ = \sum_{i=1}^{m-1} |\Delta^*_{u_{i-1}u_i}| + |\Delta^*_{u_{m-1}u_m}| + \sum_{i=m+1}^{j} |\Delta^*_{u_{i-1}u_i}| \]

This reduces the number of edges that strictly lie within $uvw$ from $m + 1$ back to $m$, while ensuring that $\sum_{i=1}^{j} |\Delta^*_{u_{i-1}u_i}|$ does not increase. We thereby prove the step case, and subsequently the inductive proof. \qed
Chapter 6

Conclusions

In Chapter 3 of this thesis we introduced a new kind of graph called the Delaunay-Kirkpatrick graph. We showed that this graph is a 1.998-spanner of size at most $8n - 6$ and hop-diameter at most $53.2 \log n$. One of the main advantages of this graph compared to other spanners with small diameter, is its fairly simple construction method, based on the Kirkpatrick [18] hierarchical data structure.

In Chapter 4 we presented an alternative version of the DK-graph, based on $\theta_k$-graphs, where $k \geq 7$. This graph is called the $\theta_k$-Kirkpatrick graph and is a $\frac{1}{1-2\sin(\frac{\pi}{k})}$-spanner of size at most $kn$ and hop-diameter at most $g(k) \cdot \log n$, where $g(k) = \frac{2}{\log(\frac{8k}{8k-1})}$. Unfortunately the function $g(k)$ returns a rather large constant. The size of this constant is heavily influenced by the minimum size of the $D$-independent set we are able to select in a $\theta_k$-graph. It may therefore be interesting to investigate whether a more clever method can be devised for selecting $D$-independent sets in $\theta_k$-graphs.

In Chapter 5 we attempted to prove that the constrained $\theta_5$-graph is a spanner of the visibility graph. We did this by generalizing the approach used by Bose et al. [7] to show that the $\theta_5$-graph is a spanner in the unconstrained setting. Unfortunately it appeared that this approach cannot be easily generalized. This makes it difficult to predict whether the spanning ratio in the constrained setting is equal to that in the unconstrained setting, or whether the graph is a spanner at all in the constrained setting. Since Bose et al. [9] showed that for all $\theta_k$-graphs where $k > 5$ those graphs are indeed spanners, we conjecture that the constrained $\theta_5$-graph is a spanner.
Bibliography


BIBLIOGRAPHY


[22] Ge Xia. The stretch factor of the delaunay triangulation is less than 1.998. SIAM Journal on Computing, 42(4):1620–1659, 2013. 3, 4, 6, 14

Appendix A

Trigonometric Identities

\[
\begin{align*}
\sin^2(x) &= \frac{1}{2} (1 - \cos(2x)) \\
\cos^2(x) &= \frac{1}{2} (1 + \cos(2x)) \\
\sin^4(x) &= \frac{1}{8} (3 - 4 \cos(2x) + \cos(4x)) \\
\cos^4(x) &= \frac{1}{8} (3 + 4 \cos(2x) + \cos(4x)) \\
\sin^2(x) \cos^2(x) &= \frac{1}{8} (1 - \cos(4x)) \\
\sin^6(x) &= \frac{1}{32} (10 - 15 \cos(2x) + 6 \cos(4x) - \cos(6x)) \\
\cos^6(x) &= \frac{1}{32} (10 + 15 \cos(2x) + 6 \cos(4x) + \cos(6x)) \\
\sin^2(x) \cos^4(x) &= \frac{1}{32} (2 + \cos(2x) - 2 \cos(4x) - \cos(6x)) \\
\sin^4(x) \cos^2(x) &= \frac{1}{32} (2 - \cos(2x) - 2 \cos(4x) + \cos(6x)) \\
\sin^8(x) &= \frac{1}{128} (35 - 56 \cos(2x) + 28 \cos(4x) - 8 \cos(6x) + \cos(8x)) \\
\cos^8(x) &= \frac{1}{128} (35 + 56 \cos(2x) + 28 \cos(4x) + 8 \cos(6x) + \cos(8x)) \\
\sin^2(x) \cos^6(x) &= \frac{1}{128} (5 + 4 \cos(2x) - 4 \cos(4x) - 4 \cos(6x) - \cos(8x)) \\
\sin^6(x) \cos^2(x) &= \frac{1}{128} (5 - 4 \cos(2x) - 4 \cos(4x) + 4 \cos(6x) - \cos(8x)) \\
\sin^4(x) \cos^4(x) &= \frac{1}{128} (3 - 4 \cos(4x) + \cos(8x)) \\
\end{align*}
\]

\[
\begin{align*}
1 - \cos^4(x) &= \frac{1}{2} \sin^2(x) (3 + \cos(2x)) \\
1 - \cos^6(x) &= \frac{1}{8} \sin^2(x) (15 + 8 \cos(2x) + \cos(4x)) \\
1 - \cos^8(x) &= \frac{1}{32} \sin^2(x) (70 + 47 \cos(2x) + 10 \cos(4x) + \cos(6x)) \\
1 - \cos^{10}(x) &= \frac{1}{128} \sin^2(x) (315 + 244 \cos(2x) + 68 \cos(4x) + 12 \cos(6x) + \cos(8x)) \\
\end{align*}
\]
Appendix B
Derivations of Lemma 5.3.4

\[0 < \alpha, \beta \leq \frac{\pi}{10}\]

\[0 \leq \cos(X_\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(X_\alpha) \sin(\beta)
= \cos(\beta + \alpha) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(\alpha + \alpha) \sin(\beta)
= (\cos(\beta) \cos(\phi) - \sin(\beta) \sin(\phi)) \sin(\alpha) + \cos(\phi) (\sin(\beta) \cos(\alpha) - \cos(\beta) \sin(\alpha))
- (\cos(\alpha) \cos(\phi) - \sin(\alpha) \sin(\phi)) \sin(\beta)
= \cos(\beta) \cos(\phi) \sin(\alpha) - \sin(\beta) \sin(\phi) \sin(\alpha) + \cos(\phi) \sin(\beta) \cos(\alpha) - \cos(\phi) \cos(\beta) \sin(\alpha)
- \cos(\alpha) \cos(\phi) \sin(\beta) + \sin(\alpha) \sin(\phi) \sin(\beta)
= 0\]

\[\frac{\pi}{10} < \alpha, \beta \leq \frac{\pi}{5}\]

\[0 \leq \cos(X_\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(X_\alpha) \sin(\beta)
= \cos(-\beta + 3\phi) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(-\alpha + 3\phi) \sin(\beta)
= \cos(3\phi) \cos(\beta) \sin(\alpha) + \sin(3\phi) \sin(\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha)
- \cos(3\phi) \cos(\alpha) \sin(\beta) - \sin(3\phi) \sin(\alpha) \sin(\beta)
= \cos(\phi) \sin(\beta - \alpha) + \cos(3\phi) \cos(\beta) \sin(\alpha) - \cos(3\phi) \cos(\alpha) \sin(\beta)
= (\cos(\phi) - \cos(3\phi)) \sin(\beta - \alpha)
= \left(\cos(\phi) - \cos(\phi) \left(\cos^2(\phi) - 3 \sin^2(\phi)\right)\right) \sin(\beta - \alpha)
= \cos(\phi) \left(1 - \cos^2(\phi) + 3 \sin^2(\phi)\right) \sin(\beta - \alpha)
= \cos(\phi) \left(\sin^2(\phi) + 3 \sin^2(\phi)\right) \sin(\beta - \alpha)
= 4 \cos(\phi) \sin^2(\phi) \sin(\beta - \alpha)\]

\[\frac{\pi}{5} < \alpha, \beta \leq \frac{3\pi}{10}\]

\[0 \leq \cos(X_\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(X_\alpha) \sin(\beta)
= \cos(\beta - \alpha) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(\alpha - \alpha) \sin(\beta)
= \cos(\beta) \cos(\phi) \sin(\alpha) + \sin(\beta) \sin(\phi) \sin(\alpha) + \cos(\phi) \sin(\beta) \cos(\alpha) - \cos(\phi) \cos(\beta) \sin(\alpha)
- \cos(\alpha) \cos(\phi) \sin(\beta) - \sin(\alpha) \sin(\phi) \sin(\beta)
= 0\]
\[ \frac{\pi}{10} < \alpha, \beta \leq \frac{2\pi}{5} \]

\[
0 \leq \cos(X_\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(X_\alpha) \sin(\beta) \\
= \cos(-\beta + 5\phi) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(-\alpha + 5\phi) \sin(\beta) \\
= \cos(5\phi) \cos(\beta) \sin(\alpha) + \sin(5\phi) \sin(\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) \\
- \cos(5\phi) \cos(\alpha) \sin(\beta) - \sin(5\phi) \sin(\alpha) \sin(\beta) \\
= \cos(5\phi) \cos(\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(5\phi) \cos(\alpha) \sin(\beta) \\
= - \cos(5\phi) \sin(\beta - \alpha) + \cos(\phi) \sin(\beta - \alpha) \\
= (\cos(\phi) - \cos(5\phi)) \sin(\beta - \alpha) \\
= \left( \cos(\phi) - \cos(\phi) \left( \cos^4(\phi) - 10 \sin^2(\phi) \cos^2(\phi) + 5 \sin^4(\phi) \right) \right) \sin(\beta - \alpha) \\
= \cos(\phi) \left( 1 - \cos^4(\phi) + 10 \sin^2(\phi) \cos^2(\phi) - 5 \sin^4(\phi) \right) \sin(\beta - \alpha) \\
= \cos(\phi) \left( \frac{1}{2} \sin^2(\phi) \left( 3 + \cos(2\phi) \right) + 10 \sin^2(\phi) \cos^2(\phi) - 5 \sin^4(\phi) \right) \sin(\beta - \alpha) \\
= \cos(\phi) \left( \frac{1}{2} (3 + \cos(2\phi)) + 10 \sin^2(\phi) \cos^2(\phi) - 5 \sin^2(\phi) \right) \sin^2(\phi) \sin(\beta - \alpha) \\
= \cos(\phi) \left( \frac{1}{2} (13 + 11 \cos(2\phi)) - 5 \sin^2(\phi) \right) \sin^2(\phi) \sin(\beta - \alpha) \\
= \cos(\phi) \left( \frac{1}{2} (13 + 11 \cos(2\phi)) - \frac{5}{2} \left( 1 + \cos(2\phi) \right) \right) \sin^2(\phi) \sin(\beta - \alpha) \\
= \cos(\phi) \left( \frac{1}{2} (8 + 16 \cos(2\phi)) \right) \sin^2(\phi) \sin(\beta - \alpha) \\
= 4 \cos(\phi) \left( 1 + 2 \cos(2\phi) \right) \sin^2(\phi) \sin(\beta - \alpha)
\]

\[ \frac{2\pi}{5} < \alpha, \beta \leq \frac{\pi}{2} \]

Since \( X_\alpha = \cos(\alpha - 3\phi) = \cos(-\alpha + 3\phi) \) (similarly for \( X_\beta \)), we conclude that the derivation for this domain is analogous to that for the domain \( \left( \frac{\pi}{10}, \frac{\pi}{5} \right) \), resulting in the expression:

\[ 0 \leq 4 \cos(\phi) \sin^2(\phi) \sin(\beta - \alpha) \]

\[ \frac{\pi}{2} < \alpha, \beta \leq \frac{3\pi}{5} \]

\[
0 \leq \cos(X_\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(X_\alpha) \sin(\beta) \\
= \cos(-\beta + 7\phi) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(-\alpha + 7\phi) \sin(\beta) \\
= \cos(7\phi) \cos(\beta) \sin(\alpha) + \sin(7\phi) \sin(\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) \\
- \cos(7\phi) \cos(\alpha) \sin(\beta) - \sin(7\phi) \sin(\alpha) \sin(\beta) \\
= \cos(7\phi) \cos(\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(7\phi) \cos(\alpha) \sin(\beta) \\
= (\cos(\phi) - \cos(7\phi)) \sin(\beta - \alpha) \\
= \ldots (continues on next page)\ldots \]
APPENDIX B. DERIVATIONS OF LEMMA 5.3.4

\[
\begin{align*}
= & \left( \cos(\phi) - \cos(\phi) \left( \cos^6(\phi) - 21 \sin^2(\phi) \cos^4(\phi) + 35 \sin^4(\phi) \cos^2(\phi) - 7 \sin^6(\phi) \right) \right) \\
= & \cos(\phi) \left( 1 - \cos^6(\phi) + 21 \sin^2(\phi) \cos^4(\phi) - 35 \sin^4(\phi) \cos^2(\phi) + 7 \sin^6(\phi) \right) \sin(\beta - \alpha) \\
= & \cos(\phi) \left( \frac{1}{8} \sin^2(\phi) \left( 15 + 8 \cos(2\phi) + \cos(4\phi) \right) + 21 \sin^2(\phi) \cos^4(\phi) - 35 \sin^4(\phi) \cos^2(\phi) \\
& \quad + 7 \sin^6(\phi) \right) \sin(\beta - \alpha) \\
= & \cos(\phi) \left( \frac{1}{8} \left( 78 + 92 \cos(2\phi) + 22 \cos(4\phi) \right) - 35 \sin^2(\phi) \cos^2(\phi) + 7 \sin^4(\phi) \right) \\
& \quad \sin^2(\phi) \sin(\beta - \alpha) \\
= & \cos(\phi) \left( \frac{1}{8} \left( 78 + 92 \cos(2\phi) + 22 \cos(4\phi) \right) - \frac{35}{8} \left( 1 + \cos(4\phi) \right) + 7 \sin^4(\phi) \right) \\
& \quad \sin^2(\phi) \sin(\beta - \alpha) \\
= & \cos(\phi) \left( \frac{1}{8} \left( 43 + 92 \cos(2\phi) + 57 \cos(4\phi) \right) + 7 \sin^4(\phi) \right) \sin^2(\phi) \sin(\beta - \alpha) \\
= & \cos(\phi) \left( \frac{1}{8} \left( 43 + 92 \cos(2\phi) + 57 \cos(4\phi) \right) + \frac{7}{8} \left( 3 - 4 \cos(2\phi) + \cos(4\phi) \right) \right) \\
& \quad \sin^2(\phi) \sin(\beta - \alpha) \\
= & \cos(\phi) \left( \frac{1}{8} \left( 64 + 64 \cos(2\phi) + 64 \cos(4\phi) \right) \right) \sin^2(\phi) \sin(\beta - \alpha) \\
= & 8 \cos(\phi) \left( 1 + \cos(2\phi) + \cos(4\phi) \right) \sin^2(\phi) \sin(\beta - \alpha)
\end{align*}
\]

\[\frac{3\pi}{4} < \alpha, \beta \leq \frac{7\pi}{10}\]

Since \(X_\alpha = \cos(\alpha - 5\phi) = \cos(-\alpha + 5\phi)\) (similarly for \(X_\beta\)), we conclude that the derivation for this domain is analogous to that for the domain \(\left(\frac{3\pi}{10}, \frac{7\pi}{10}\right]\), resulting in the expression:

\[0 \leq 4 \cos(\phi) \left( 1 + 2 \cos(2\phi) \right) \sin^2(\phi) \sin(\beta - \alpha)\]

\[\frac{7\pi}{10} < \alpha, \beta \leq \frac{4\pi}{5}\]

\[0 \leq \cos(X_\phi) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(X_\phi) \sin(\beta)\]

\[= \cos(-\beta + 9\phi) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(-\alpha + 9\phi) \sin(\beta)\]

\[= \cos(9\phi) \cos(\beta) \sin(\alpha) + \sin(9\phi) \sin(\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) \]

\[- \cos(9\phi) \cos(\alpha) \sin(\beta) - \sin(9\phi) \sin(\alpha) \sin(\beta)\]

\[= \cos(9\phi) \cos(\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(9\phi) \cos(\alpha) \sin(\beta)\]

\[= \left( \cos(\phi) - \cos(9\phi) \right) \sin(\beta - \alpha)\]

\[= \ldots(\text{continues on next page})\]
\[
\begin{aligned}
&= \left( \cos(\phi) - \cos(\phi) \left( \cos^8(\phi) - 36 \sin^2(\phi) \cos^6(\phi) + 126 \sin^4(\phi) \cos^4(\phi) - 84 \sin^6(\phi) \cos^2(\phi) \\
&+ 9 \sin^8(\phi) \right) \right) \sin(\beta - \alpha) \\
&= \cos(\phi) \left( 1 - \cos^8(\phi) + 36 \sin^2(\phi) \cos^6(\phi) - 126 \sin^4(\phi) \cos^4(\phi) + 84 \sin^6(\phi) \cos^2(\phi) \\
&- 9 \sin^8(\phi) \right) \sin(\beta - \alpha) \\
&= \cos(\phi) \left( \frac{1}{32} \sin^2(\phi) \left( 70 + 47 \cos(2\phi) + 10 \cos(4\phi) + \cos(6\phi) \right) + 36 \sin^2(\phi) \cos^6(\phi) \\
&- 126 \sin^4(\phi) \cos^4(\phi) + 84 \sin^6(\phi) \cos^2(\phi) - 9 \sin^8(\phi) \right) \sin(\beta - \alpha)
\end{aligned}
\]

Since \( \phi = \frac{\pi}{10} \), it holds that \( \cos(4\phi) = -\cos(6\phi) \). From this we derive

\[
\begin{aligned}
&= \cos(\phi) \left( \frac{1}{32} \sin^2(\phi) \left( 70 + 47 \cos(2\phi) + 9 \cos(4\phi) \right) + 36 \sin^2(\phi) \cos^6(\phi) - 126 \sin^4(\phi) \cos^4(\phi) \\
&+ 84 \sin^6(\phi) \cos^2(\phi) - 9 \sin^8(\phi) \right) \sin(\beta - \alpha) \\
&= \cos(\phi) \left( \frac{1}{32} \left( 70 + 47 \cos(2\phi) + 9 \cos(4\phi) \right) \right) + 36 \sin^2(\phi) \cos^6(\phi) - 126 \sin^4(\phi) \cos^4(\phi) \\
&+ 84 \sin^6(\phi) \cos^2(\phi) - 9 \sin^8(\phi) \right) \sin(\beta - \alpha) \\
&= \cos(\phi) \left( \frac{1}{32} \left( 430 + 587 \cos(2\phi) + 189 \cos(4\phi) \right) \\
&- 126 \sin^2(\phi) \cos^6(\phi) + 84 \sin^4(\phi) \cos^2(\phi) - 9 \sin^6(\phi) \right) \sin(\beta - \alpha) \\
&= \cos(\phi) \left( \frac{1}{32} \left( 430 + 587 \cos(2\phi) + 189 \cos(4\phi) \right) - \frac{126}{32} \left( 2 + \cos(2\phi) - \cos(4\phi) \right) \\
&+ 84 \sin^4(\phi) \cos^2(\phi) - 9 \sin^6(\phi) \right) \sin(\beta - \alpha) \\
&= \cos(\phi) \left( \frac{1}{32} \left( 178 + 461 \cos(2\phi) + 315 \cos(4\phi) \right) + 84 \sin^4(\phi) \cos^2(\phi) - 9 \sin^6(\phi) \right) \sin(\beta - \alpha) \\
&= \cos(\phi) \left( \frac{1}{32} \left( 178 + 461 \cos(2\phi) + 315 \cos(4\phi) \right) + \frac{84}{32} \left( 2 - \cos(2\phi) - 3 \cos(4\phi) \right) - 9 \sin^6(\phi) \right) \sin(\beta - \alpha) \\
&= \cos(\phi) \left( \frac{1}{32} \left( 346 + 377 \cos(2\phi) + 63 \cos(4\phi) \right) - 9 \sin^6(\phi) \right) \sin(\beta - \alpha) \\
&= \cos(\phi) \left( \frac{1}{32} \left( 346 + 377 \cos(2\phi) + 63 \cos(4\phi) \right) - \frac{9}{32} \left( 10 - 15 \cos(2\phi) + 7 \cos(4\phi) \right) \right) \sin(\beta - \alpha) \\
&= \cos(\phi) \left( \frac{1}{32} \left( 256 + 512 \cos(2\phi) \right) \right) \sin(\beta - \alpha) \\
&= 8 \cos(\phi) (1 + \cos(2\phi)) \sin(\beta - \alpha)
\end{aligned}
\]
Appendix B. Derivations of Lemma 5.3.4

\[
\frac{4\pi}{5} < \alpha, \beta \leq \frac{9\pi}{10}
\]

Since \(X_\alpha = \cos(\alpha - 7\phi) = \cos(-\alpha + 7\phi)\) (similarly for \(X_\beta\)), we conclude that the derivation for this domain is analogous to that for the domain \((\frac{\pi}{5}, \frac{3\pi}{5})\), resulting in the expression:

\[
0 \leq 8 \cos(\phi)(1 + \cos(2\phi) + \cos(4\phi)) \sin^2(\phi) \sin(\beta - \alpha)
\]

\[
\frac{9\pi}{10} < \alpha, \beta \leq \pi
\]

Since this derivation is even lengthier that those of case 6 and 8, but follows the exact same pattern, we only provide the first part and final expression. The remainder is left for reader to complement.

\[
0 \leq \cos(X_\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(X_\alpha) \sin(\beta)
\]

\[
= \cos(-\beta + 11\phi) \sin(\alpha) + \cos(\phi) \sin(\beta) - \cos(-\alpha + 11\phi) \sin(\beta)
\]

\[
= \cos(11\phi) \cos(\beta) \sin(\alpha) + \sin(11\phi) \sin(\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha)
- \cos(11\phi) \cos(\beta) \sin(\beta) - \sin(11\phi) \sin(\alpha) \sin(\beta)
\]

\[
= \cos(11\phi) \cos(\beta) \sin(\alpha) + \cos(\phi) \sin(\beta - \alpha) - \cos(11\phi) \cos(\alpha) \sin(\beta)
= (\cos(\phi) - \cos(11\phi)) \sin(\beta - \alpha)
\]

\[
= \left(\cos(\phi) - \cos(\phi) \left(\cos^{10}(\phi) - 55 \sin^2(\phi) \cos^8(\phi) + 330 \sin^4(\phi) \cos^6(\phi) - 462 \sin^6(\phi) \cos^4(\phi)
+ 165 \sin^8(\phi) \cos^2(\phi) - 11 \sin^{10}(\phi)\right)\right) \sin(\beta - \alpha)
\]

\[
= \cos(\phi) \left(1 - \cos^{10}(\phi) + 55 \sin^2(\phi) \cos^8(\phi) - 330 \sin^4(\phi) \cos^6(\phi) + 462 \sin^6(\phi) \cos^4(\phi)
- 165 \sin^8(\phi) \cos^2(\phi) + 11 \sin^{10}(\phi)\right) \sin(\beta - \alpha)
\]

\[
= \cos(\phi) \left(\frac{1}{128} \sin^2(\phi) \left(315 + 244 \cos(2\phi) + 68 \cos(4\phi) + 12 \cos(6\phi) + \cos(8\phi))
+ 55 \sin^2(\phi) \cos^8(\phi) - 330 \sin^4(\phi) \cos^6(\phi) + 462 \sin^6(\phi) \cos^4(\phi)
- 165 \sin^8(\phi) \cos^2(\phi) + 11 \sin^{10}(\phi)\right)\right) \sin(\beta - \alpha)
\]

Since \(\phi = \frac{\pi}{10}\), it holds that \(\cos(6\phi) = -\cos(4\phi)\) and \(\cos(8\phi) = -\cos(2\phi)\), and we derive

\[
= \cos(\phi) \left(\frac{1}{128} \sin^2(\phi) \left(315 + 243 \cos(2\phi) + 56 \cos(4\phi)\right) + 55 \sin^2(\phi) \cos^8(\phi)
- 330 \sin^4(\phi) \cos^6(\phi) + 462 \sin^6(\phi) \cos^4(\phi) - 165 \sin^8(\phi) \cos^2(\phi) + 11 \sin^{10}(\phi)\right) \sin(\beta - \alpha)
\]

\[
= \cos(\phi) \left(\frac{1}{128} \left(315 + 243 \cos(2\phi) + 56 \cos(4\phi)\right) + 55 \cos^8(\phi) - 330 \sin^2(\phi) \cos^6(\phi)
+ 462 \sin^4(\phi) \cos^4(\phi) - 165 \sin^6(\phi) \cos^2(\phi) + 11 \sin^8(\phi)\right) \sin^2(\phi) \sin(\beta - \alpha)
\]

\[
= ... \text{ (left to the reader)}...
\]

\[
= 4 \cos(\phi) (3 + 2 \cos(2\phi) + 2 \cos(4\phi)) \sin^2(\phi) \sin(\beta - \alpha)
\]