Handbook of Approximation Algorithms and Metaheuristics
Contemporary and Emerging Applications
Teofilo F. Gonzalez

Dilation and Detours in Geometric Networks

Publication details
Joachim Gudmundsson, Christian Knauer
Published online on: 23 May 2018

How to cite :- Joachim Gudmundsson, Christian Knauer. 23 May 2018, Dilation and Detours in Geometric Networks from: Handbook of Approximation Algorithms and Metaheuristics, Contemporary and Emerging Applications CRC Press
Accessed on: 01 Oct 2019

PLEASE SCROLL DOWN FOR DOCUMENT

Full terms and conditions of use: https://www.routledgehandbooks.com/legal-notices/terms

This Document PDF may be used for research, teaching and private study purposes. Any substantial or systematic reproductions, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The publisher shall not be liable for an loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
3.1 Introduction

In the current chapter, we consider geometric networks and two important quality measures of such networks; namely, dilation and detour. A geometric network is an undirected graph the vertices of which are points in $d$-dimensional space, and the edges are straight-line segments connecting the vertices. The sites are usually located in the Euclidean plane, but other metrics and higher dimensions are also common. Geometric networks arise in many applications. Road networks, railway networks, telecommunication, pattern matching, bioinformatics—any collection of objects in space that have some connections between them can be modeled as a geometric network.

The weight of an edge $e = (u, v)$ in a geometric network $G = (S, E)$ on a set $S$ of $n$ points is the (usually Euclidean) distance between $u$ and $v$, which we denote by $d(u, v)$. The graph distance $d_G(u, v)$ between two vertices $u, v ∈ S$ is the length of the shortest path in $G$ connecting $u$ to $v$.

The current chapter will consider the problem of designing a “good” network and the dual problem, that is, evaluating how “good” a given network is. When designing a network for a given set $S$ of points, several criteria have to be taken into account. In particular, in many applications it is important to ensure a fast connection between every pair of points in $S$. For this, it would be ideal to have a direct connection between every pair of points; the network would then be a complete graph. In most applications, however, this is unacceptable due to the high costs. This leads to the concepts of dilation and detour of a graph, which we define next.

The dilation or stretch factor of $G$, denoted as $Δ(G)$, is the maximum factor by which graph distance $d_G$ differs from the geometric distance $d$ between every pair of vertices.

**Definition 3.1** Let $S$ be a set of points in $\mathbb{R}^d$, let $M = (\mathbb{R}^d, d_M)$ be a metric space on $\mathbb{R}^d$, and let $G$ be a graph in $M$ with vertex set $S$. For any two vertices $x, y ∈ S$, let $d_G(x, y)$ be the infimum length of all paths connecting $x$ to $y$ in $G$. We call

$$
\Delta(G) = \max_{u, v ∈ S} \frac{d_G(u, v)}{d(u, v)}
$$

the dilation of $G$. If $\Delta(G) ≤ k$, we say that $G$ is a $k$-spanner of $S$. The smallest possible $k$ for which there exists a $k$-spanner of $S$ is called the dilation of $S$ and denoted as $\Delta(S)$.
the M-dilation between x and y in G, and \( \Delta_M(G) = \sup_{x,y \in S} \Delta_M^G(x,y) \) the M-dilation of G.

A graph \( G = (S,E) \) with \( \Delta_M^G(G) \leq t \) is said to be a t-spanner of S.

The notion of dilation can be generalized to arbitrary connected sets \( P \subset \mathbb{R}^d \). This measure is called detour and compares the length of a shortest path inside \( P \) between any two points \( x, y \in P \) with their distance measured, for example, in the Euclidean metric on \( \mathbb{R}^d \).

**Definition 3.2** Let \( P \subset \mathbb{R}^d \) be a connected set, and \( M = (\mathbb{R}^d, d_M) \) be a metric space on \( \mathbb{R}^d \). For any two points, \( x, y \in P \), let \( d_P(x,y) \) be the infimum length of all curves connecting x to y that are contained in \( P \). We call
\[
\delta_M^P(x,y) = \frac{d_P(x,y)}{d_M(x,y)}
\]
the M-detour between x and y in \( P \), and \( \delta_M(P) = \sup_{x,y \in P} \delta_M^P(x,y) \) the M-detour of \( P \).

In some sense, the detour measures how much two metric spaces on \( \mathbb{R}^d \)—namely, \( M \) and \( \mathbb{R}^d \) with the shortest path metric induced by \( P \)—resemble each other.

In the current chapter, we will mainly consider the case \( M = \mathbb{R}^d \), the d-dimensional Euclidean space. We then write \( \Delta(G) \) and \( \delta(P) \) instead of \( \Delta_E(G) \) and \( \delta_E(P) \), respectively. Whenever we speak about the dilation or detour without specifying \( M \), we refer to the case \( M = \mathbb{R}^d \).

The chapter is organized as follows. In Section 3.2, we give an overview of the construction of t-spanners. Section 3.3 briefly considers the dual problem, namely, computing the dilation of a given graph. Then in Section 3.4, we turn our attention to the problem of computing the detour. Finally, in Section 3.5 we end this chapter by looking at structures with small dilation.

### 3.2 Constructing t-Spanners

The problem considered in this section is the construction of t-spanners given a set \( S \) of \( n \) points in \( \mathbb{R}^d \) and a positive real value \( t > 1 \). The aim is to compute a t-spanner for \( S \) with certain desirable properties, such as

- **Size**: Defined to be the number of edges in the graph.
- **Degree**: The maximum number of edges incident to a vertex.
- **Weight**: The weight of a Euclidean network \( G \) is the sum of the edge weights.
- **Spanner diameter (or simply Diameter)**: Defined as the smallest integer \( d \) such that for any pair of vertices \( u \) and \( v \) in \( S \), there is a t-path in the graph (a path of length at most \( t \cdot |uv| \)) between \( u \) and \( v \) containing at most \( d \) edges.

There are trade-offs between different properties, for example, between the degree and the diameter [1]; a graph with constant degree will have diameter \( \Omega(\log n) \). A further example is the trade-off between the diameter and the weight [2], that is, if the diameter of a Euclidean graph \( G \) is bounded by \( O(\log n) \), then the weight of \( G \) is \( \Omega(wt(MST(S)) \cdot \frac{\log n}{\log \log n}) \), where \( wt(MST(S)) \) denotes the weight of the minimum spanning tree of \( S \). Finally, there is also an \( \Omega(n \log n) \) time lower bound in the algebraic computation tree model for computing any t-spanner for a given set of points \( S \) in \( \mathbb{R}^d \) is \( \Omega(n \log n) \) [3].

The most well-known t-spanners can be divided into three groups: \( \Theta \)-graphs, WSPD-graphs, and greedy-graphs. In Sections 3.2.1–3, we give the main idea of each of these, together with the known bounds. Throughout this section, it will be assumed that the set of input points is given in \( \mathbb{R}^d \)-dimensional Euclidean space. For a more detailed description of the construction of t-spanners, see the extensive and thorough work by Narasimhan and Smid [4].
3.2.1 The Θ-Graph

The Θ-graph was discovered independently by Clarkson [5] and Keil [6]. Keil considered the graph in two dimensions, whereas Clarkson extended his construction to also include three dimensions. Althöfer et al. [7] defined the Θ-graph for higher dimensions, and Ruppert and Seidel [8] improved the construction time to $O(n \log^{d-1} n)$. The general approach is stated in the following. Note that it is possible to cover $\mathbb{R}^d$ by $k$ simplicial cones of angular diameter $\theta$, where $k = O(1/\theta^{d-1})$ as defined in the algorithm.

**Algorithm**: $\Theta$-Graph($S$, $t$)

1. Set $k := 2d \left\lceil \sqrt{\frac{2(d-1)}{t \cos \theta}} \right\rceil$ such that $t = \frac{1}{\cos \theta - \sin \theta}$ for $\theta = 2\pi/k$.
2. Set $E := \emptyset$.
3. For each point $u \in S$:
   4. Consider $k$ cones $C_1, \ldots, C_k$ with angular diameter $\theta$ and apex at $u$ that cover $\mathbb{R}^d$.
   5. For each cone $C_i$
      6. Find the point $v$ within $C_i$ whose orthogonal projection onto the bisector of $C_i$ is closest to $u$.
      7. Add $(u, v)$ to $E$.
4. Return $G = (S, E)$.

A similar construction was already defined by Yao [9] in 1982, with the difference that for every point $u$ and every cone $C_i$, $u$ is connected to the closest point in $C_i$. Defining the edges as in the Θ-graph algorithm has the advantage of faster computation.

**Theorem 3.1** The Θ-graph is a $t$-spanner of $S$ for $t = \frac{1}{\cos \theta - \sin \theta}$ with $O(n)$ edges and can be computed in $O(\frac{n}{\theta^{d-1}} \log^{d-1} n)$ time using $O(\frac{n}{\theta^{d-1}} + n \log^{d-2} n)$ space.

Even though the “out-degree” of each vertex is bounded by $k$, the “in-degree” could be linear. Moreover, in worst case, the weight and the diameter of the Θ-graph can be $\Omega(n \cdot wt(MST(S)))$ and $n - 1$, respectively. However, there are several variants of the Θ-graph that improve these bounds.

**Sink-spanners**: To obtain a spanner with constant degree, one can use the construction of sink-spanners by Arya et al. [1]. The basic idea is as follows. Start with a Θ-graph that is a $\sqrt{t}$-spanner. Direct all the edges such that the out-degree is bounded by a constant for every vertex. To handle the vertices with high in-degree, replace each high degree node $q$ and its adjacent neighbors, that is, the star centered at $q$, with a bounded degree $\sqrt{t}$-sink-spanner. A $\sqrt{t}$-sink-spanner is a directed graph in which each point has a directed $\sqrt{t}$-sink-spanner path to the center $q$. This is done in a way that may increase the dilation by a factor of $\sqrt{t}$, thus resulting in a $t$-spanner with degree $O(\frac{1}{(t-1)^2})$.

**Theorem 3.2** The sink spanner is a $t$-spanner of $S$ for $t = \frac{1}{\cos \theta - \sin \theta}$ with $O(n)$ edges and can be computed in $O(\frac{n}{\theta^{d-1}} \log^{d-1} n \frac{n}{\theta^{d-1}})$ time using $O(\frac{n}{\theta^{d-1}} + n \log^{d-2} n)$ space.

The transformation from a directed $\sqrt{t}$-spanner with bounded out-degree to a $t$-spanner with bounded degree is called a sink-spanner transformation.

**Skip-list spanners**: The idea is to generalize skip-lists [10] and apply them to the construction of spanners. Construct a sequence of subsets, as follows: Let $S_1 = S$. Let $i > 1$ and assume that we already have constructed the subset $S_i$. For each point in $S_i$, flip a fair coin. The set $S_{i+1}$ is defined as the set of all points of $S_i$ where the coin flip of which produced heads. The construction stops if $S_{i+1} = \emptyset$. We have $\emptyset = S_{h+1} \subseteq S_h \subseteq S_{h-1} \subseteq \cdots \subseteq S_1 = S$. It holds that $h = O(\log n)$ with high probability and that $\sum_{i=1}^h |S_i| = O(n)$ with high probability. For each $1 \leq i \leq h$, construct a Θ-graph $G(S_i)$. The union of the graphs $G(S_1), \ldots, G(S_h)$ is the skip-list spanner $G$. The skip-list spanner is a $t$-spanner having $O(n)$ edges and $O(\log n)$ spanner diameter with high probability.
Ordered $\Theta$-graphs: A recent modification of the $\Theta$-graph by Bose et al. [11] is the so-called Ordered $\Theta$-graph that considers the order in which the points of $S$ are processed, that is, the graph is built incrementally by inserting and processing each point in some predefined order. When a new point is processed, it only considers the points in the graph that have already been processed. They show an ordering that guarantees that the degree is bounded by $O(k \log n)$ and that a random order gives a $t$-spanner for which the diameter is bounded by $O(\log n)$ with high probability.

Gap-greedy: The final variant is a combination of the $\Theta$-graph and a greedy approach. A set of directed edges is said to satisfy the gap property if the sources of any two edges in the set are separated by a distance that is at least proportional to the length of the shorter of the two edges. Chandra et al. [12] showed that any directed graph $G$ that fulfills the gap property has weight $O(\log n \cdot \text{wt}(\text{MST}(S)))$. However, the gap property is limited in power. Lenhof et al. [13] showed that there exists a graph that satisfies the gap property and has weight $\Omega(\log \log n \cdot \text{wt}(\text{MST}(S)))$.

By using the previous idea, Arya and Smid [14] proposed an algorithm that uses the gap property to decide if an edge should be added to the $t$-spanner graph or not. They consider pairs of points in order of increasing distance, adding an edge $(p, q)$ if and only if it does not violate the gap property.

**Theorem 3.3** Let $t = 1/(\cos \theta - \sin \theta - 2w)$ for some real numbers $0 < \theta < \pi/4$ and $0 \leq w < (\cos \theta - \sin \theta)/2$. The gap-greedy algorithm produces a $t$-spanner $G$ of $S$ in time $O(n/\theta^{d-1} \log^d n)$ such that each vertex has degree $O(1/\theta^{d-1})$ and weight $O\left(\frac{\log n}{\log \log n} \cdot (1 + \frac{1}{\theta}) \log n \cdot \text{wt}(\text{MST}(S))\right)$.

### 3.2.2 The Well-Separated Pair Decomposition Graph

The well-separated pair decomposition (WSPD) was developed by Callahan and Kosaraju [15]. A detailed description of the WSPD can be found in Volume 1, Chapter 4 by Smid in this handbook. The WSPD-graph was first described by Callahan and Kosaraju [16], but similar ideas were used earlier by Salowe [17,18] and Vaidya [19–21].

**WSPD-Graph($S$, $t$)**

1. $E' := \emptyset$
2. $G' := (S, E')$.
3. $\{A_1, B_1\}, \ldots, \{A_m, B_m\} \leftarrow$ the well-separated pair decomposition of $S$ w.r.t. $s = \frac{4(t+1)}{(t-1)}$.
4. **for** each well-separated pair $\{A_i, B_i\}$
   5. Let $a_i$ and $b_i$ be arbitrary points in $A_i$ and $B_i$, respectively.
   6. **Add** $(a_i, b_i)$ to $E'$.
5. **return** $G' = (S, E')$.

The following theorem summarizes the properties.

**Theorem 3.4** The WSPD-graph is a $t$-spanner for $S$ with $O(s^d \cdot n)$ edges and can be constructed in time $O(s^d n + n \log n)$, where $s = 4(t+1)/(t-1)$.

There are modifications that can be made to obtain bounded diameter or bounded degree.

**Bounded diameter:** Arya et al. [22] showed how the construction algorithm can be modified such that the diameter of the graph is bounded by $2 \log n$. In the basic construction of a WSPD-graph, a graph is constructed by adding an edge for every well-separated pair in the WSPD. Instead of selecting an arbitrary point in each well-separated set, they choose a representative point by a search in the fair-split tree (see Vol. 1, Chap. 4), that is, for a node $u$ in the split tree, follow the path down the tree by always choosing the larger subtree. The point stored at the leaf in which the path ends is the representative point for $u$. This approach guarantees that the diameter of the constructed $t$-spanner is bounded by $2 \log n$. 

---

---
Bounded degree: The main problem in the construction of the WSPD-graph is that a single point \( v \) can be part of many well-separated pairs, and each of the pairs generates an edge with an endpoint at \( v \). Arya et al. [1] suggest to keep only the shortest edge for each cone direction, thus combining the \( \Theta \)-graph approach with the WSPD-graph. The resulting spanner has bounded "out-degree" and by applying the sink-spanner transformation, a \( t \)-spanner of degree \( O(n^{2/2d}) \) is obtained.

### 3.2.3 The Greedy-Graph

The greedy algorithm was first presented in 1989 by Bern. Althöfer et al. [7] gave the first theoretical bounds, and since then, the greedy algorithm has been subject to considerable research [12,23–28]. The graph constructed using the greedy algorithm is called a greedy graph, and the general approach is given in the following.

**Algorithm Greedy-Graph**

1. Construct the complete graph of \( S \), denoted by \( G = (S, E) \).
2. \( E' := \emptyset \)
3. \( G' := (S, E') \).
4. for each edge \( (u, v) \in E \) in order of increasing weight
5. \( \text{if shortestPath}(G', u, v) > t \cdot d_G(u, v) \)
6. \( \text{Add } (u, v) \text{ to } E' \).
7. return \( G' = (S, E') \).

Chandra et al. [12] proved that the maximum degree of the graph is bounded by a constant. The running time of the naïve implementation of Greedy-Graph considered in their paper is \( O(n^3 \log n) \). Das and Narasimhan [25] made a breakthrough in 1994 when they showed that a modified greedy graph could be constructed in \( O(n \log^2 n) \) time. They detailed how to use clustering to speed up shortest path queries, by showing that approximate shortest path queries suffice to produce sparse spanners. However, their algorithm was not efficient as the clusters were not maintained efficiently and had to be frequently rebuilt. This problem was solved by Gudmundsson et al. [27] who developed techniques to efficiently perform clustering. The following theorem summarizes the known bounds.

**Theorem 3.5** The greedy graph is a \( t \)-spanner of \( S \) with \( O\left(\frac{n}{(t-1)^d} \log(\frac{1}{t-1})\right) \) edges and \( O\left(\frac{1}{(t-1)^d} \log(\frac{1}{t-1})\right) \) maximum degree and can be computed in time \( O\left(\frac{n}{(t-1)^d} \log^2 n\right) \).

A note on the weight of a \( t \)-spanner: Arya et al. [1] presented an algorithm that claimed to produce \( t \)-spanners of weight \( O(wt(MST(S))) \). However, (Personal communication, 1998) showed that the claimed result was incorrect.

Das and Narasimhan [25] proved that the greedy-graph satisfies the so-called leapfrog property and claimed that any graph satisfying this property has weight \( O(wt(MST(S))) \). At this moment, however, no complete proof of this claim has been published.

### 3.2.4 Experimental Studies

The first experimental study of the construction of \( t \)-spanners was performed by Navarro and Paredes [29] who presented four heuristics for point sets in high-dimensional metric space \( (d = 20) \) and showed by empirical methods that the running time was \( O(n^{2.24}) \), and the number of edges in the produced graphs was \( O(n^{1.13}) \). Sigurd and Zachariasen [30] considered the problem of constructing a minimum weight \( t \)-spanner of a given graph, but they only considered sparse graphs of small size, that is, graphs with at
most 64 vertices and with average vertex degree 4 or 8. In the case in which the input points are given in the Euclidean plane, an extensive study of the main algorithms presented in Sections 3.2.1–3.2.3 was performed by Farshi and Gudmundsson [31].

3.3 Computing the Dilation of a Graph

The previous section considered the problem of constructing a graph for a given point set. In Section 3.2 we considered the problem, that is, given a graph \( G \) compute \( \Delta(G) \).

The problem of calculating the dilation of a given geometric graph can be solved by computing the All-Pairs-Shortest-Path of \( G \). Running Dijkstra’s algorithm—implemented using Fibonacci heaps—gives the dilation of \( G \) in time \( O(mn + n^2 \log n) \) using linear space, where \( n \) is the number of edges in \( G \). For a long time, there were no considerable improvements but in 2002, Langemer et al. [32] and Agarwal et al. [33] showed the first subquadratic bounds for any type of graph. They proved that the dilation of a planar polygonal path can be computed in \( O(n \log n) \) expected time. The algorithm can be generalized to planar trees and cycles, with a randomized expected running time of \( O(n \log^2 n) \), or \( O(n \log^3 n) \) worst case running time. This results holds in the plane and was later extended to three dimensions [34]. Agarwal et al. [34] also showed that in three dimensions, one can compute the dilation of a path, cycle, or tree in \( O(n^{1/3+\epsilon}) \) in randomized expected time. More details about their construction can be found in Section 3.4.2.

Eppstein and Wortman [35] presented an \( O(n \log n) \)-time algorithm for evaluating the dilation when the input graph \( G \) is a star. Computing the shortest path between two points in a star obviously takes constant time, their idea is to identify \( O(n) \) candidate pairs and prove that the pair deciding the dilation of \( G \) is among these pairs. The point pairs are identified using two techniques, each generating \( O(n) \) pairs.

Assume that \((x, y)\) is a pair of points in \( G \) with dilation \( \Delta(G) \).

In the case when the dilation of \( G \) is high, that is, greater than 3, then it holds that \( y \) is one of \( x \)’s \( k \) nearest neighbors, for a constant \( k \). The \( k \) nearest neighbors of every point in \( V \) may be reported in time \( O(kn \log n) \) using the algorithm by Vaidya [20]. So the process of identifying the \( O(n) \) candidate point pairs takes \( O(n \log n) \) time.

In the case when the dilation of \( G \) is low, that is, smaller than or equal to 3, then it holds that \( x \) and \( y \) must have almost the same distance to the center of \( G \). Assume that the vertices of \( G \) are sorted with respect to their distance from the center of \( G \), \( \langle v_1, \ldots, v_n \rangle \), and that \( x = v_i \) and \( y = v_j \). It holds that \( |i – j| \leq \ell \), where \( \ell \) is a constant, and thus identifying \( O(n) \) candidate point pairs requires \( O(n \log n) \) time in this case.

3.3.1 Approximating the Dilation

For general geometric graphs, it seems unavoidable to test all the \( \binom{n}{2} \) pairs of vertices that may decide the dilation of the graph. However, in the case when it suffices to approximate the dilation, this bound is no longer correct. Narasimhan and Smid [36] showed that \( O(n/e^d) \) pairs of vertices are sufficient to test to give a good approximation. Their algorithm is very simple, and it is stated in the following. It is assumed that an algorithm \( \text{ASP}_{c}(p, q, G) \) is given that takes a graph \( G \) and two vertices \( p \) and \( q \) as input and computes a \( c \)-approximation of \( \Delta_G(p, q) \), where \( \Delta_G(p, q) = \frac{d_G(p, q)}{d(p, q)} \). Denote by \( T(n, m, k) \) the running time of \( \text{ASP}_{c} \), when given (i) a graph having \( n \) vertices, \( m \) edges and (ii) a sequence of \( k \) \( c \)-approximate shortest path queries.

The algorithm takes a Euclidean graph \( G = (V, E) \) and a real constant \( \epsilon > 0 \) as input. A WSPD of \( V \) is computed with separation constant \( 4(1 + \epsilon)/\epsilon \). For each well-separated pair \( \{A_i, B_i\} \), two arbitrary points \( a_i \in A_i \) and \( b_i \in B_i \) are chosen, and the dilation of \( a_i \) and \( b_i \) is estimated by taking the ratio between \( \text{ASP}_{c}(a_i, b_i, G) \) and \( d(a_i, b_i) \). The maximum over all the values is returned.

The main theorem can now be stated.
**Theorem 3.6** Let $G$ be a Euclidean graph in $\mathbb{E}^d$ and let $\epsilon$ be a real constant such that $0 < \epsilon \leq 3$, one can compute a $((1 + \epsilon)c)$-approximate dilation of $G$ in time

$$O(n \log n + T(n, m, n/e^d)).$$

By using known data structures to answer (approximate) distance queries together with Theorem 3.6 gives an $O(n \log n)$ time $(1 + \epsilon)$-approximation algorithm for paths, cycles, and trees, an $O(n \sqrt{n})$ time $(1 + \epsilon)$-approximation algorithm for plane graphs [37], an $O(m + n(5/\epsilon)^d (\log n + (t/\epsilon)^d))$ time $(1 + \epsilon)$-approximation algorithm for $t$-spanners [38,39], and an $O(mn^{1/\beta} \log^2 n)$ expected time $O(2\beta(1 + \epsilon)^2)$-approximation algorithm for any Euclidean graph [40].

Note that any algebraic computation tree algorithm that computes a $c$-approximate dilation of a path or a cycle has running time $\Omega(n \log n)$ [36].

## 3.4 Detour

When a geometric network $G$ models an urban street system, the dilation is not necessarily an appropriate measure to assess the quality of $G$. As houses are spread everywhere along the streets, one has to take into account not only the vertices of $G$ but also all the points on its edges, that is, consider the detour of $G$. The detour is also of particular interest in various other applications:

- Analyzing on-line navigation strategies often involves estimating the detour of curves: The length of a path created by some robot must be compared with the shortest path connecting two points [41].
- When comparing the Fréchet distance $F(P, Q)$ between two curves $P, Q$ of detour at most $\kappa$ with their Hausdorff distance $H(P, Q)$ (see Reference 42 for the definition of $F$ and $H$), it turns out that (under some additional technical condition) $F(P, Q) \leq (1 + \kappa)H(P, Q)$, whereas no such bound is known for general curves [43,44].

In Section 3.4.1, we describe properties of pairs of points with maximum detour for various scenarios, and in Section 3.4.2, we give algorithms for computing the detour in these scenarios. All the algorithms exploit the structural properties described earlier. The problem of constructing graphs of small detour that contain a prescribed finite point set will be considered in Section 3.4.3.

### 3.4.1 Structural Properties

We describe properties of pairs of points with maximum detour for the following scenarios:

- $P$ is a simple polygonal curve (possibly closed), or a simple tree in $\mathbb{R}^2$.
- $P$ is a simple polygon in $\mathbb{R}^2$.
- $P$ is a geometric graph in $\mathbb{R}^2$.

As already mentioned, the case of the detour of a planar polygonal chain $P$ is of particular interest in various applications. The problem of (approximately) computing $\delta(P)$ in that setting was first addressed in Ebbers-Baumann et al. [45]. The proposed algorithm exploits several structural properties of the problem.

Consider for instance an edge $e$ of $P$ with endpoints $r, s$, and let $q$ be a fixed point on $P$, such that $d_P(q, s) > d_P(q, r)$, cf. Figure 3.1 on page 60. The function $\delta^P(., q)$ takes on a unique maximum on $e$, and if $\beta := \cos \angle(q, P)$ holds, then this maximum is attained at $p$.

To see this, let us assume that $0 < \beta < \pi$. For $-||p - r|| \leq t \leq 0$, let $p(t)$ be the point on $\overline{pr}$ that has distance $|t|$ to $p$, and for $0 \leq t \leq ||p - s||$ let $p(t)$ be the point on $\overline{ps}$ that has distance $|t|$ to $p$. We have

$$f(t) := \delta^P(p(t), q) = \frac{d_P(p(t), q)}{||p(t) - q||} = \frac{t + d_P(p, q)}{\sqrt{t^2 + ||p - q||^2 - 2t||p - q|| \cos \beta}}$$

with $t = ||p - r||$. One checks that $f(t)$ is decreasing for $0 < t < ||p - r||$, and increasing for $||p - r|| < t < ||p - s||$. Hence $f(t)$ has a unique maximum in the interval $[0, ||p - s||]$. Let $\tau := \frac{||p - r|| + ||p - s||}{2}$, and let $\theta$ be the unique solution of $\cos \theta = \frac{||p - r||}{\tau}$. Then the distance $d_P(p(t), q)$ is of the form $d_P(p(t), q) = d_P(p, q) - ||p(t) - q|| = d_P(p, q) - \tau \min(1, \cos \theta)$, and hence $\delta^P(p(t), q)$ is of the form $-\tau \min(1, \cos \theta)$. If $0 < \beta < \pi/2$, then $0 < \cos \theta < 1$, and hence $\delta^P(p(t), q)$ has a unique maximum $t^*$, which is attained at $p(t^*)$.
Figure 3.1 The detour $\delta^P(p(t), q)$ is larger than $\delta^P(p, q)$.

As $||p - q|| \cos \beta + d_P(p, q) > 0$, the derivative of $f(t)$ has a positive denominator and its numerator has the same sign as

$$n(t) := ||p - q|| \frac{||p - q|| + d_P(p, q) \cos \beta}{||p - q|| \cos \beta + d_P(p, q) - t}.$$

Thus, if $||p - q|| + d_P(p, q) \cos \beta$ is positive (resp. negative), the detour can be increased by moving $p$ toward $s$ (resp. $r$).

Another crucial property is that there is always a pair of points $(p', q')$ attaining the detour of a simple polygonal chain $P$, such that $p'$ and $q'$ are covisible, that is, $p'q' \cap P = \{p', q'\}$. This can be seen as follows: Let $p', q' \in P$ attain the detour of $P$, that is, $\delta^P(p, q) = \delta(P)$, and $p = p_0, \ldots, p_k = q$ be the points of $P$ intersected by the segment $s = pq$, ordered by their appearance on $s$, cf Figure 3.2.

Then

$$\delta^P(p, q) = \frac{d_P(p, q)}{||p - q||} \leq \frac{\sum_{i=0}^{k-1} d_P(p_i, p_{i+1})}{\sum_{i=0}^{k-1} ||p_i - p_{i+1}||} \leq \max_{0 \leq i < k} \frac{d_P(p_i, p_{i+1})}{||p_i - p_{i+1}||} = \max_{0 \leq i < k} \delta^P(p_i, p_{i+1})$$

that is, some covisible pair $p_i$ and $p_{i+1}$ attains the detour of $P$.

We can summarize the previous discussion in the following.

**Lemma 3.1** [45] The detour of a simple polygonal chain $P$ in the plane is attained by a pair of points $(p, q)$ on $P$, where $p$ is a vertex of $P$, and $p$ and $q$ are covisible.

Similar results were obtained for various other cases. Lemma 3.1 was generalized by Agarwal et al. [34] to the case of simple trees in the plane, and by Ebbers-Baumann et al. [46] it is shown that Lemma 3.1 also holds for simple polygons in the plane.

Figure 3.2 There is always a covisible pair of points attaining the detour.
Lemma 3.2 [47] For any metric space \( M \) on \( \mathbb{R}^2 \), the \( M \)-detour of a simple polygon \( P \) in the plane with \( \delta_M(P) > 1 \) is attained by a pair of points \((p, q)\) on the boundary of \( P \), with the property that \( \overline{pq} \cap P = \{p, q\} \), and at least one of \( p, q \) is a vertex of \( P \).

For the Euclidean metric, one can even show that every pair of points \((p, q)\) on the boundary that attains the detour of any nonconvex simple polygon \( P \) must have the property that \( \overline{pq} \cap P = \{p, q\} \).

It is easy to see that the detour of a closed simple polygonal curve is not necessarily attained at a vertex. Still, a somewhat weaker property was shown by Agarwal et al. [34] for this case:

Lemma 3.3 [34] The detour of a closed simple polygonal curve \( P \) of length \( \ell \) in the plane is attained by a pair \((p, q)\) of points of \( P \), such that, either one of them is a vertex of \( P \), or \( d_p(p, q) = \ell/2 \).

Moreover, Ebbers-Baumann et al. [46] observed that it is still the case, that a covisible pair of points attains the detour of \( P \). This is in fact true for arbitrary connected simple straight-line graphs in the plane.

Lemma 3.4 [46] The detour of a connected simple straight-line graph \( P \) in the plane is attained by a pair of covisible points of \( P \).

Note that the dilation of a geometric graph is not necessarily attained at a covisible pair of vertices. Moreover, most of these properties fail to hold in higher dimensions. The detour of a polygonal curve in \( \mathbb{R}^3 \), for instance, is not necessarily attained at a vertex of the chain [34].

3.4.2 Algorithmic Questions

Based on the earlier work of Narasimhan and Smid [36], Grüne [48] has shown that there is an \( \Omega(n \log n) \) lower bound in the algebraic decision tree model for computing the dilation of a monotone and hence simple planar polygonal curve. On the contrary, for the problem of computing the detour of such curves, no nontrivial lower bound is known.

However, as was shown by Agarwal et al. [34], computing the detour of a 3-dimensional polygonal path is as hard as Hopcroft’s problem: Given a set \( L \) of \( n \) lines in \( \mathbb{R}^2 \) and a set \( Q \) of \( n \) points in \( \mathbb{R}^2 \), decide whether any line of \( L \) contains any point of \( Q \).

The idea is to reduce an instance of Hopcroft’s problem to the problem of computing the detour of a 3-dimensional path. To this end, a 3-dimensional path \( P_{L,Q} \) is built in such a way that \( P_{L,Q} \) has infinite detour (i.e., it self-intersects), iff any line of \( L \) contains any point of \( Q \). By using techniques developed by Erickson [49], the construction is then modified to cover the case in which it is known in advance that the input chain is not self-intersecting.

The construction shows that, if there is an algorithm to compute \( \delta(P) \) for a simple polygonal chain \( P \) on \( n \) vertices in \( \mathbb{R}^3 \) in \( T(n) \) time, Hopcroft’s problem can be solved in \( O(n \log n + T(n)) \) time. There is an abundance of evidence that suggests that Hopcroft’s problem has an \( \Omega(n^{4/3}) \) lower bound [49] in any reasonable model of computation.

On the positive side, for arbitrary connected plane graphs \( P \) on \( n \) vertices, the detour can be computed in \( O(n^2) \) time in the following way: Compute the shortest path distance for all pairs of vertices of \( P \). As \( P \) is planar this can be done in \( O(n^2) \) time [50]. For every pair of edges \( e, f \) of \( P \), compute the detour \( \delta^P(e, f) = \max(\delta^P(x, y) \mid x \in e, y \in f) \). This can be done in constant time per pair, as there are only four combinatorial different types of shortest paths going from points on \( e \) to points on \( f \).

The structural properties shown in the previous section can be exploited to obtain faster algorithms in some cases. The case in which \( P \) is a planar polygonal chain without self-intersections was first studied by Ebbers-Baumann et al. [45], where the following result is shown.

Theorem 3.7 [45] Let \( P \) be a simple polygonal chain on \( n \) vertices in the plane, and let \( \epsilon \) be a positive constant. In \( O(n \log n) \) time a pair of points \((p, q)\) on \( P \) can be computed, such that \( \delta(P) \leq (1 + \epsilon)\delta^P(p, q) \).
The problem of exactly computing $\delta(P)$ for a simple polygonal chain $P$ was independently considered by Langermann et al. and Agarwal et al. [32,33] (see also Reference 34 for a combined version).

The approach in Reference 34 is as follows: First, a deterministic $O(n \log n)$ time algorithm for the decision problem is developed, that is, an algorithm that decides on input $(P, \kappa)$ whether $\delta(P) \leq \kappa$. Note that according to Lemma 3.1, $\delta(P) \leq \kappa$ iff $\delta^P(q, p) \leq \kappa$ for all vertices $p \in P$ and all points $q \in P$.

This problem can be restated in a form that makes it amenable to range-searching techniques: Let $p_0$ be the first point of $P$. For a point $p \in P$, define the weight of $p$ to be $\omega(p) = d_p(p_0, p)/\kappa$. Let $C$ denote the cone $z = \sqrt{x^2 + y^2}$ in $\mathbb{R}^3$ and map each vertex $p = (x, y, \omega(p)) \in P$ to the cone $C_p = C + (x, y, \omega(p))$, that is, translate the apex of $C$ (that is, the origin) to the point $(x, y, \omega(p))$. If $C_p$ is regarded as the graph of a bivariate function, which will also be denoted by $C_p$, then for any point $x \in \mathbb{R}^2$, $C_p(x) = |xp| + \omega(p)$.

Map a point $q = (q_x, q_y) \in P$ to the point $\hat{q} = (q_x, q_y, \omega(q))$ in $\mathbb{R}^3$, cf Figure 3.3 on page 62. Now, for any point $q \in P$ and a vertex $p \in P$ that lies before $q$ on $P$ (in the sense that $d_p(p, q) = d_p(p_0, q) - d_p(p_0, p)$), $\delta^P(q, p) \leq \kappa$ if and only if $\hat{q}$ lies below the cone $C_p$:

$$\delta^P(q, p) \leq \kappa \iff \frac{\omega(q) - \omega(p)}{q \hat{p}} \leq \kappa \iff \frac{d_p(p_0, q) - d_p(p_0, p)}{q \hat{p}} \leq \kappa \iff \frac{d_p(p_0, q)}{q \hat{p}} \leq \kappa \iff \omega(q) \leq C_p(q).$$

That is, $\delta^P(q, p) \leq \kappa$ iff $\hat{q}$ lies below the cone $C_p$, and consequently $\delta(P) \leq \kappa$ iff the polygonal chain $\hat{P} = \{\hat{p} | p \in P\}$ lies below the lower envelope of the cones $\{C_p | p$ is a vertex of $P\}$. By exploiting the covisibility property from Lemma 3.1, this condition can be verified in $O(n \log n)$ deterministic time.

By using a randomized technique of Chan [51] or parametric search [52], the decision procedure is then turned into an algorithm for actually computing $\delta(P)$ (parametric search incurs a polylogarithmic overhead).

**Theorem 3.8 [34]** Let $P$ be a simple polygonal chain on $n$ vertices in the plane. There is

- A deterministic algorithm to decide in $O(n \log n)$ time whether $\delta(P) \leq \kappa$, for any $\kappa > 0$
- A randomized algorithm to compute $\delta(P)$ in $O(n \log n)$ expected time
- A deterministic algorithm to compute $\delta(P)$ in $O(n \log^{O(1)} n)$ time.

By using appropriate recursive partitioning schemes—based on Lemma 3.3 or the variant of Lemma 3.1 for trees—similar techniques can be applied to the case in which $P$ is a tree or a cycle.

**Theorem 3.9 [34]** Let $P$ be a simple closed polygonal chain or a plane tree on $n$ vertices in the plane. There is a randomized algorithm to compute $\delta(P)$ in $O(n \log^2 n)$ expected time.

As shown by Grüne et al. [47,48], the approach of Reference 45 can be modified to handle the case in which $P$ is a simple polygon in the plane. This yields an efficient approximation algorithm.

FIGURE 3.3 The chain $P$ lifted to $\mathbb{R}^3$. 
Theorem 3.10 [47,48] Let \( P \) be a simple polygon on \( n \) vertices in the plane, and let \( \epsilon \) be a positive constant. In \( O(n \log n) \) time, a pair of points \((p, q)\) on the boundary of \( P \) can be computed, such that \( \delta(P) \leq (1 + \epsilon)\delta^*(p, q) \).

The fastest known algorithm for computing \( \delta(P) \) for a simple polygon \( P \) exactly is similar to the brute-force approach described at the beginning of this section. Of course Lemma 3.2 plays a crucial role here. Moreover, the shortest path computation is more involved as shortest geodesic paths inside \( P \) have to be computed.

Theorem 3.11 [47] Let \( P \) be a simple polygon on \( n \) vertices in the plane. There is a deterministic algorithm to compute \( \delta(P) \) in \( O(n^2) \) time.

As already mentioned, it is no longer true that the detour is attained at a vertex of \( P \), when \( P \) is a simple polygonal chain in \( \mathbb{R}^3 \). This makes the 3-dimensional algorithm considerably more complicated, and less efficient, than its 2-dimensional counterpart.

Theorem 3.12 [34] Let \( P \) be a simple polygonal chain on \( n \) vertices in \( \mathbb{R}^3 \). There is a randomized algorithm to compute \( \delta(P) \) in \( O(n^{16/9} + \epsilon \log n) \) expected time for any \( \epsilon > 0 \).

3.4.3 Low Detour Embeddings of Point Sets

Besides computing the detour of given graphs, the problem of constructing plane graphs of small detour that contain a given finite point set was also investigated.

Definition 3.3 (Detour of a Point Set) The detour \( \hat{\delta}(P) \) of a finite point set \( P \) in the plane is the smallest possible detour of any finite plane graph that contains all points of \( P \), that is,

\[ \hat{\delta}(P) := \inf_{P \subseteq G} \delta(G) \]

Even for a point set \( P \) of size 3, computing \( \hat{\delta}(P) \) is a nontrivial task. For the dilation, the optimum solution must be a triangulation, as an optimal solution only contains straight edges, and adding edges never increases the dilation. Still, it is not known how to efficiently compute the triangulation that minimizes the dilation of a given point set.

As a consequence of Lemmas 3.3 and 3.4, the detour of any rational point set \( P \) is bounded by two, as it can be embedded into a square grid, that is, \( \hat{\delta}(P) \leq 2 \) for all \( \mathbb{Q} \). A construction of Reference 46 shows that this can be improved:

Theorem 3.13 [46] There is a periodic, plane covering graph \( G_{\infty} \) of detour \( 1.67784... \), such that each finite set of rational points is contained in a finite part of a scaled copy of \( G_{\infty} \).

On the other side of the spectrum, Reference 46 gives an example of a point set \( P \) with detour \( \hat{\delta}(P) \geq \pi/2 = 1.57079... \). A subsequent work in References 53, 54 improved this lower bound by exhibiting a point set \( P \) with \( \hat{\delta}(P) > \pi/2 \).

Theorem 3.14 [46,54]

- Let \( P \) be the vertex set of the regular \( n \)-gon on the unit circle. Then \( \hat{\delta}(P) \geq \pi/2 \).
- Let \( P = \{(x, y) \mid x, y \in \{-9, \ldots, 9\}\} \). Then \( \hat{\delta}(P) \geq (1 + 10^{-11})\pi/2 \).
3.5 Low-Dilation Networks

Besides computing the dilation of given graphs, the problem of constructing certain finite plane graphs \( G = (V, E) \) of small dilation that contain a given finite point set \( P \) is also interesting. There are several different variants of this problem, depending on whether \( G \) may or may not contain Steiner-points, or if \( G \) is restricted to belong to a certain class of graphs \( G \), such as triangulations and trees.

**Definition 3.4 (Dilation of a Point Set)** Let \( G \) be a class of graphs and \( P \) be a finite point set in the plane. The dilation \( \Delta^G(P) \) of \( P \) w.r.t. \( G \) is the smallest possible dilation of any finite plane graph \( G = (P, E) \) in \( G \), that is,

\[
\Delta^G(P) := \min_{G=(P,E)\in G} \Delta(G).
\]

If \( G \) is the class of all graphs, we can assume \( G \) to be a triangulation, as an optimal solution only contains straight edges, and adding edges never increases the dilation. We omit the superscript \( G \) in this case.

### 3.5.1 Triangulations

A triangulation defining \( \Delta(P) \) is called a *minimum dilation triangulation of \( P \).* So far, only little research has been conducted on minimum dilation triangulations. The complexity status of the problem is open. Most work upperbounds the dilation of certain types of triangulations. Chew [55] has shown that the rectilinear Delaunay triangulation has dilation at most \( \sqrt{10} \). Dobkin et al. [56] gave a similar result for the Euclidean Delaunay triangulation. They show that its dilation can be bounded from above by \( ((1 + \sqrt{5})/2) \pi \approx 5.08 \). This bound was further improved to \( 2\pi/(3 \cos(\pi/6)) \approx 2.42 \) by Keil and Gutwin [57,58]. Das and Joseph [59] generalized all these results by identifying two properties of planar graphs such that if \( A \) is an algorithm that computes a planar graph from a given set of points and if all the graphs constructed by \( A \) meet these properties, then the dilation of all the graphs constructed by \( A \) is bounded by a constant.

**Exclusion and inclusion regions:** When investigating optimal triangulations, it is usually instructive to consider local properties of the edges in these triangulations. One important class of local properties that has been studied extensively, for example, for minimum weight triangulations are *exclusion regions*. They provide a necessary condition for the inclusion of an edge into an optimal triangulation: If \( p \) and \( q \) are two points in \( P \), then the edge \( e := pq \) can only be contained in an optimal triangulation of \( P \) if no other points of \( P \) lie in (certain parts of) the exclusion region of \( e \).

To obtain an exclusion region for the minimum dilation triangulation, one can observe the following: We know from Reference 57 that the dilation of the Delaunay triangulation of \( P \) is bounded by \( \gamma = 2\pi/(3 \cos(\pi/6)) \). Moreover, if we have an edge \( e \) and two points \( x, y \) on opposite sides of \( e \) that are close to the center of \( e \), then the dilation between \( x \) and \( y \) is large, because \( e \) constitutes an obstacle that the shortest path between \( x \) and \( y \) has to surpass. In fact, if we can quantify this and show that the dilation between any pair of points in a certain region \( D_{e,Y} \) that lie on opposing sides of \( e \) is larger than \( \gamma \), we can conclude that, if \( D_{e,Y} \) contains such a pair of points, then \( e \) cannot be contained in the minimum dilation triangulation of \( P \), as the Delaunay triangulation gives a better dilation than any triangulation containing \( e \).

The upper bound of Reference 57 can also be used to obtain a *sufficient* condition for the inclusion of an edge. More specifically, consider for two points \( p, q \in P \) the ellipsoid \( E_{p,q,y} \) with foci \( p \) and \( q \) that is given by \( E_{p,q,y} = \{ x \in \mathbb{R}^2 | |px| + |qx| \leq \gamma \cdot |pq| \}. \) If \( E_{p,q,y} \) is empty, then the line segment \( pq \) has to be included in the minimum dilation triangulation of \( P \), as otherwise the dilation between \( p \) and \( q \) would be larger than \( \gamma \).
Theorem 3.15 [60,61] Let \( \gamma = 2\pi / (3 \cos(\pi/6)) \) and \( p, q \in P \).

1. For any \( 0 < \alpha < 1/(2\gamma) \), the disk \( D_{\|pq\|,\alpha} \) of radius \( \alpha \|pq\| \) centered at the midpoint of \( \overline{pq} \) is an exclusion region for the minimum dilation triangulation.
2. The ellipsoid \( E_{p,q,\gamma} = \{ x \in \mathbb{R}^2 \mid \|px\| + \|qx\| \leq \gamma \|pq\| \} \) is an inclusion region for the minimum dilation triangulation.

Regular \( n \)-gons: Even for the vertex set \( S_n = \{s_1, \ldots, s_n\} \) of a regular \( n \)-gon, it is not known how to efficiently compute a minimum dilation triangulation. There is however some additional understanding of the structure of optimal triangulations in that case. In particular, there is a simple lower bound for the dilation of \( S_n \).

Theorem 3.16 [61] Let \( n \geq 74 \) and assume that \( \max_{1 \leq i < j \leq n} \|s_i - s_j\| = 2 \). For any triangulation \( T \) of \( S_n \) and any maximum dilation pair \( s_x, s_y \in S_n \) of \( T \), we have that

1. \( \Delta(T) \geq \sqrt{2 - \sqrt{3}} + \sqrt{3}/2 \approx 1.3836 \),
2. \( |a - b| > 5n/12 \), and
3. \( \|s_a - s_b\| > \left( \sqrt{6 + 3\sqrt{3} + \sqrt{2 - \sqrt{3}}} \right)/2 \approx 1.93185 \).

This can be used to derive an efficient approximation algorithm that computes a triangulation the dilation of which is within a factor of 1 + \( O(1/\sqrt{\log n}) \) of the optimum.

Theorem 3.17 [61] In \( O(n\sqrt{\log n}) \), a triangulation \( T^* \) of \( S_n \) can be computed, such that

\[
\Delta(T^*) \leq \left( 1 + O\left(1/\sqrt{\log n}\right) \right) \Delta(S_n).
\]

3.5.2 Stars

The problem of computing a minimum dilation star of \( P \), that is, a graph \( G \in \mathcal{G} \) defining \( \Delta^P \) where \( \mathcal{G} \) is a star, was considered for the first time by Eppstein and Wortman [35]. They proved the following:

Theorem 3.18 Let \( P \) be a set of \( n \) points in \( \mathbb{R}^2 \), one can compute a minimum dilation star of \( P \) in \( O(n2^{2(n)} \log n) \) expected time, where \( \alpha \) is the functional inverse of Ackermann’s function [62].

The algorithm works by iteratively selecting a random vertex \( c \) in a region \( R \), evaluating the dilation that would result from using \( c \) as a center, computing the region \( R \) that could contain a center yielding a lower dilation, and discarding the vertices outside \( R \). By evaluating the dilation \( \Delta_c \) of a given star with center at \( c \) in \( O(n \log n) \) time was discussed in Section 3.3.

The region \( R \) is the intersection of \( O(n) \) ellipses defined by the \( O(n) \) pairs of points identified in Section 3.3, that is, for each of the pairs \( v_i \) and \( v_j \) the level set \( f^l_{i,j} = \{ x \in \mathbb{R}^2 \mid f_{i,j}(x) = \frac{|v_i x| + |v_j y|}{|v_i y|} \leq l \} \) defines an ellipsoid with foci \( v_i \) and \( v_j \). The intersection of those \( O(n) \) ellipses can be described by \( O(n2^{2(n)}) \) arcs.

In each iteration, any vertex in \( R \) will be removed with probability \( 1/2 \); so the expected number of iterations is \( O(\log n) \), resulting in an \( O(n2^{2(n)} \log n) \) expected time algorithm.

3.5.3 Small Spanners with Small Dilation

Aronov et al. [63] considered the problem of constructing a minimum dilation graph given the number of edges as a parameter. Any spanner of a set of \( n \) points \( S \) must have at least \( n - 1 \) edges, otherwise the graph would not be connected and the dilation would be infinite. The quantity \( \Delta(S,k) \) is defined as:

\[
\Delta(S,k) = \min_{\|V(G)\|=S, |E(G)|=n-1+k} \Delta(G).
\]
Thus $\Delta(S,k)$ is the minimum dilation one can achieve with a network on $S$ that has $n - 1 + k$ edges.

**Theorem 3.19** For any $n$ and any $k$ with $1 \leq k \leq 2n$, there is a set $S$ of $n$ points such that any graph on $S$ with $n - 1 + k$ edges has dilation at least $\frac{2}{\pi} \cdot \lfloor \frac{n}{k+1} \rfloor - 1$.

Consider a set $S$ of $n$ points $p_1, \ldots, p_n$ spaced equally on the unit circle, and let $o$ be the center of the circle. The first step is to prove a lower bound on any tree $T$ for $S$.

Let $x$ and $y$ be two points in $S$ and let $\gamma$ and $\gamma'$ be two paths from $x$ to $y$ avoiding $o$. The paths $\gamma$ and $\gamma'$ are (homotopy) equivalent if $\gamma$ and $\gamma'$ belong to the same homotopy class in the punctured plane $\mathbb{R}^2 \setminus \{o\}$. Let $\gamma_i$ be the unique path in $T$ from $p_i$ to $p_{i+1}$ (where $p_{n+1} := p_1$). Aronov et al. [63] prove that there must be at least one index $i$ for which $\gamma_i$ is not equivalent to the straight segment $p_i p_{i+1}$. As the path $\gamma_i$ must not only “go around” $o$ but must do so using points $p_j$ on the circle only it follows that $\Delta(S,0) \geq \frac{2}{\pi} n - 1$.

Theorem 3.19 follows from the above-mentioned argument by letting $S$ consist of $k + 1$ copies of the earlier construction, that is, sets $S_i$, for $1 \leq i \leq k + 1$, each consisting of at least $\lfloor n/(k+1) \rfloor$ points. The points in $S_i$ are placed equally spaced on a unit-radius circle with center at $(2i - 1, 0)$. The set $S$ is the union of $S_1, \ldots, S_{k+1}$.

In the same paper, Aronov et al. [63] give a matching upper bound. The algorithm constructs a graph $G = (S,E)$ with $n - 1 + k$ edges, and dilation $O(n/(k + 1))$ in $O(n \log n)$ time.

Let $m \leftarrow \lfloor (k + 5)/2 \rfloor$. Partition a minimum spanning tree $T$ of $S$ into $m$ disjoint connected subtrees, $T_1, \ldots, T_m$, each containing $O(n/m)$ points. The edges of each subtree is added to $E$. Next, consider a Delaunay triangulation of $S$. For each pair of subtrees $T_i$ and $T_j$, the shortest Delaunay edge (if any) is added to $E$. This completes the construction of $G = (S,E)$.

The number of edges in $G$ is at most $n - 1 + k$ and $\Delta(G)$ can be bounded by $O(n/(k + 1))$ [63].

**Acknowledgments**

We would like to thank Michiel Smid for valuable comments on an earlier draft of this chapter.

**References**

52. Megiddo, N., Applying parallel computation algorithms in the design of serial algorithms, JACM, 30(4), 852, 1983.
55. Chew, L.P., There are planar graphs almost as good as the complete graph, JCSS, 39, 205, 1989.


