May 2016

Algorithmic and Combinatorial Results on Fence Patrolling, Polygon Cutting and Geometric Spanners

Anirban Ghosh
University of Wisconsin-Milwaukee

ALGORITHMIC AND COMBINATORIAL RESULTS ON FENCE PATROLLING, POLYGON CUTTING AND GEOMETRIC SPANNERS

by

Anirban Ghosh

A Dissertation Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

in Engineering

at

The University of Wisconsin-Milwaukee

May 2016

ABSTRACT

ALGORITHMIC AND COMBINATORIAL RESULTS ON FENCE PATROLLING, POLYGON CUTTING AND GEOMETRIC SPANNERS

by

Anirban Ghosh

The University of Wisconsin-Milwaukee, 2016
Under the Supervision of Professor Adrian Dumitrescu

The purpose of this dissertation is to study problems that lie at the intersection of geometry and computer science. We have studied and obtained several results from three different areas, namely–geometric spanners, polygon cutting, and fence patrolling. Specifically, we have designed and analyzed algorithms along with various combinatorial results in these three areas. For geometric spanners, we have obtained combinatorial results regarding lower bounds on worst case dilation of plane spanners. We also have studied low degree plane lattice spanners, both square and hexagonal, of low dilation. Next, for polygon cutting, we have designed and analyzed algorithms for cutting out polygon collections drawn on a piece of planar material using the three geometric models of saw, namely, line, ray and segment cuts. For fence patrolling, we have designed several strategies for robots patrolling both open and closed fences.

To my parents and my loving wife

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I express my gratitude and deep respect for my doctoral advisor Prof. Adrian Dumitrescu. His constant inspiration, patience and generous guidance impelled me towards accomplishing my doctoral work. He has been a wonderful mentor and a source of great inspiration and will always be held in high regard for the rest of my life.

I am thankful to my parents and my loving wife Sunetra for always inspiring me in every moment of my life, especially during my doctoral journey. I am grateful to Prof. Subhas C. Nandy of the Indian Statistical Institute, Kolkata for bringing into my knowledge about undertaking this doctoral opportunity. I hold deep regards for my brother-in-law Supratim Ghosh and my sister-in-law Atrayee Basu for inspiring me to pursue my PhD degree. I would like to convey my sincere regards to my colleagues, friends and relatives for supporting me during these doctoral years.

I am grateful to all my doctoral committee members – Prof. Christine Cheng, Prof. Chiu-Tai Law, Prof. Jeb Willenbring and Prof. Guangwu Xu for rendering their valuable inputs and suggestions in my research work.

Finally, my deepest gratitude and appreciation goes to the Department of Computer Science at the University of Wisconsin-Milwaukee for supporting my research.

PUBLICATIONS

1. A. Dumitrescu, A. Ghosh, and C. D. Tóth, On fence patrolling by mobile agents, *The Electronic Journal of Combinatorics*, **21(3)**, 2014, P3.4. A preliminary version in: *Proceedings of the 25th Canadian Conference on Computational Geometry*, (CCCG 2013), Waterloo, Ontario, Canada, August 2013, pp. 271–276.

2. A. Dumitrescu, A. Ghosh, and M. Hasan, On Collections of Polygons Cuttable with a Segment Saw, *Proceedings of the International Conference on Algorithms and Discrete Applied Mathematics*, (CALDAM 2015), IIT Kanpur, India, February 2015, vol. 8959 of LNCS, pp. 58–68.

3. A. Dumitrescu and A. Ghosh, Lower bounds on the dilation of plane spanners, *International Journal of Computational Geometry and Applications*, 2016, to appear. A preliminary version in: *Proceedings of the International Conference on Algorithms and Discrete Applied Mathematics*, (CALDAM 2016), Kerala, Thiruvanthapuram, India, February 2016; vol. 9602 of LNCS, pp. 139–151. Also available at `http://arxiv.org/abs/1509.07181`.

4. A. Dumitrescu and A. Ghosh, Lattice spanners of low degree, *Proceedings of the International Conference on Algorithms and Discrete Applied Mathematics*, (CALDAM 2016), Kerala, Thiruvanthapuram, India, February 2016; vol. 9602 of LNCS, pp. 152-163. Preprint with improved results available at `http://arxiv.org/abs/1602.04381`.

Chapter 0

Introduction

0.1 Geometry and computer science

Geometry as a branch of mathematics is being studied since ancient times. Several civilizations around the globe had exhibited keen interest in the development of geometry as a discipline bringing forth a plethora of results and observations. The beauty of geometry lies in its expressibility, mostly, using diagrams and visualization, which had always attracted the interest of philosophers and scientists. Geometry is omnipresent. From the shape of a snowflake to the galaxies – everywhere we can witness the grandeur of geometry and its far and wide applications. On the other hand, computer science in the modern civilization has emerged as an important field of research whose presence can be realized in almost every step of our lives.

Henceforth, it becomes quite interesting to study the problems that are common to both the fields – geometry and computer science. In this dissertation, we have studied various geometric

problems that belong to the intersection of the aforesaid domains.

0.2 Research areas

The following, provides a glimpse of the problems which are studied in this dissertation.

0.2.1 Geometric spanners

Let P be a set of points. Geometric spanners are graphs that approximate well the pairwise distances in P. The area of design and analysis of geometric spanners has been widely researched since the last three decades. Popular goals include constructions of low stretch factor geometric spanners that have few edges, bounded degree and so on. Geometric spanners find their applications in various areas of research such as robotics, computer networks, distributed systems, road constructions and plenty of others.

In this area, we have focused on lower bound constructions of plane spanners. In particular, we found a new lower bound construction as a partial answer to a decade long standing open question about the best lower bound on the spanning ratio of plane geometric graphs. We have also presented lower bounds in the domain of degree constrained plane geometric graphs. In this regard, experimental algorithmics in the domain of parallel programming helped us to improve the bounds. We discuss the bounds in Chapter 1. In addition, we have investigated the dilations of low degree lattice spanners (square and hexagonal), which were previously not studied in the area of geometric spanners. The topic will be studied in Chapter 2.

Relevant papers.

1. A. Dumitrescu and A. Ghosh, Lower bounds on the dilation of plane spanners, *International Journal of Computational Geometry and Applications*, 2016, to appear. A preliminary version in: *Proceedings of the International Conference on Algorithms and Discrete Applied Mathematics*, (CALDAM 2016), Kerala, Thiruvanthapuram, India, February 2016;

vol. 9602 of LNCS, pp. 139–151. Also available at http://arxiv.org/abs/1509.07181.

2. A. Dumitrescu and A. Ghosh, Lattice spanners of low degree, *Proceedings of the International Conference on Algorithms and Discrete Applied Mathematics*, (CALDAM 2016), Kerala, Thiruvanthapuram, India, February 2016; vol. 9602 of LNCS, pp. 152–163. Preprint with improved results available at http://arxiv.org/abs/1602.04381.

0.2.2 Polygon cutting

The problem of polygon cutting was first introduced by Overmars and Welzl, in their seminal paper in 1985. Since then the problem has been deeply studied by geometers under several settings. The problem is to efficiently cut out a polygon collection drawn on a planar piece of material, such as, paper. Here, by efficiency we refer to the number of cuts and also the total length of the cuts. Cutting tools are geometrically classified into three types – line cut, ray cut, and segment cut. In this dissertation, we present algorithms for cutting out polygon collections drawn on a piece of planar material using the three geometric models of saw. Furthermore, we also investigate and study various uncuttable polygon collections. For the results, we refer the reader to Chapter 3.

Polygon cutting algorithms are particularly useful in the industrial applications where cutting out polygonal objects from metal sheets etc. are required. Also, from geometrical perspective, the algorithms hold a source of great interest to the theoretical computer scientists.

Relevant paper. A. Dumitrescu, A. Ghosh, and M. Hasan, On Collections of Polygons Cuttable with a Segment Saw, *Proceedings of the International Conference on Algorithms and Discrete Applied Mathematics*, (CALDAM 2015), IIT Kanpur, India, February 2015, vol. 8959 of LNCS, pp. 58–68.

0.2.3 Multi-agent systems and fence patrolling

By multi-agent systems we refer to a collection of agents (possibly interacting) for accomplishing an objective, usually under some constrained setting. Agents can be robots, computer programs or even human beings. Fence patrolling is intensely studied in computer science. It is applied to numerous situations where surveillance is necessary. For instance, network administrators may use patrolling for detecting network failures or discovering web-pages to be indexed by a search engine. Patrolling algorithms are also applied in disaster environments where rescuing is necessary.

A fence can be open or closed. A closed fence is represented geometrically using a closed curve, such as a circle, and an open fence using a line segment. In this dissertation, we consider the multi-agent systems mainly from theoretical perspective where we design and analyze efficient strategies for agents patrolling a fence (open or closed) of finite length. Also, we have investigated (joint work with Cs. D. Tóth) an important conjecture regarding the optimality of an existing algorithm for unidirectional patrolling of a closed fence. We discuss these results in Chapter 4.

Relevant paper. A. Dumitrescu, A. Ghosh, and C. D. Tóth, On fence patrolling by mobile agents, *The Electronic Journal of Combinatorics*, **21(3)**, 2014, P3.4. A preliminary version in: *Proceedings of the 25th Canadian Conference on Computational Geometry*, (CCCG 2013), Waterloo, Ontario, Canada, August 2013, pp. 271–276.

Chapter 1

Lower bounds on the dilation of plane spanners

1.1 Introduction

Given a set of points P in the Euclidean plane, a *geometric graph* on P is a weighted graph $G = (V, E)$ where $V = P$ and an edge $uv \in E$ is the line segment with endpoints $u, v \in V$ weighted by the Euclidean distance $|uv|$ between them. For $t \geq 1$, a geometric graph G is a *t-spanner*, if for every pair of vertices u, v in V, the length of the shortest path $\pi_G(u, v)$ between them in G is at most t times $|uv|$, i.e., $\forall u, v \in V$, $|\pi_G(u, v)| \leq t|uv|$. A complete geometric graph on a set of points is a 1-spanner. Where there is no necessity to specify t, we use the term

geometric spanner. A geometric spanner G is *plane* if no two edges in G cross. In this chapter we only consider plane geometric spanners. A geometric spanner of degree at most k is referred to as a *degree k geometric spanner*.

Given a geometric spanner $G = (V, E)$, the *vertex dilation* or *stretch factor* of $u, v \in V$, denoted $\delta_G(u, v)$, is defined as

$$\delta_G(u, v) = |\pi_G(u, v)|/|uv|.$$

When G is clear from the context, we simply write $\delta(u, v)$. The *vertex dilation* or *stretch factor* of G, denoted $\delta(G)$, is defined as

$$\delta(G) = \sup_{u,v \in V} \delta_G(u, v).$$

The terms *graph theoretic dilation* and *spanning ratio* are also used in the literature. Refer to [22, 29, 35] for such definitions.

Given a point set P, let the *dilation* of P, denoted by $\delta_0(P)$, be the minimum stretch factor of a plane geometric graph (equivalently, triangulation) on vertex set P; see [34]. Similarly, let the *degree k dilation* of P, denoted by $\delta_0(P, k)$, be the minimum stretch factor of a plane geometric graph of degree at most k on vertex set P. Clearly, $\delta_0(P, k) \geq \delta_0(P)$ holds for any k. Furthermore, $\delta_0(P, j) \geq \delta_0(P, k)$ holds for any $j < k$. (Note that the term *dilation* has been also used with different meanings in the literature, see for instance [11, 30].)

In the last few decades, great progress has been made in the field of geometric spanners; for an overview refer to [25, 35]. Common goals include constructions of low stretch factor geometric spanners that have few edges, bounded degree and so on. A survey of open problems in this area along with existing results can be found in [11]. Geometric spanners find their applications in the areas of robotics, computer networks, distributed systems and many others. Refer to [1, 2, 4, 13, 23, 32] for various algorithmic results.

The existence of plane t-spanners for some constant $t > 1$ (with no restriction on degree)

was first investigated by Chew [15] in the 80s. He showed that it is always possible to construct a plane 2-spanner with $O(n)$ edges on a set of n points; he also observed that every plane geometric graph embedded on the 4 points placed at the vertices of a square has stretch factor at least $\sqrt{2}$. This was the best lower bound on the worst-case dilation of plane spanners for almost 20 years until it was shown by Mulzer [34] using a computer program that every triangulation of a regular 21-gon has stretch factor at least $(2\sin\frac{\pi}{21}+\sin\frac{5\pi}{21}+\sin\frac{3\pi}{21})/\sin\frac{10\pi}{21} = 1.4161\ldots$ Henceforth, it was posed as an open problem by Bose and Smid [11, Open Problem 1] (as well as by Kanj in his survey [26, Open Problem 5]): "*What is the best lower bound on the spanning ratio of plane geometric graphs? Specifically, is there a* $t > \sqrt{2.005367532} \approx 1.41611\ldots$ *and a point set* P*, such that every triangulation of* P *has spanning ratio at least* t*?*". We give a positive answer to the second question by showing that a set S of 23 points placed at the vertices of a regular 23-gon, has dilation $\delta_0(S) \geq (2\sin\frac{2\pi}{23}+\sin\frac{8\pi}{23})/\sin\frac{11\pi}{23} = 1.4308\ldots$

The problem can be traced back to a survey written by Eppstein [24, Open Problem 9]: "*What is the worst case dilation of the minimum dilation triangulation?*". The point set S also provides a partial answer for this question. From the other direction, the current best upper bound of 1.998 was proved by Xia [37] using Delaunay triangulations. Note that this bound is only slightly better than the bound of 2 obtained by Chew [15] in the 1980s. For previous results on the upper bound refer to [16, 18, 19, 29].

The design of low degree plane spanners is of great interest to geometers. Bose et al. [9] were the first to show that there always exists a plane t-spanner of degree at most 27 on any set of points in the Euclidean plane where $t \approx 10.02$. The result was subsequently improved in [5, 6, 7, 12, 27, 33] in terms of degree. Recently, Kanj et al. [28] showed that $t = 20$ can be achieved with degree 4. However, the question whether the degree can be reduced to 3 remains open at the time of this writing. If one does not insist on having a plane spanner, Das et al. [17] showed that degree 3 is achievable. While numerous papers have focused on upper bounds on the dilation of bounded degree plane spanners, not much is known about lower bounds. In this chapter, we explore this direction and provide new lower bounds for unrestricted degrees and

when degrees 3 and 4 are imposed.

A *greedy triangulation* of a finite point set P is constructed in the following way: starting with an empty set of edges E, repeatedly add edges to E in non-decreasing order of length as long as edges in E are noncrossing. Bose et al. [10] have showed that the greedy triangulation is a t-spanner, where $t = 8(\pi - \alpha)^2/(\alpha^2 \sin^2(\alpha/4)) \approx 11739.1$ and $\alpha = \pi/6$. Here we obtain a worst-case lower bound of 2.0268; in light of computational experiments we carried out, we believe that the aforementioned upper bound is very far from the truth.

Related work. If S_n is the set of n vertices of a regular n-gon, Mulzer [34] showed that

$$1.3836\ldots = \sqrt{2-\sqrt{3}} + \sqrt{3}/2 \leq \delta_0(S_n) \leq 0.471\pi/\sin 0.471\pi = 1.4858\ldots,$$

for every $n \geq 74$; the upper bound holds for every $n \geq 3$. Amarnadh and Mitra [3] have shown that in the case of a cyclic polygon (a polygon whose vertices are co-circular), the stretch factor of any *fan* triangulation (i.e., with a vertex of degree $n-1$) is ≤ 1.4846.

As mentioned earlier, low degree plane spanners for general point sets have been studied in [5, 7, 9, 12, 27, 33]. The construction of low degree plane spanners for the infinite square and hexagonal lattices has been recently investigated in [21].

Bose et al. [8] presented a finite convex point set for which there is a Delaunay triangulation whose stretch factor is at least $1.581 > \pi/2$, thereby disproving a widely believed $\pi/2$ upper bound conjectured by Chew [15]. They also showed that this lower bound can be slightly raised to 1.5846 if the point set need not be convex. This lower bound for non-convex point sets has been further improved to 1.5932 by Xia and Zhang [38].

Klein et al. [30] proved the following interesting structural result. Let S be a finite set of points in the plane. Then either S is a subset of one of the well-known sets of points whose triangulation is unique and has dilation 1, or there exists a number $\Delta(S) > 1$ such that each finite plane graph containing S among its vertices has dilation at least $\Delta(S)$.

Cheong et al. [14] showed that for every $n \geq 5$, there are sets of n points in the plane that do

not have a minimum-dilation spanning tree without edge crossings and that 5 is minimal with this property. They also showed that given a set S of n points with integer coordinates in the plane and a rational dilation $t > 1$, it is NP-hard to decide whether a spanning tree of S with dilation at most t exists, regardless if edge crossings are allowed or not.

Knauer and Mulzer [31] showed that for each edge e of a *minimum dilation triangulation* of a point set, at least one of the two half-disks of diameter about $0.2|e|$ on each side of e and centered at the midpoint of e must be empty of points[1].

When the stretch factor (or dilation) is measured over all pairs of points on edges or vertices of a plane graph G (rather than only over pairs of vertices) one arrives at the concept of *geometric dilation* of G; see [20, 22].

Our results. (I) Let S be a set of 23 points placed at the vertices of a regular 23-gon. Then, $\delta_0(S) = (2\sin\frac{2\pi}{23} + \sin\frac{8\pi}{23})/\sin\frac{11\pi}{23} = 1.4308\ldots$ (Theorem 1.1, Section 1.2). This improves the previous bound of $(2\sin\frac{\pi}{21} + \sin\frac{5\pi}{21} + \sin\frac{3\pi}{21})/\sin\frac{10\pi}{21} = 1.4161\ldots$, due to Mulzer [34], on the worst case dilation of plane spanners.

(II) (a) For every $n \geq 13$, there exists a set S of n points such that $\delta_0(S, 3) \geq 1 + \sqrt{3} = 2.7321\ldots$ (Theorem 1.2, Section 1.3).

(b) For every $n \geq 6$, there exists a set S of n points such that $\delta_0(S, 4) \geq 1 + \sqrt{(5-\sqrt{5})/2} = 2.1755\ldots$ (Theorem 1.3, Section 1.3). The previous best lower bound of $(2\sin\frac{\pi}{21} + \sin\frac{5\pi}{21} + \sin\frac{3\pi}{21})/\sin\frac{10\pi}{21} = 1.4161\ldots$, due to Mulzer [34] holds for any degree. Here we sharpen it for degrees 3 and 4.

(III) For every $n \geq 6$, there exists a set S of n points such that the stretch factor of the greedy triangulation of S is at least 2.0268.

Notations and assumptions. Let P be a planar point set and $G = (V, E)$ be a plane geometric graph on vertex set P. For $p, q \in P$, pq denotes the connecting segment and $|pq|$ denotes its Euclidean length. The degree of a vertex (point) $p \in P$ is denoted by $\deg(p)$. For a specific point

[1]Their result inaccurately states that the entire disk of that diameter is an exclusion region.

set $P = \{p_1, \ldots, p_n\}$, we denote a path in G consisting of vertices in the order $p_i, p_j, p_k, \ldots$ using $\varrho(i, j, k, \ldots)$ and by $|\varrho(i, j, k, \ldots)|$ its total Euclidean length. The graphs we construct have the property that no edge contains a point of P in its interior.

1.2 A new lower bound on the dilation of plane spanners

In this section, we show that the set $S = \{s_0, \ldots, s_{22}\}$ of 23 points placed at the vertices of a regular 23-gon has dilation $\delta_0(S) \geq (2\sin\frac{2\pi}{23} + \sin\frac{8\pi}{23})/\sin\frac{11\pi}{23} = 1.4308\ldots$ (see Fig. 1.1).

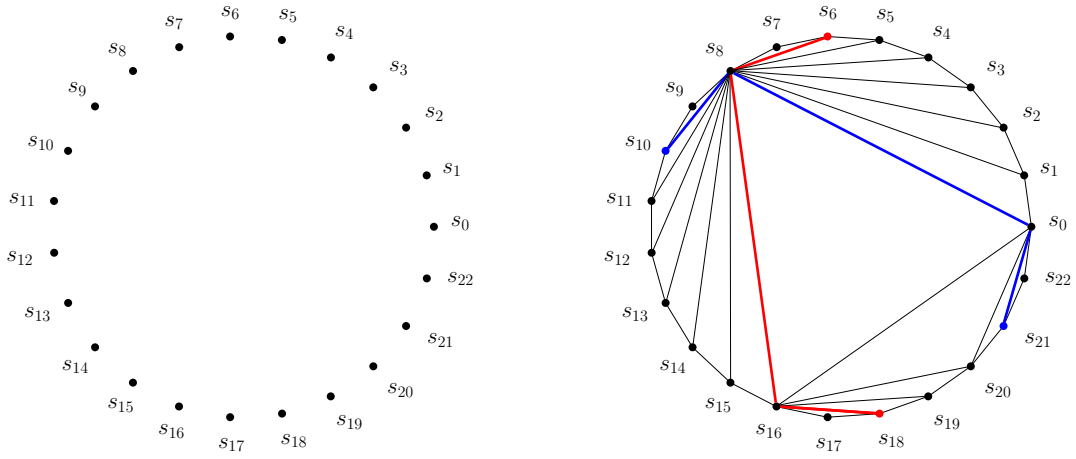

Figure 1.1: Left: The set S of 23 points placed at the vertices of a regular 23-gon. Right: A triangulation of S with stretch factor $(2\sin\frac{2\pi}{23} + \sin\frac{8\pi}{23})/\sin\frac{11\pi}{23} = 1.4308\ldots$, which is achieved by the detours for the pairs s_{10}, s_{21} and s_6, s_{18}. The shortest paths connecting the pairs are shown in blue and red, respectively.

Assume that the points lie on a circle of unit radius. We first present a theoretical proof showing that $\delta_0(S) \geq (\sin\frac{2\pi}{23} + \sin\frac{4\pi}{23} + \sin\frac{5\pi}{23})/\sin\frac{11\pi}{23} = 1.4237\ldots$; we then raise the bound to $\delta_0(S) \geq (2\sin\frac{2\pi}{23} + \sin\frac{8\pi}{23})/\sin\frac{11\pi}{23} = 1.4308\ldots$ using a computer program. The result obtained by the program is tight as there exists a triangulation of S (see Fig. 1.1 (right)) with stretch factor exactly $(2\sin\frac{2\pi}{23} + \sin\frac{8\pi}{23})/\sin\frac{11\pi}{23} = 1.4308\ldots$

Define the *convex hull length* of a chord $s_is_j \in S$ as $\mu(i, j) = \min(|i - j|, 23 - |i - j|)$. Observe that $1 \leq \mu(i, j) \leq 11$. Since triangulations are maximal planar graphs, we only consider triangulations of S while computing $\delta_0(S)$; in particular, every edge of the convex hull of S is present. Note that there are $C_{21} = 24,466,267,020$ triangulations of S. Here $C_n = \frac{1}{n+1}\binom{2n}{n}$

is the n^{th} *Catalan number* and there are C_n ways to triangulate a convex polygon with $n+2$ vertices.

If $s_i, s_j \in S$, then $|s_i s_j| = 2\sin\frac{\mu(i,j)\pi}{23}$. Consider a shortest path connecting $s_i, s_j \in S$ consisting of k edges with convex hull lengths $n_1, \ldots, n_k$; its length is $|\varrho(i, \ldots, j)| = 2\sum_{h=1}^{k} \sin\frac{n_h\pi}{23}$. Let $\lambda = \mu(i,j)$ and

$$g(\lambda, n_1, \ldots, n_k) = \frac{|\varrho(i, \ldots, j)|}{|s_i s_j|} = \frac{\sum_{h=1}^{k} \sin\frac{n_h\pi}{23}}{\sin\frac{\lambda\pi}{23}}. \tag{1.1}$$

We will use $\lambda = 11$ in all subsequent proofs of this section and therefore we set

$$f(n_1, \ldots, n_k) := g(11, n_1, \ldots, n_k). \tag{1.2}$$

Various values of f, as given by (1.1) and (1.2), will be repeatedly used in lower-bounding the stretch factor of point pairs in specific configurations, i.e., when some edges are assumed to be present. Observe that f is a symmetric function that can be easily computed (tabulated) at each tuple $n_1, \ldots, n_k$; see Table 1.1.

	$f(4,7)$	$1.3396\ldots$	❖	$f(2,2,8)$	$1.4308\ldots$
	$f(5,6)$	$1.3651\ldots$	❖	$f(3,3,5)$	$1.4312\ldots$
❖	$f(5,7)$	$1.4514\ldots$	❖	$f(3,4,4)$	$1.4409\ldots$
❖	$f(6,6)$	$1.4650\ldots$	❖	$f(1,4,7)$	$1.4761\ldots$
	$f(2,3,6)$	$1.4023\ldots$	❖	$f(2,3,7)$	$1.4886\ldots$
	$f(1,5,5)$	$1.4061\ldots$	❖	$f(3,3,6)$	$1.5312\ldots$
❖	$f(2,4,5)$	$1.4237\ldots$	❖	$f(1,1,4,5)$	$1.4263\ldots$
❖	$f(1,3,8)$	$1.4257\ldots$	❖	$f(1,2,3,5)$	$1.4388\ldots$

Table 1.1: Relevant values of $f(n_1, \ldots, n_k)$ as required by the proofs in this section. Values used explicitly in the proofs are marked using ❖.

Given a chord $s_0 s_i$, let $\texttt{lower}(s_0 s_i) = \{s_{i+1}, \ldots, s_{22}\}$ and $\texttt{upper}(s_0 s_i) = \{s_1, \ldots s_{i-1}\}$. The range of possible convex hull lengths of the longest chord in a triangulation of S is given by the following.

Lemma 1.1. *If ℓ is the convex hull length of the longest chord in a triangulation of S, then $\ell \in$*

$\{8, 9, 10, 11\}$.

Proof. We clearly have $\ell \geq 2$. Since S is symmetric, we can assume that $s_0 s_\ell$ is the longest chord. Since $\mu(i, j) \leq 11$ for any $0 \leq i, j \leq 22$, we have $\ell \leq 11$. Suppose for contradiction that $2 \leq \ell \leq 7$. Then $s_0 s_\ell$ is an edge of some triangle $\Delta s_0 s_\ell s_m$, where $\ell + 1 \leq m \leq 22$. In particular,

$$\mu(0, \ell), \mu(\ell, m), \mu(0, m) \leq \ell \leq 7. \tag{1.3}$$

If $m \leq 11$, then $\mu(0, m) = \min(m, 23 - m) = m \geq \ell + 1$, a contradiction to ℓ's maximality. Assume now that $m \geq 12$; then $\mu(0, m) = 23 - m \leq \ell$, since ℓ is the length of a longest chord. It follows that $m \geq 23 - \ell \geq 23 - 7 = 16$. If $m - \ell \leq 11$, then $\mu(\ell, m) = m - \ell \geq 16 - 7 = 9$, a contradiction to (1.3). If $m - \ell \geq 12$, then $\mu(\ell, m) = 23 - (m - \ell) = 23 - m + \ell \geq \ell + 1$, a contradiction to ℓ's maximality. Consequently, we have $8 \leq \ell \leq 11$, as required. □

Proof outline. For every $\ell \in \{8, 9, 10, 11\}$, if the longest chord in a triangulation T has length ℓ, we show that $\delta(T) \geq f(2, 4, 5) = 1.4237\ldots$ Assuming that $s_0 s_\ell$ is a longest chord, we consider the triangle with base $s_0 s_\ell$ and third vertex in $\texttt{upper}(s_0 s_\ell)$ or $\texttt{lower}(s_0 s_\ell)$, depending on ℓ. For each such triangle, we show that if the edges of the triangle along with the convex hull edges of S are present, then in any resulting triangulation there is a pair whose stretch factor is at least $f(2, 4, 5) = 1.4237\ldots$ Essentially, the long chords act as obstacles which contribute to long detours for some point pairs. In four subsequent lemmas, we consider the convex hull lengths $8, 9, 10, 11$ (from Lemma 1.1) successively.

In some arguments, we consider a *primary pair* s_i, s_j, and possible shortest paths between the two vertices. We show that if certain intermediate vertices are present in $\pi(s_i, s_j)$, then $\delta(s_i, s_j) \geq f(2, 4, 5)$. Otherwise if certain long edges are present in $\pi(s_i, s_j)$, then $\delta(s_u, s_v) \geq f(2, 4, 5)$, where s_u, s_v is a *secondary pair*. In the figures, wherever required, we use circles and squares to mark the primary and secondary pairs, respectively (see for instance Fig. 1.2). In some of the cases, a primary pair suffices in the argument, i.e., no secondary pair is needed.

Lemma 1.2. *If* $\ell = 8$*, then* $\delta(T) \geq f(2,4,5) = 1.4237\ldots$

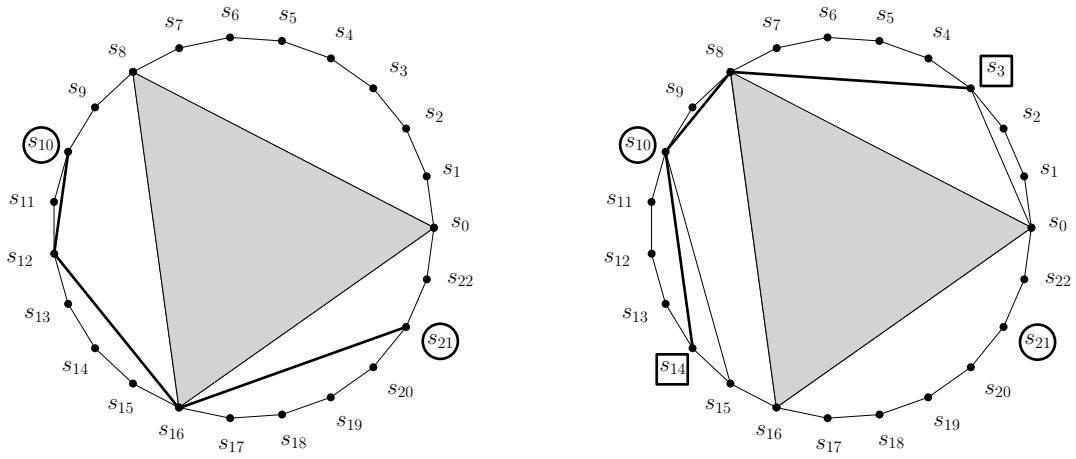

Figure 1.2: Illustrating Lemma 1.2. Left: $s_{12} \in \pi(s_{10}, s_{21})$, primary pair: s_{10}, s_{21}. Right: $s_{10}s_{15} \in \pi(s_{10}, s_{21})$, primary pair: s_{10}, s_{21}, secondary pair: s_3, s_{14}.

Proof. Refer to Fig. 1.2. Let s_0s_8 be the longest chord. The triangle with base s_0s_8 and third vertex in $\texttt{lower}(s_0s_8)$ has two other sides of convex hull lengths 7 and 8. It thus suffices to consider the triangle $\Delta s_0s_8s_{16}$ only and assume that the edges s_0s_8, s_8s_{16} and s_0s_{16} are present.

In this proof, the primary pair is s_{10}, s_{21} and the secondary pair is s_3, s_{14}. Now, consider the pair s_{10}, s_{21}. Note that either $s_0 \in \pi(s_{10}, s_{21})$ or $s_{16} \in \pi(s_{10}, s_{21})$. In the former case, $\delta(s_{10}, s_{21}) \geq |\varrho(10, 8, 0, 21)|/|s_{10}s_{21}| \geq f(2, 8, 2) = 1.4308\ldots$ We may thus assume that $s_{16} \in \pi(s_{10}, s_{21})$.

Similarly, for the pair s_3, s_{14} either $s_0 \in \pi(s_3, s_{14})$ or $s_8 \in \pi(s_3, s_{14})$. If $s_0 \in \pi(s_3, s_{14})$, then $\delta(s_3, s_{14}) \geq |\varrho(3, 0, 16, 14)|/|s_3s_{14}| \geq f(3, 7, 2) = 1.4886\ldots$ Thus, assume that $s_8 \in \pi(s_3, s_{14})$.

If at least one of s_{12}, s_{13}, or s_{14} is in $\pi(s_{10}, s_{21})$, then

$$\begin{aligned}\delta(s_{10}, s_{21}) &\geq \frac{\min(|\varrho(10, 12, 16, 21)|, |\varrho(10, 13, 16, 21)|, |\varrho(10, 14, 16, 21)|)}{|s_{10}s_{21}|} \\ &\geq \min(f(2, 4, 5), f(3, 3, 5), f(4, 2, 5)) = f(2, 4, 5) = 1.4237\ldots\end{aligned}$$

Otherwise, one of $s_{10}s_{15}, s_{10}s_{16}, s_{11}s_{15}$, or $s_{11}s_{16}$ must be in $\pi(s_{10}, s_{21})$, and

$$\begin{aligned}\delta(s_3, s_{14}) &\geq \frac{\min(|\varrho(3,8,10,14)|, |\varrho(3,8,11,14)|), |\varrho(3,8,15,14)|)}{|s_3 s_{14}|} \\ &\geq \min(f(5,2,4), f(5,3,3), f(1,5,7)) = f(2,4,5) = 1.4237\ldots \qquad \square\end{aligned}$$

Lemma 1.3. *If $\ell = 9$, then $\delta(T) \geq f(2,4,5) = 1.4237\ldots$*

Proof. Let $s_0 s_9$ be the longest chord and consider the triangle with base $s_0 s_9$ and the third vertex in $\texttt{lower}(s_0 s_9)$. There are three possible cases depending on the convex hull lengths of other two sides of the triangle: $\{7,7\}$, $\{8,6\}$ or $\{9,5\}$. We consider them successively.

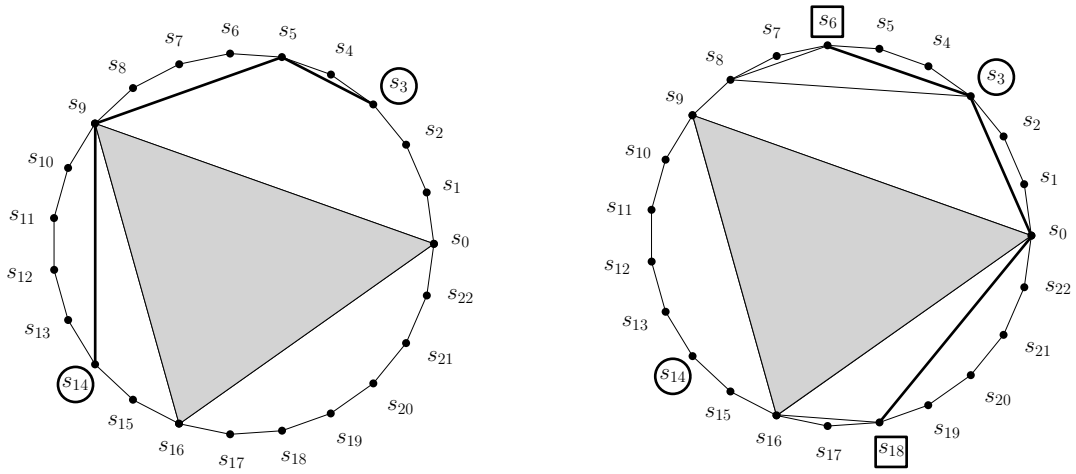

Figure 1.3: Illustrating Case A from Lemma 1.3. Left: $s_5 \in \pi(s_3, s_{14})$. Right: $s_3 s_8 \in \pi(s_3, s_{14})$.

Case A: The convex hull lengths of the other two sides are $\{7,7\}$. Let $\Delta s_0 s_9 s_{16}$ be the required triangle; refer to Fig. 1.3. In this case, the primary pair is s_3, s_{14} and the secondary pair is s_6, s_{18}. Either $s_0 \in \pi(s_3, s_{14})$ or $s_9 \in \pi(s_3, s_{14})$. If $s_0 \in \pi(s_3, s_{14})$, then $\delta(s_3, s_{14}) \geq |\varrho(3,0,16,14)|/|s_3 s_{14}| \geq f(3,7,2) = 1.4886\ldots$ Thus, we assume that $s_9 \in \pi(s_3, s_{14})$.

Similarly, for the pair s_6, s_{18}, either $s_9 \in \pi(s_6, s_{18})$ or $s_0 \in \pi(s_6, s_{18})$. If $s_9 \in \pi(s_6, s_{18})$ then $\delta(s_6, s_{18}) \geq |\varrho(6,9,16,18)|/|s_6 s_{18}| \geq f(3,7,2) = 1.4886\ldots$ Thus, assume that $s_0 \in \pi(s_6, s_{18})$.

Now, if at least one of s_5, s_6, or s_7 is in $\pi(s_3, s_{14})$, then

$$\begin{aligned}\delta(s_3, s_{14}) &\geq \frac{\min(|\varrho(3,5,9,14)|, |\varrho(3,6,9,14)|, |\varrho(3,7,9,14)|)}{|s_3 s_{14}|} \\ &\geq \min(f(2,4,5), f(3,3,5), f(4,2,5)) = f(2,4,5) = 1.4237\ldots\end{aligned}$$

Otherwise, one of s_3s_8, s_3s_9, s_4s_8, or s_4s_9 must be in $\pi(s_3, s_{14})$, and

$$\begin{aligned}\delta(s_6, s_{18}) &\geq \frac{\min(|\varrho(6,3,0,18)|, |\varrho(6,4,0,18)|)}{|s_6 s_{18}|} \\ &\geq \min(f(3,3,5), f(2,4,5)) = f(2,4,5) = 1.4237\ldots\end{aligned}$$

CASE B: The convex hull lengths of the other two sides are $\{8, 6\}$. Let $\Delta s_0 s_9 s_{17}$ be the required triangle; refer to Fig. 1.4 (left). As in CASE A, the primary pair is s_3, s_{14} and the secondary pair is s_6, s_{18}. Consider the pair s_3, s_{14}. Either $s_{17} \in \pi(s_3, s_{14})$ or $s_9 \in \pi(s_3, s_{14})$. If $s_{17} \in \pi(s_3, s_{14})$, then $\delta(s_3, s_{14}) \geq |\varrho(3,0,17,14)|/|s_3 s_{14}| \geq f(3,6,3) = 1.5312\ldots$ So we assume that $s_9 \in \pi(s_3, s_{14})$.

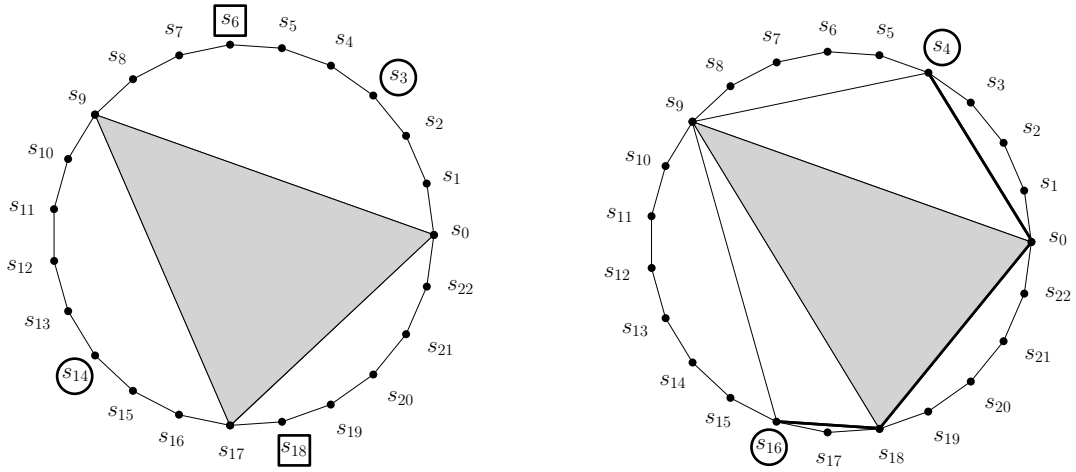

Figure 1.4: Illustrating CASE B (left) and CASE C (right) from Lemma 1.3.

Similarly, for the pair s_6, s_{18}, either $s_9 \in \pi(s_6, s_{18})$ or $s_0 \in \pi(s_6, s_{18})$. If $s_9 \in \pi(s_6, s_{18})$ then $\delta(s_6, s_{18}) \geq |\varrho(6,9,17,18)|/|s_6 s_{18}| \geq f(3,8,1) = 1.4257\ldots$ Thus, we assume that $s_0 \in \pi(s_6, s_{18})$. Now, it can be checked that by the same analysis as in CASE A, the same lower bound

of $f(2,4,5)$ holds.

CASE C: The convex hull lengths of the other two sides of the triangle are $\{9,5\}$. Let $\Delta s_0 s_9 s_{18}$ be the required triangle; refer to Fig. 1.4 (right). Then,

$$\begin{aligned}\delta(s_4, s_{16}) &\geq \frac{\min(|\varrho(4,0,18,16)|, |\varrho(4,9,16)|)}{|s_4 s_{16}|} \\ &\geq \min(f(4,5,2), f(5,7)) = f(2,4,5) = 1.4237\ldots\end{aligned}$$ □

Lemma 1.4. *If $\ell = 10$, then $\delta(T) \geq f(2,4,5) = 1.4237\ldots$*

Proof. Let $s_0 s_{10}$ be the longest chord. The possible convex hull lengths of the other two sides of the triangle with base $s_0 s_{10}$ and the third vertex in $\texttt{lower}(s_0 s_{10})$ are $\{10,3\}$,$\{9,4\}$,$\{8,5\}$,$\{7,6\}$. We consider these cases successively.

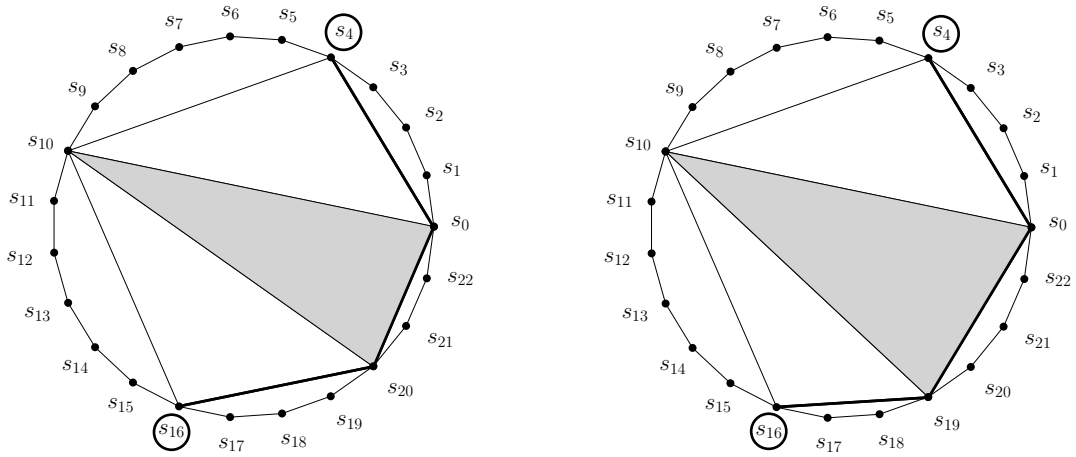

Figure 1.5: Illustrating CASE A from Lemma 1.4.

CASE A: The convex hull lengths of the other two sides of the triangle are $\{10,3\}$, $\{9,4\}$ or $\{8,5\}$. Let $\Delta s_0 s_{10} s_{20}$, $\Delta s_0 s_{10} s_{19}$, $\Delta s_0 s_{10} s_{18}$ be the required triangles, respectively; refer to Fig. 1.5 and Fig. 1.6 (left). Then,

$$\begin{aligned}\delta(s_4, s_{16}) &\geq \frac{\min(|\varrho(4,0,20,16)|, |\varrho(4,0,19,16)|, |\varrho(4,0,18,16)|, |\varrho(4,10,16)|)}{|s_4 s_{16}|} \\ &\geq \min(f(4,3,4), f(4,4,3), f(4,5,2), f(6,6)) = f(2,4,5) = 1.4237\ldots\end{aligned}$$

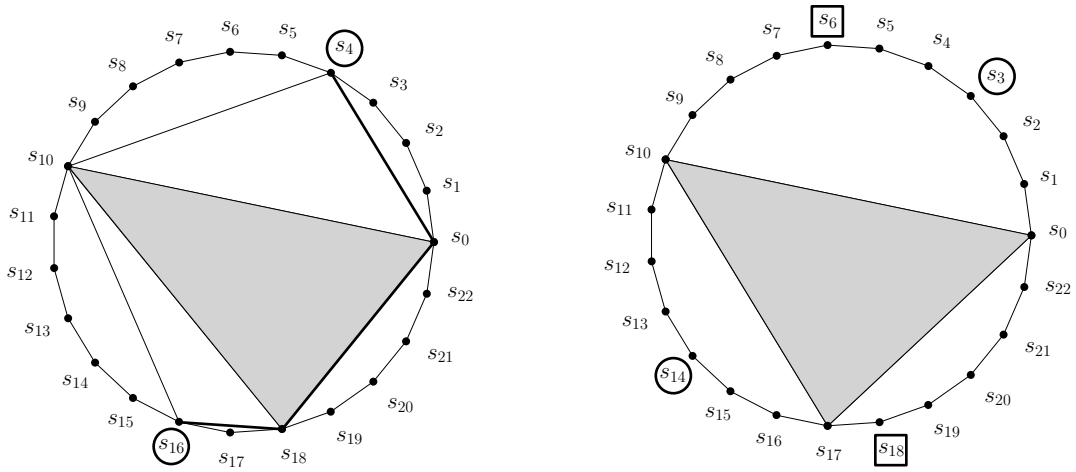

Figure 1.6: Illustrating CASE A (left) and CASE B (right) from Lemma 1.4.

CASE B: The convex hull lengths of the other two sides are $\{7, 6\}$. Let $\Delta s_0 s_{10} s_{17}$ be the required triangle; refer to Fig. 1.6 (right). In this case, the primary pair is s_3, s_{14} and the secondary pair is s_6, s_{18}. Either $s_0 \in \pi(s_3, s_{14})$ or $s_{10} \in \pi(s_3, s_{14})$. If $s_0 \in \pi(s_3, s_{14})$, then $\delta(s_3, s_{14}) \geq |\varrho(3, 0, 17, 14)|/|s_3 s_{14}| \geq f(3, 6, 3) = 1.5312\ldots$ Thus, we assume that $s_{10} \in \pi(s_3, s_{14})$.

Similarly, for the pair s_6, s_{18}, either $s_{10} \in \pi(s_6, s_{18})$ or $s_0 \in \pi(s_6, s_{18})$. If $s_{10} \in \pi(s_6, s_{18})$, then $\delta(s_6, s_{18}) \geq |\varrho(6, 10, 17, 18)|/|s_6 s_{18}| \geq f(4, 7, 1) = 1.4761\ldots$ Thus, assume that $s_0 \in \pi(s_6, s_{18})$.

Now, if at least one of s_5, s_6, s_7, or s_8 is in $\pi(s_3, s_{14})$, then

$$\begin{aligned}\delta(s_3, s_{14}) &\geq \frac{\min(|\varrho(3, 5, 10, 14)|, |\varrho(3, 6, 10, 14)|, |\varrho(3, 7, 10, 14)|, |\varrho(3, 8, 10, 14)|)}{|s_3 s_{14}|} \\ &\geq \min(f(2, 5, 4), f(3, 4, 4), f(4, 3, 4), f(5, 2, 4)) = f(2, 4, 5) = 1.4237\ldots\end{aligned}$$

Otherwise, one of $s_3 s_9, s_3 s_{10}, s_4 s_9$, or $s_4 s_{10}$ must be in $\pi(s_3, s_{14})$, and

$$\begin{aligned}\delta(s_6, s_{18}) &\geq \frac{\min(|\varrho(6, 3, 0, 18)|, |\varrho(6, 4, 0, 18)|)}{|s_6 s_{18}|} \\ &\geq \min(f(3, 3, 5), f(2, 4, 5)) = f(2, 4, 5) = 1.4237\ldots\end{aligned}$$ □

Lemma 1.5. *If* $\ell = 11$, *then* $\delta(T) \geq f(2, 4, 5) = 1.4237\ldots$

Proof. Let s_0s_{11} be the longest chord. Since the size of $\texttt{upper}(s_0s_{11})$ is smaller than the size of $\texttt{lower}(s_0s_{11})$, we consider $\texttt{upper}(s_0s_{11})$ is our analysis. The possible convex hull lengths of the other two sides of the triangle with base s_0s_{11} and the third vertex in $\texttt{upper}(s_0s_{11})$ are $\{1, 10\}$, $\{2, 9\}$, $\{3, 8\}$, $\{4, 7\}$, $\{5, 6\}$. We consider the following cases successively.

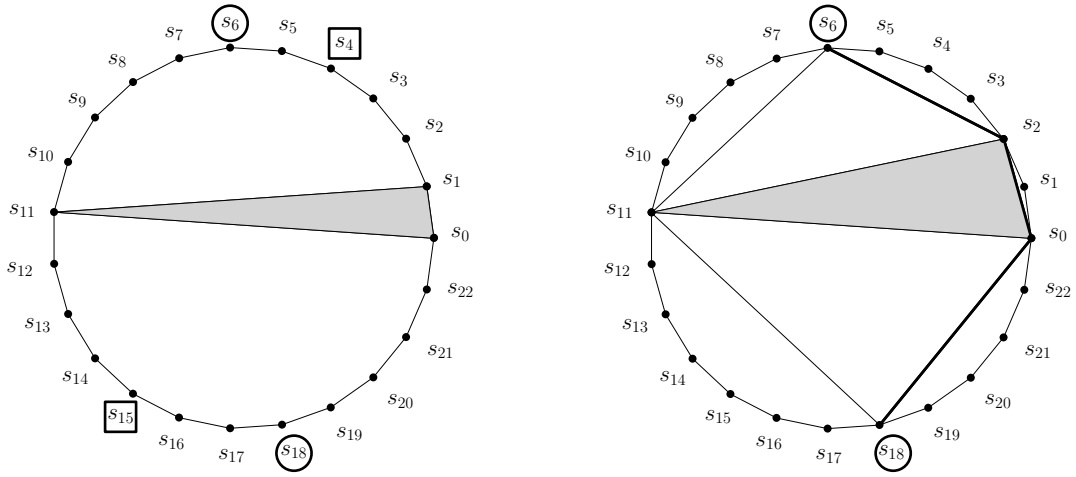

Figure 1.7: Illustrating CASE A (left) and CASE B (right) from Lemma 1.5.

CASE A: The convex hull lengths of the other two sides are $\{1, 10\}$. Let $\Delta s_0s_1s_{11}$ be the required triangle; refer to Fig. 1.7 (left). In this case, the primary pair is s_6, s_{18} and the secondary pair is s_4, s_{15}. Consider the pair s_6, s_{18}. Either $s_{11} \in \pi(s_6, s_{18})$ or $s_0 \in \pi(s_6, s_{18})$. If $s_{11} \in \pi(s_6, s_{18})$, then $\delta(s_6, s_{18}) \geq |\varrho(6, 11, 18)|/|s_6s_{18}| \geq f(5, 7) = 1.4514\ldots$ Hence, we assume that $s_0 \in \pi(s_6, s_{18})$.

If at least one of s_2, s_3, s_4, or s_5 is in $\pi(s_6, s_{18})$, then

$$\begin{aligned}\delta(s_6, s_{18}) &\geq \frac{\min(|\varrho(6, 2, 1, 0, 18)|, |\varrho(6, 3, 1, 0, 18)|, |\varrho(6, 4, 1, 0, 18)|, |\varrho(6, 5, 1, 0, 18)|)}{|s_6s_{18}|} \\ &\geq \min(f(4, 1, 1, 5), f(3, 2, 1, 5), f(2, 3, 1, 5), f(1, 4, 1, 5)) = f(1, 1, 4, 5) = 1.4263\ldots\end{aligned}$$

Otherwise, s_1s_6 is in $\pi(s_6, s_{18})$, and then

$$\begin{aligned}\delta(s_4, s_{15}) &\geq \frac{\min(|\varrho(4,6,11,15)|, |\varrho(4,1,0,15)|)}{|s_4s_{15}|} \\ &\geq \min(f(2,5,4), f(3,1,8)) = f(2,4,5) = 1.4237\ldots\end{aligned}$$

CASE B: The convex hull lengths of the other two sides are $\{2,9\}, \{3,8\}$ or $\{4,7\}$. Let $\Delta s_0s_2s_{11}$, $\Delta s_0s_3s_{11}$, $\Delta s_0s_4s_{11}$ be the required triangles, respectively. Refer to Fig. 1.7 (right) and Fig. 1.8 (left) for illustrations.

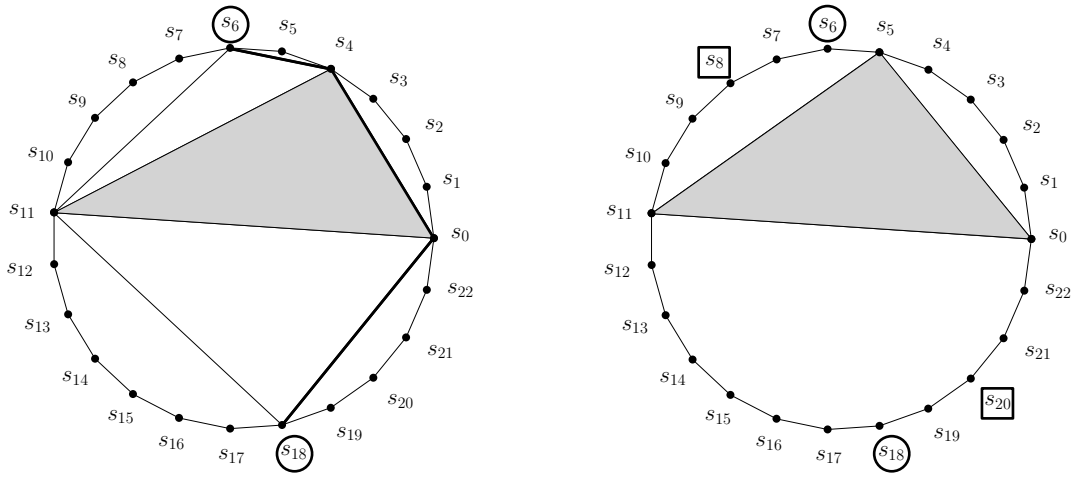

Figure 1.8: Illustrating CASE B (left) and CASE C (right) from Lemma 1.5.

As in the CASE A, we may assume that $s_0 \in \pi(s_6, s_{18})$. Then,

$$\begin{aligned}\delta(s_6, s_{18}) &\geq \frac{\min(|\varrho(6,2,0,18)|, |\varrho(6,3,0,18)|, |\varrho(6,4,0,18)|)}{|s_6s_{18}|} \\ &\geq \min(f(4,2,5), f(3,3,5), f(2,4,5)) = f(2,4,5) = 1.4237\ldots\end{aligned}$$

CASE C: The convex hull lengths of the other two sides are $\{5,6\}$. Let $\Delta s_0s_5s_{11}$ be the required triangle; refer to Fig. 1.8 (right). In this case, the primary pair is s_6, s_{18} and the secondary pair is s_8, s_{20}. As in the CASE A, we assume that $s_0 \in \pi(s_6, s_{18})$.

Now, if at least one of s_{19}, s_{20}, s_{21}, or s_{22} is in $\pi(s_6, s_{18})$, then

$$\begin{aligned}\delta(s_6, s_{18}) &\geq \frac{\min(|\varrho(6,5,0,19,18)|, |\varrho(6,5,0,20,18)|, |\varrho(6,5,0,21,18)|, |\varrho(6,5,0,22,18)|)}{|s_6 s_{18}|} \\ &\geq \min(f(1,5,4,1), f(1,5,3,2), f(1,5,2,3), f(1,5,1,4)) = f(1,1,4,5) = 1.4263\ldots\end{aligned}$$

Otherwise, $s_0 s_{18}$ is in $\pi(s_6, s_{18})$, and then

$$\begin{aligned}\delta(s_8, s_{20}) &\geq \frac{\min(|\varrho(8,5,0,20)|, |\varrho(8,11,18,20)|)}{|s_8 s_{20}|} \\ &\geq \min(f(3,5,3), f(3,7,2)) = f(3,3,5) = 1.4312\ldots\end{aligned}$$ □

Putting these facts together yields the main result of this section:

Theorem 1.1. *Let S be a set of 23 points placed at the vertices of a regular 23-gon. Then*

$$\delta_0(S) = f(2,2,8) = \left(2\sin\frac{2\pi}{23} + \sin\frac{8\pi}{23}\right) \Big/ \sin\frac{11\pi}{23} = 1.4308\ldots$$

Proof. By Lemmas 1.2-1.5, we conclude that

$$\delta_0(S) \geq f(2,4,5) = \left(\sin\frac{2\pi}{23} + \sin\frac{4\pi}{23} + \sin\frac{5\pi}{23}\right) / \sin\frac{11\pi}{23} = 1.4237\ldots$$

On the other hand, the triangulation of S in Fig. 1.1 (right) has stretch factor $f(2,2,8) = 1.4308\ldots$ and thus $f(2,4,5) = 1.4237\ldots \leq \delta_0(S) \leq f(2,2,8) = 1.4308\ldots$

A parallel C++ program (see Appendix) that generates all triangulations of S based on a low memory algorithm by Parvez et al. [36, Section 4] shows that each of the C_{21} triangulations has stretch factor at least $f(2,2,8)$. We thereby obtain the following final result: $\delta_0(S) = f(2,2,8) = (2\sin\frac{2\pi}{23} + \sin\frac{8\pi}{23})/\sin\frac{11\pi}{23} = 1.4308\ldots$ □

Remarks. Using the program we have also checked that the next largest stretch factor among all triangulations is $f(3,3,5) = 1.4312\ldots$, and further that there is no triangulation of S that

has stretch-factor < 1.4312 other than $f(2, 2, 8)$. Thus, the result in Theorem 1.1 is not affected by floating-point precision errors.

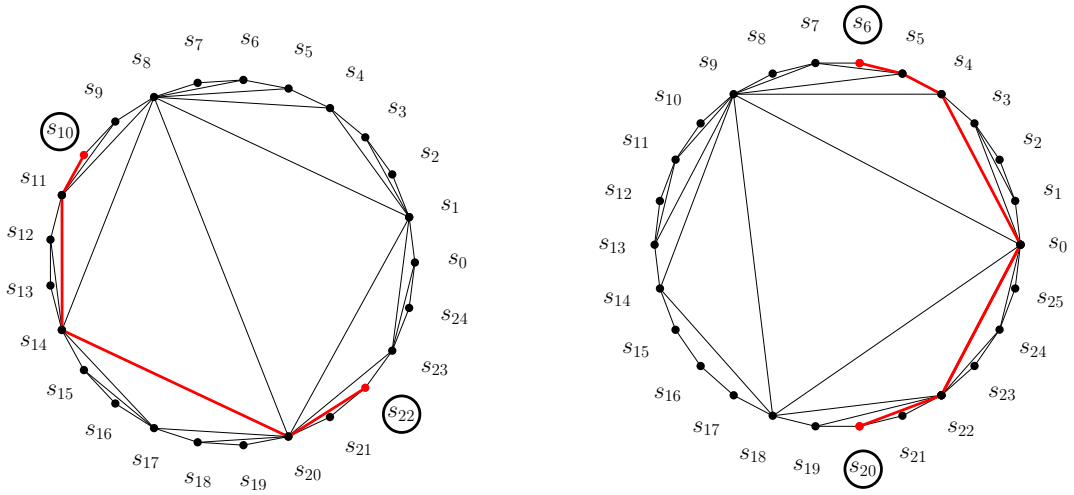

Figure 1.9: Triangulations of S_{25} and S_{26} with stretch factors < 1.4296 and < 1.4202, respectively. Worst stretch factor pairs are marked in circles and the corresponding shortest paths are shown in red.

Let S_n denote the set of points placed at the vertices of a regular n-gon. Using a computer program, Mulzer obtained the values $\delta_0(S_n)$ for $4 \leq n \leq 21$ in his thesis [34, Chapter 3]. Using our C++ program, we confirmed the previous values and extended the range up to $n = 24$: $\delta_0(S_{22}) = 1.4047\ldots$, $\delta_0(S_{24}) = 1.4013\ldots$ and somewhat surprisingly, $\delta_0(S_{23}) = 1.4308\ldots$ By upper bound constructions, it follows that $\delta_0(S_{25}) < 1.4296$ and $\delta_0(S_{26}) < 1.4202$; see Fig. 1.9. Observe that $\delta_0(S_n)$ does not exhibit a monotonic behavior; see Table 1.2.

n	$\delta_0(S_n)$	n	$\delta_0(S_n)$	n	$\delta_0(S_n)$
4	$1.4142\ldots$	12	$1.3836\ldots$	20	$1.4142\ldots$
5	$1.2360\ldots$	13	$1.3912\ldots$	21	$1.4161\ldots$
6	$1.3660\ldots$	14	$1.4053\ldots$	22	$1.4047\ldots$
7	$1.3351\ldots$	15	$1.4089\ldots$	**23**	$\mathbf{1.4308}\ldots$
8	$1.4142\ldots$	16	$1.4092\ldots$	24	$1.4013\ldots$
9	$1.3472\ldots$	17	$1.4084\ldots$	25	< 1.4296
10	$1.3968\ldots$	18	$1.3816\ldots$	26	< 1.4202
11	$1.3770\ldots$	19	$1.4098\ldots$		

Table 1.2: The values of $\delta_0(S_n)$ for $n = 4, \ldots, 26$.

1.3 Lower bounds for the degree 3 and 4 dilation

In this section, we provide lower bounds for the worst case degree 3 and 4 dilation of point sets in the Euclidean plane. We begin with degree 3 dilation. We first present a set P of $n = 13$ points (a section of the hexagonal lattice with six boundary points removed) that has $\delta_0(P, 3) \geq 1+\sqrt{3}$ and then extend P to achieve this lower bound for any $n > 13$.

Theorem 1.2. *For every $n \geq 13$, there exists a set S of n points such that $\delta_0(S, 3) \geq 1 + \sqrt{3} = 2.7321\ldots$ The inequality is tight for the presented sets.*

Proof. Let $P = \{p_0\} \cup P_1 \cup P_2$ be a set of 13 points as shown in Fig. 1.10 (left) where $P_1 = \{p_1, p_3, p_5, p_7, p_9, p_{11}\}$ and $P_2 = \{p_2, p_4, p_6, p_8, p_{10}, p_{12}\}$. The points in P_1 and P_2 lie on the ver-

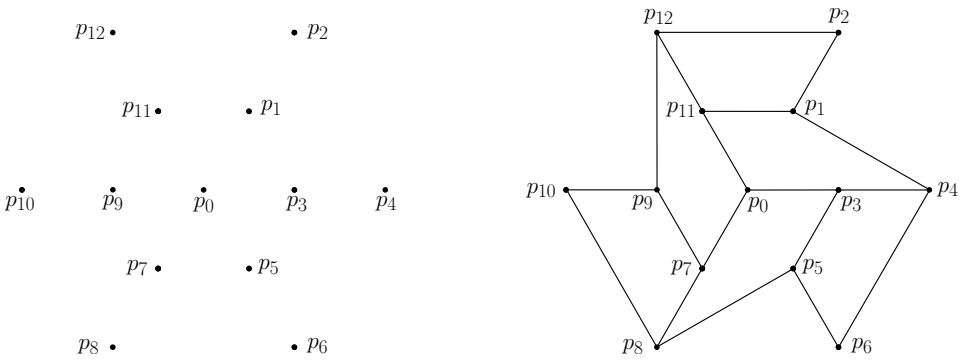

Figure 1.10: Left: the point set $P = \{p_0, p_1, \ldots, p_{12}\}$; some pairwise distances are: $|p_2p_{12}| = 2$, $|p_2p_3| = |p_1p_5| = |p_1p_{12}| = \sqrt{3}$. Right: a plane degree 3 geometric spanner on P with stretch factor $1+\sqrt{3}$, which is achieved by the detours for the point pairs $\{p_1, p_3\}$, $\{p_5, p_7\}$ and $\{p_9, p_{11}\}$.

tices of two regular homothetic hexagons centered at p_0 of radius 1 and 2 respectively. Furthermore, the points in each of the sets $\{p_2, p_1, p_0, p_7, p_8\}$, $\{p_4, p_3, p_0, p_9, p_{10}\}$ and $\{p_{12}, p_{11}, p_0, p_5, p_6\}$ are collinear.

We show that $\delta_0(P, 3) \geq 1 + \sqrt{3}$. Since no edge can contain a point in its interior, the point p_0 can have connecting edges only with the points from P_1. First, assume that the six edges in $E = \{p_1p_3, p_3p_5, p_5p_7, p_7p_9, p_9p_{11}, p_1p_{11}\}$ are present (see Fig. 1.11 (left)).

We can also assume that the edge p_0p_1 is present since p_0 must be connected to at least one

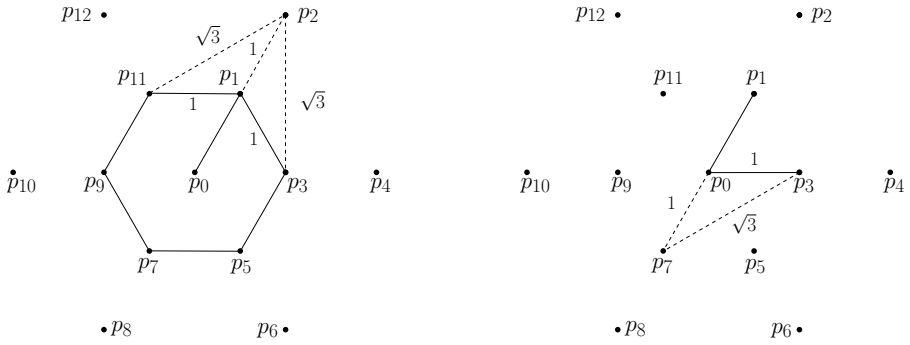

Figure 1.11: Left: all edges in E are present. Right: CASE B.

of the points in P_1. Observe that now $\texttt{deg}(p_1) = 3$. In this case,

$$\delta(p_1, p_2) \geq \frac{|\varrho(1, i, 2)|}{|p_1 p_2|} \geq 1 + \sqrt{3}, \text{ where } i \in \{3, 11\}.$$

Now assume that an edge in E, say p_1p_3, is missing. Then, the following three cases arise depending on $\texttt{deg}(p_0) \in \{1, 2, 3\}$.

CASE A: If $\texttt{deg}(p_0) = 1$, then

$$\delta(p_1, p_3) \geq \frac{|\varrho(1, i, 3)|}{|p_1 p_3|} \geq 1 + \sqrt{3} \text{ where } i \in \{2, 5, 4, 11\}.$$

CASE B: If $\texttt{deg}(p_0) = 2$, consider the edges p_0p_1, p_0p_3; see Fig. 1.11 (right). If p_0p_1, p_0p_3 are present $\delta(p_0, p_7) \geq |\varrho(0, 3, 7)|/|p_0p_7| = 1 + \sqrt{3}$ else if at least one edge in $\{p_0p_1, p_0p_3\}$ is absent then since p_1p_3 is absent, $\delta(p_1, p_3) \geq 1 + \sqrt{3}$ by the same analysis as in CASE A.

CASE C: If $\texttt{deg}(p_0) = 3$, then if at least one of the edges p_0p_1, p_0p_3 is absent, $\delta(p_1, p_3) \geq 1+\sqrt{3}$ as shown in CASE A. Thus, assume that p_0p_1, p_0p_3 are present. Now, the following two non-symmetric cases will arise. Either p_0p_5 is present or p_0p_7 is present.

If p_0p_5 is present (refer to Fig. 1.12 (left)) then,

$$\delta(p_0, p_9) \geq \frac{|\varrho(0, i, 9)|}{|p_0 p_9|} \geq 1 + \sqrt{3}, \text{ where } i \in \{1, 5\}.$$

Now assume that p_0p_7 is present (refer to Fig. 1.12 (right)). Observe that if p_7p_9 is absent

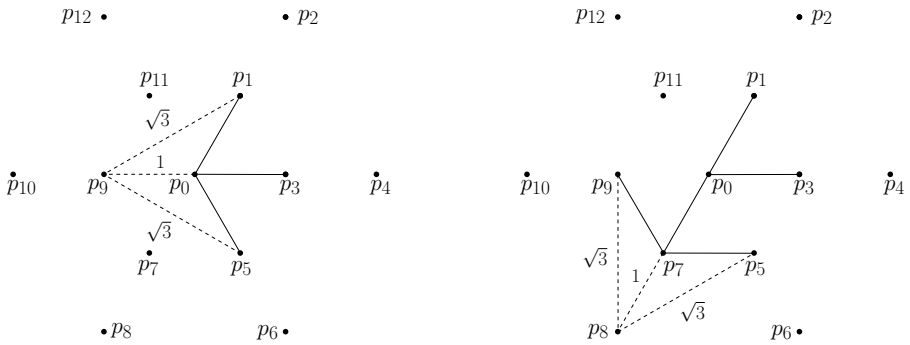

Figure 1.12: Left: the edges p_0p_1, p_0p_3, p_0p_5 are present. Right: the edges p_0p_1, p_0p_3, p_0p_7 are present.

then,

$$\delta(p_7, p_9) \geq \frac{|\varrho(7, i, 9)|}{|p_7p_9|} \geq 1 + \sqrt{3}, \text{ where } i \in \{8, 10, 11\}.$$

Thus, assume that p_7p_9 is present. Similarly, assume that p_5p_7 is present, otherwise

$$\delta(p_7, p_5) \geq \frac{|\varrho(7, i, 5)|}{|p_7p_5|} \geq 1 + \sqrt{3}, \text{ where } i \in \{3, 5, 6\}.$$

Now, as p_0p_7, p_5p_7 and p_7p_9 are present, $\mathtt{deg}(p_7) = 3$. In this case,

$$\delta(p_7, p_8) = \frac{|\varrho(7, 9, 8)|}{|p_7p_8|} \geq 1 + \sqrt{3}, \text{ where } i \in \{5, 9\}.$$

We have thus just shown that $\delta_0(P, 3) \geq 1+\sqrt{3}$. For $n \geq 14$, we may assume that $p_0 = (0, 0)$, $p_3 = (1, 0)$, and let $p_i = (x + i, 0)$ for $i = 13, \ldots, n - 1$, where $x \gg 1$ (e.g., setting $x = 100$ suffices); finally, let $S = P \cup P'$, where $P' = \{p_{13}, \ldots, p_{n-1}\}$. See Fig. 1.13.

If $u, v \in P \subset S$, then going from u to v via P' is inefficient, so as shown earlier in this proof, $\delta(u, v) \geq 1 + \sqrt{3}$. Thus, $\delta_0(S, 3) \geq 1 + \sqrt{3}$, as required. Moreover, this lower bound is tight for both P and S; see Fig. 1.11 (right). □

Remark. If Λ is the infinite hexagonal lattice, it is shown in [21] that $\delta_0(\Lambda, 3) = 1 + \sqrt{3}$.

We now continue with degree 4 dilation. We first exhibit a point set P of $n = 6$ points with degree 4 dilation $1 + \sqrt{(5 - \sqrt{5})/2}$, and then extend it so to achieve the same lower bound

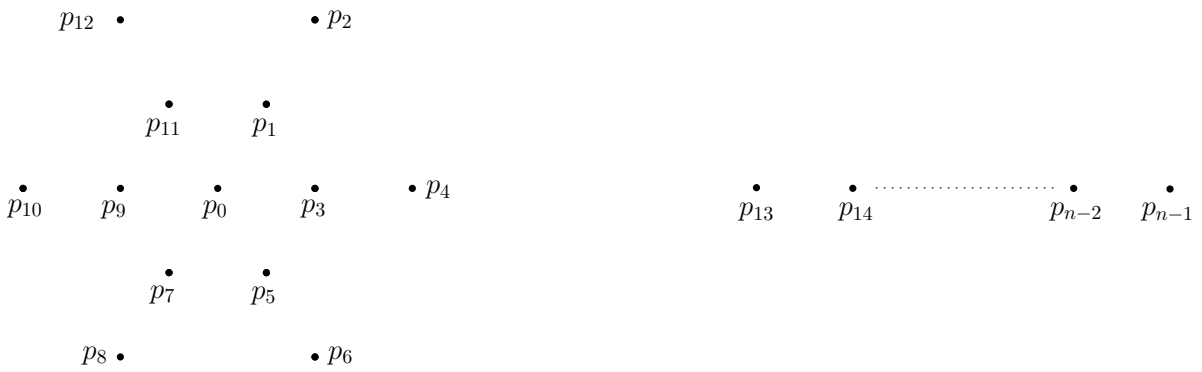

Figure 1.13: A set of $n \geq 13$ points with degree 3 dilation $1 + \sqrt{3}$. The figure is not drawn to scale.

for any larger n. Consider the 6-element point set $P = \{p_0, \ldots, p_5\}$, where $p_1, \ldots, p_5$ are the vertices of a regular pentagon centered at p_0.

Theorem 1.3. *For every $n \geq 6$, there exists a set S of n points such that*

$$\delta_0(S, 4) \geq 1 + \sqrt{(5 - \sqrt{5})/2} = 2.1755\ldots$$

The inequality is tight for the presented sets.

Proof. Assume that $p_1, \ldots, p_5$ lie on a circle of unit radius centered at p_0. Since $\deg(p_0) \leq 4$, there exists a point $p_i, 1 \leq i \leq 5$ such that p_0p_i is not present; we may assume that $i = 1$; see Fig. 1.14. Observe that

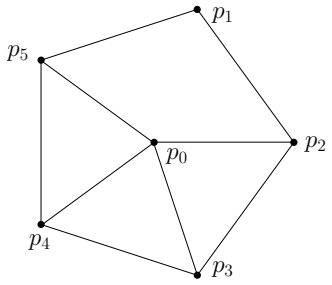

Figure 1.14: A plane degree 4 geometric graph on the point set $\{p_0, \ldots, p_5\}$ that has stretch factor exactly $1 + \sqrt{(5 - \sqrt{5})/2}$, which is achieved by the detour between the pair p_0, p_1.

$$|p_0p_1| = 1 \text{ and } |p_1p_2| = |p_1p_5| = \sqrt{1^2 + 1^2 - 2 \cdot 1 \cdot 1 \cos(2\pi/5)} = \sqrt{(5 - \sqrt{5})/2}.$$

Now,

$$\delta(p_0, p_1) \geq \frac{|\varrho(0, i, 1)|}{|p_0 p_1|} \geq 1 + \sqrt{(5 - \sqrt{5})/2} = 2.1755\ldots, \text{ where } i \in \{2, 5\}.$$

Thus, $\delta_0(P, 4) \geq 1 + \sqrt{(5 - \sqrt{5})/2}$. As in the proof of Theorem 1.2, the aforesaid six points can be used to obtain the same lower bound for any $n \geq 6$.

To see that the above lower bound is tight, consider the degree 4 geometric graph on P in Fig. 1.14 whose stretch factor is exactly that, due to the detour between p_0, p_1. □

1.4 A lower bound on the dilation of the greedy triangulation

In this section, we present a lower bound on the worst case dilation of the greedy triangulation. Place four points at the vertices of a unit square U, and two other points in the exterior of U on the vertical line through the center of U and close to the lower and upper sides of U, as shown in Fig. 1.15 (left). For any small $\varepsilon > 0$, the points can be placed so that the resulting stretch factor is at least $\delta(p_0, p_3) \geq 2 - \varepsilon$. A modification of this idea gives a slightly better lower bound.

Theorem 1.4. *For every $n \geq 6$, there exists a set S of n points such that the stretch factor of the greedy triangulation of S is at least* 2.0268.

Proof. Replace the unit square by a parallelogram V with two horizontal unit sides, unit height and angle $\alpha \in (\pi/4, \pi/2)$ to be determined, as shown in Fig. 1.15 (right). Place four points at the vertices of V and two other points in the exterior of V on the vertical line through the center of the V and close to the lower and upper side of V. First, observe that the greedy triangulation is unique for this point set. Second, observe that there are two candidate detours connecting p_0 with p_3: one of length (slightly longer than) $1 + a$ and one of length (slightly longer than) $2x + b$, where a is the length of the slanted side of V, b is the length of the short diagonal of V, and x is the horizontal distance between the upper left corner and the center of V.

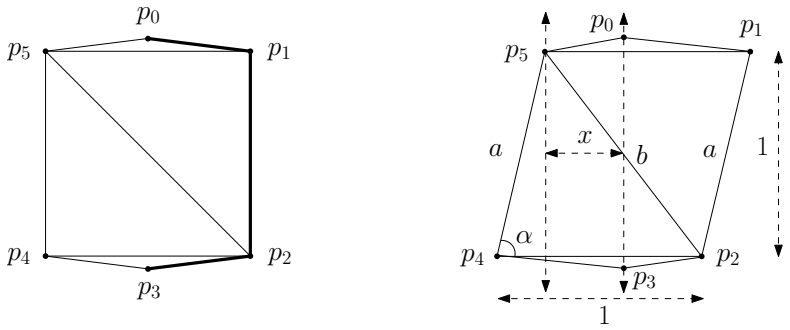

Figure 1.15: Greedy triangulation of 6 points with stretch factors $2-\varepsilon$ (left) and 2.0268 (right).

A straightforward calculation gives:

$$a = \frac{1}{\sin\alpha},\ b = \frac{\sqrt{1+\sin^2\alpha-2\sin\alpha\cos\alpha}}{\sin\alpha},\ \text{and}\ x = \frac{1-\cot\alpha}{2}.$$

Let $f(\alpha) = \min\left(1+\frac{1}{\sin\alpha}, 1-\cot\alpha+\frac{\sqrt{1+\sin^2\alpha-2\sin\alpha\cos\alpha}}{\sin\alpha}\right)$, for $\alpha\in\left(\frac{\pi}{4},\frac{\pi}{2}\right)$.

Setting $\alpha = 1.3416$ (i.e., $\alpha = 76.87°$) yields

$$\delta(p_0,p_3) \geq \max_{\alpha\in(\pi/4,\pi/2)} f(\alpha) \geq f(1.3416) = 2.0268\ldots,$$

as required. As in the proofs of Theorems 1.2 and 1.3, the lower bound can be extended for every $n \geq 6$ in a straightforward way. □

1.5 Concluding remarks

In Section 1.2, we have shown that any plane spanning graph of the vertices of a regular 23-gon requires a stretch factor of $(2\sin\frac{2\pi}{23}+\sin\frac{8\pi}{23})/\sin\frac{11\pi}{23} = 1.4308\ldots$ Henceforth, the question of Bose and Smid [11, Open Problem 1] mentioned in the Introduction can be restated:

Problem 1.1. *Does there exist a point set S in the Euclidean plane such that $\delta_0(S) > (2\sin\frac{2\pi}{23}+\sin\frac{8\pi}{23})/\sin\frac{11\pi}{23} = 1.4308\ldots$?*

Next in Section 1.3, it has been shown that there exist point sets that require degree 3 dila-

tion $1+\sqrt{3}=2.7321\ldots$ (Theorem 1.2) and degree 4 dilation $1+\sqrt{(5-\sqrt{5})/2}=2.1755\ldots$ (Theorem 1.3). Perhaps these lower bounds can be improved.

Problem 1.2. *Does there exist a point set in the Euclidean plane that has degree 3 dilation greater than $1+\sqrt{3}$? Does there exist a point set in the Euclidean plane that has degree 4 dilation greater than $1+\sqrt{(5-\sqrt{5})/2}$?*

Finally in Section 1.4, we show that the stretch factor of the greedy triangulation is at least 2.0268, in the worst case. Perhaps this lower bound is not far from the truth. Using a computer program we have generated 1000 random uniformly distributed n-element point sets in a unit square for every n in the range $4 \leq n \leq 250$, and computed the greedy triangulations and corresponding stretch factors. The highest stretch factor among these was only 1.97 (as attained for a 168-element point set), and so this suggests the following.

Problem 1.3. *Is the worst case stretch factor of the greedy triangulation attained by points in convex position?*

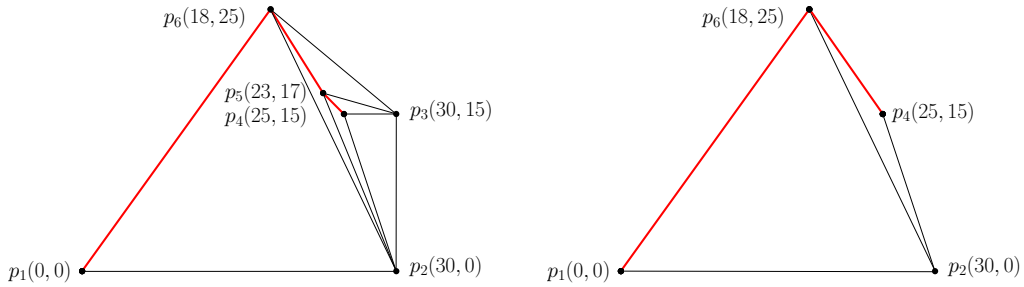

Figure 1.16: Left: greedy triangulation of a set of 6 points not in convex position with stretch factor $\Delta = 1.4772\ldots$ attained by the pair $\{p_1, p_4\}$. Right: the largest stretch factor of the greedy triangulation of a convex subset is that for the subset $S' = \{p_1, p_2, p_4, p_6\}$; it is attained by the same pair $\{p_1, p_4\}$ and equals $1.4753\ldots < \Delta$. The corresponding shortest paths are drawn in red color.

Observe that the point set used in the lower bound construction in Theorem 1.4 is convex, so it is natural to ask: given a non-convex point set S and a greedy triangulation of S having stretch factor Δ, does there always exist a convex subset $S' \subset S$ such that the stretch factor of a greedy triangulation for S' also equals Δ? The point set $S = \{p_1, \ldots, p_6\}$ illustrated in

Fig. 1.16 shows that this is not the case. It is routine to verify that the stretch factor of the greedy triangulation of each convex subset $S' \subset S$ is at most $1.4753\ldots < \Delta = 1.4772\ldots$

Acknowledgment. We convey our special thanks to an anonymous reviewer for suggesting an elegant simplification of the case analysis in the proof of Theorem 1.1. We express our satisfaction with the software package *JSXGraph, Dynamic Mathematics with JavaScript* and the OpenMP API, used in our experiments.

References

[1] P. K. Agarwal, R. Klein, C. Knauer, S. Langerman, P. Morin, M. Sharir, and M. Soss, Computing the detour and spanning ratio of paths, trees, and cycles in 2D and 3D, *Discrete Comput. Geom.* **39(1-3)** (2008), 17–37.

[2] I. Althöfer, G. Das, D. P. Dobkin, D. Joseph, and J. Soares, On sparse spanners of weighted graphs, *Discrete Comput. Geom.* **9** (1993), 81–100.

[3] N. Amarnadh and P. Mitra, Upper bound on dilation of triangulations of cyclic polygons, In *Proc. Conf. Comput. Science and Appl.*, 2006, Springer, pp. 1–9.

[4] B. Aronov, M. de Berg, O. Cheong, J. Gudmundsson, H. J. Haverkort, and A. Vigneron, Sparse geometric graphs with small dilation, *Comput. Geom.* **40(3)** (2008), 207–219.

[5] N. Bonichon, C. Gavoille, N. Hanusse, and L. Perković, Plane spanners of maximum degree six, In *Proc. Internat. Colloq. Automata, Lang. and Prog.*, Springer, 2010, pp. 19–30.

[6] N. Bonichon, I. Kanj, L. Perković, and G. Xia, There are plane spanners of degree 4 and moderate stretch factor, *Discrete Comput. Geom.* **53(3)** (2015), 514–546.

[7] P. Bose, P. Carmi, and L. Chaitman-Yerushalmi, On bounded degree plane strong geometric spanners, *J. Discrete Algorithms* **15** (2012), 16–31.

[8] P. Bose, L. Devroye, M. Löffler, J. Snoeyink, and V. Verma, Almost all Delaunay triangulations have stretch factor greater than $\pi/2$, *Comput. Geom.* **44(2)** (2011), 121–127.

[9] P. Bose, J. Gudmundsson, and M. Smid, Constructing plane spanners of bounded degree and low weight, *Algorithmica* **42** (2005), 249–264.

[10] P. Bose, A. Lee, and M. Smid, On generalized diamond spanners, in *Proc. 10th Internat. Workshop on Algorithms and Data Structures*, vol 4619 of LNCS, Springer, 2007, pp. 325—336.

[11] P. Bose and M. Smid, On plane geometric spanners: A survey and open problems, *Comput. Geom.* **46(7)** (2013), 818–830.

[12] P. Bose, M. Smid, and D. Xu, Delaunay and diamond triangulations contain spanners of bounded degree, *Internat. J. Comput. Geom. Appl.* **19(2)** (2009), 119–140.

[13] B. Chandra, G. Das, G. Narasimhan, and J. Soares, New sparseness results on graph spanners, *Internat. J. Comput. Geom. Appl.* **5** (1995), 125–144.

[14] O. Cheong, H. Herman, and M. Lee, Computing a minimum-dilation spanning tree is NP-hard, *Comput. Geom.* **41(3)** (2008), 188–205.

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

[16] S. Cui, I. Kanj, and G. Xia, On the stretch factor of Delaunay triangulations of points in convex position, *Comput. Geom.* **44(2)** (2011), 104–109.

[17] G. Das and P. Heffernan, Constructing degree-3 spanners with other sparseness properties, *Internat. J. Found. Comput. Sci.* **7(2)** (1996), 121–136.

[18] G. Das and D. Joseph, Which triangulations approximate the complete graph? In *Proc. International Sympos. on Optimal Algorithms*, vol 401 of LNCS, Springer, 1989, pp. 168–192.

[19] D. P. Dobkin, S. J. Friedman, and K. J. Supowit, Delaunay graphs are almost as good as complete graphs, *Discrete Comput. Geom.* **5** (1990), 399–407.

[20] A. Dumitrescu, A. Ebbers-Baumann, A. Grüne, R. Klein, and G. Rote, On the geometric dilation of closed curves, graphs, and point sets, *Comput. Geom.* **36** (2006), 16–38.

[21] A. Dumitrescu and A. Ghosh, Lattice spanners of low degree, preprint, 2016, `arXiv:1602.04381.v1`. A preliminary version in *Proc. Internat. Conf. Algor. and Discrete Appl. Math.*, 2016, vol. 9602 of LNCS, pp. 152–163.

[22] A. Ebbers-Baumann, A. Grüne, and R. Klein, On the geometric dilation of finite point sets, *Algorithmica* **44** (2006), 137–149.

[23] A. Ebbers-Baumann, R. Klein, E. Langetepe, and A. Lingas, A fast algorithm for approximating the detour of a polygonal chain, *Comput. Geom.* **27** (2004), 123–134.

[24] D. Eppstein, Spanning trees and spanners, in *Handbook of Computational Geometry* (J. R. Sack and J. Urrutia, editors), North-Holland, Amsterdam, 2000, pp. 425–461.

[25] J. Gudmundsson and C. Knauer, Dilation and detour in geometric networks, in *Handbook on Approximation Algorithms and Metaheuristics, Chap. 52* (T. Gonzalez, editor), Chapman & Hall/CRC, Boca Raton, 2007.

[26] I. Kanj, Geometric spanners: recent results and open directions, in *Proc. 3rd Internat. Conf. Comm. and Inform. Technology*, IEEE, 2013, pp. 78–82.

[27] I. Kanj and L. Perković, On geometric spanners of Euclidean and unit disk graphs, in *Proc. 25th Annual Sympos. on Theoretical Aspects of Computer Science*, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2008, pp. 409–420.

[28] I. Kanj, L. Perković, and D. Türkoğlu, Degree four plane spanners: simpler and better, preprint, 2016, `arXiv:1603.03818.v1`.

[29] M. Keil and C. A. Gutwin, Classes of graphs which approximate the complete Euclidean graph, *Discrete Comput. Geom.* **7** (1992), 13–28.

[30] R. Klein, M. Kutz, and R. Penninger, Most finite point sets in the plane have dilation > 1, *Discrete Comput. Geom.* **53(1)** (2015), 80–106.

[31] C. Knauer and W. Mulzer, An exclusion region for minimum dilation triangulations, in *Proc. 21st Europ. Workshop Comput. Geom.*, Eindhoven, 2005, pp. 33–36.

[32] C. Levcopoulos and A. Lingas, There are planar graphs almost as good as the complete graphs and almost as cheap as minimum spanning trees, *Algorithmica* **8** (1992), 251–256.

[33] X.Y. Li and Y. Wang, Efficient construction of low weighted bounded degree planar spanner, *Internat. J. Comput. Geom. Appl.* **14(1–2)** (2004), 69–84.

[34] W. Mulzer, Minimum dilation triangulations for the regular n-gon, Masters Thesis, Freie Universität Berlin, 2004.

[35] G. Narasimhan and M. Smid, *Geometric Spanner Networks*, Cambridge University Press, 2007.

[36] M. T. Parvez, Md. S. Rahman and S.-I. Nakano, Generating all triangulations of plane graphs, *J. Graph Algorithms Appl.* **15(3)** (2011), 457-482.

[37] G. Xia, The stretch factor of the Delaunay triangulation is less than 1.998, *SIAM J. Comput.*, **42(4)** (2013), 1620–1659.

[38] G. Xia and L. Zhang, Toward the tight bound of the stretch factor of Delaunay triangulations, In *Proc. 23rd Canadian Conf. Comput. Geom. (CCCG)*, 2011.

Appendix

Source code. The following parallel C++ code is written using OpenMP in C++11 (notice the `pragma` directives present in the code). For the set of $N = 23$ points, the program ran for approximately 2 days on a computer with quad core processor. The program was compiled with `g++ 4.9.2`. Please note that older versions of g++ might have issues with OpenMP support. Following is a correct way of compiling the program.

```
g++ program.cpp -std=c++11 -fopenmp -O3
```

The number of threads has been set to 4 using the variable `numberOfThreads` in `main()`. The user may alter the value of the variable depending on the processor.

Following is the output from the program.

```
Execution started...
Triangulations checked:  24466267020
Dilation:  1.4308143191
Time taken:  162829 seconds
```

```
#include <iostream>
#include <cmath>
#include <omp.h>
#include <list>
#include <vector>
#include <chrono>

#define M_PI 3.14159265358979323846

using namespace std;
```

```
using namespace chrono;

struct Edge { unsigned u,v; };
struct gc { unsigned vis_v; Edge oppositeEdge; };

unsigned numberOfThreads;
vector<unsigned long long> countTriangulations;
vector<double> minStretchFactor;
list<unsigned> jobList;

typedef vector<vector<double>> Matrix2D;
vector<Matrix2D> distarrayCollection;

struct Triangulation
{
    unsigned N;
    list<gc> gcList;
    list<Edge> bcList;

    Triangulation(const unsigned numberOfPoints)
    {
        N = numberOfPoints;
        for(unsigned vertexID = 2; vertexID < N-1; vertexID++)
            gcList.push_front({vertexID,{vertexID-1,vertexID+1}});
    }

    void flipgc(const unsigned gchord)
    {
        list<gc>::iterator hold;
```

```
        for(auto it = gcList.begin(); it != gcList.end(); it++)
            if(it->vis_v == gchord)
            {
                bcList.push_front(it->oppositeEdge);

                if(next(it,1) != gcList.end())
                    next(it,1)->oppositeEdge.v = it->oppositeEdge.v;

                if(it != gcList.begin())
                    prev(it,1)->oppositeEdge.u = it->oppositeEdge.u;

                hold = it;
                break;
            }
        gcList.erase(hold);
    }
};

inline double distance(const unsigned p1, const unsigned p2, const unsigned N)
{
    unsigned absVal = max(p1,p2) - min(p1,p2);
    unsigned lambda = min(absVal,N-absVal);
    return 2*sin((lambda*M_PI)/N);
}


void calculateStretchFactorOfTriangulation(const Triangulation T, const unsigned
    thread_ID)
{
```

```
double stretchFactor = 0;

Matrix2D dist = distarrayCollection[thread_ID];

for(unsigned i = 0; i < T.N; i++)
    for(unsigned j = 0; j < T.N; j++)
        dist[i][j] = (i != j)? INFINITY : 0;

for(unsigned i = 0; i < T.N-1; i++)
    dist[i][i+1] = dist[i+1][i] = distance(i,i+1,T.N);
dist[T.N-1][0] = dist[0][T.N-1] = distance(T.N-1,0,T.N);

for(auto it = T.gcList.begin(); it != T.gcList.end() ; it++)
    dist[0][it->vis_v] = dist[it->vis_v][0] = distance(0,it->vis_v,T.N);

for(auto it = T.bcList.begin(); it != T.bcList.end() ; it++)
    dist[it->u][it->v] = dist[it->v][it->u] = distance(it->u,it->v,T.N);

for(unsigned k = 0; k < T.N; k++)
    for(unsigned i = 0; i < T.N; i++)
        for(unsigned j = 0; j < T.N; j++)
            if (dist[i][j] > dist[i][k] + dist[k][j])
                dist[i][j] = dist[i][k] + dist[k][j];

for(unsigned i = 0; i < T.N; i++)
    for(unsigned j = 0; j < T.N && i != j; j++)
    {
        double tempRatio = dist[i][j] / distance(i,j,T.N);
        if(tempRatio > stretchFactor)
```

```
                stretchFactor = tempRatio;
        }

    if(minStretchFactor[thread_ID] > stretchFactor)
        minStretchFactor[thread_ID] = stretchFactor;
}

void findChildren(Triangulation &T, const unsigned gchord,const unsigned
    thread_ID)
{
    T.flipgc(gchord);

    countTriangulations[thread_ID]++;
    calculateStretchFactorOfTriangulation(T,thread_ID);

    if(!T.gcList.empty())
        for(auto it = T.gcList.begin(); it != T.gcList.end(); it++)
            if(it->vis_v >= T.bcList.front().u)
            {
                Triangulation *childTriangulation = new Triangulation(T);
                findChildren(*childTriangulation,it->vis_v,thread_ID);
                delete childTriangulation;
            }
}

void thread_job(const unsigned N, const unsigned thread_ID)
{
    Triangulation *childTriangulation;
    unsigned gchord;
```

```
    while(true)
    {
        gchord = 0;

        #pragma omp critical
        {
            if(jobList.size() > 0)
            {
                gchord = jobList.front();
                jobList.pop_front();
            }
        }

        if(gchord == 0)
            break;

        childTriangulation = new Triangulation(N);
        findChildren(*childTriangulation,gchord,thread_ID);
        delete childTriangulation;
    }
}

void findAllTriangulations(const unsigned N)
{
    Triangulation rootTriangulation(N);
    countTriangulations[0]++;

    calculateStretchFactorOfTriangulation(rootTriangulation,0);
```

```
    #pragma omp parallel for num_threads(numberOfThreads)
    for(unsigned thread_id = 0; thread_id < numberOfThreads; thread_id++)
        thread_job(N,thread_id);

}

void initDataStructures(const unsigned N)
{
    minStretchFactor.assign(numberOfThreads,INFINITY);
    countTriangulations.assign(numberOfThreads,0);

    for(unsigned gc = N-2; gc >= 2; gc--)
        jobList.push_front(gc);

    distarrayCollection.resize(numberOfThreads);
    for(unsigned thread_ID = 0; thread_ID < numberOfThreads; thread_ID++)
    {
        Matrix2D matrix(N);
        for(unsigned pos = 0; pos < N; pos++)
            matrix[pos].resize(N);
        distarrayCollection[thread_ID] = matrix;
    }
}

int main()
{
    unsigned N = 23;
    numberOfThreads = 4;
```

```
    double dilation = INFINITY;
    unsigned long long totalNoOfTriangulations = 0;

    cout << "Execution started...\n";

    auto start = system_clock::now();
    initDataStructures(N);
    findAllTriangulations(N);

    for(unsigned thread_ID = 0; thread_ID < numberOfThreads; thread_ID++)
    {
        if( countTriangulations[thread_ID] > 0 && minStretchFactor[thread_ID] <
            dilation)
            dilation = minStretchFactor[thread_ID];
        totalNoOfTriangulations += countTriangulations[thread_ID];
    }

    printf("Triangulations checked: %lld\n",totalNoOfTriangulations);
    printf("Dilation: %1.10f\n",dilation);

    system_clock::time_point stop = system_clock::now();
    auto duration = duration_cast<seconds>( stop - start ).count();
    cout << "Time taken: " << duration << " seconds" << endl;

    return EXIT_SUCCESS;
}
```

Chapter 2

Lattice spanners of low degree

2.1 Introduction

Let P be a (possibly infinite) set of points in the Euclidean plane. A *geometric graph* embedded on P is a graph $G = (V, E)$ where $V = P$ and an edge $uv \in E$ is the line segment connecting u and v. View G as a edge-weighted graph, where the weight of uv is the Euclidean distance between u and v. A geometric graph G is a *t-spanner*, for some $t \geq 1$, if for every pair of vertices u, v in V, the length of the shortest path $\pi_G(u, v)$ between u and v in G is at most t times $|uv|$, i.e., $\forall u, v \in V, |\pi_G(u, v)| \leq t|uv|$. Obviously, the complete geometric graph on a set of points is a 1-spanner. When there is no need to specify t, the rather imprecise term *geometric spanner* is also used. A geometric spanner G is *plane* if no two edges in G cross. Here we only consider

plane geometric spanners. A geometric spanner of degree at most k is called *degree k geometric spanner*.

Consider a geometric spanner $G = (V, E)$. The *vertex dilation* or *stretch factor* of a pair $u, v \in V$, denoted $\delta_G(u, v)$, is defined as $\delta_G(u, v) = |\pi_G(u, v)|/|uv|$. If G is clear from the context, we simply write $\delta(u, v)$. The *vertex dilation* or *stretch factor* of G, denoted $\delta(G)$, is defined as $\delta(G) = \sup_{u,v \in V} \delta_G(u, v)$. The terms *graph theoretic dilation* and *spanning ratio* are also used [17, 22, 30].

Given a point set P, let the *dilation* of P, denoted by $\delta_0(P)$, be the minimum stretch factor of a plane geometric graph (equivalently, triangulation) on vertex set P; see [29]. Similarly, let the *degree k dilation* of P, denoted by $\delta_0(P, k)$, be the minimum stretch factor of a plane geometric graph of degree at most k on vertex set P. Clearly, $\delta_0(P, k) \geq \delta_0(P)$ holds for any k. Furthermore, $\delta_0(P, j) \geq \delta_0(P, k)$ holds for any $j < k$. (Note that the term *dilation* has been also used with different meanings in the literature, see for instance [8, 23].)

The field of geometric spanners has witnessed a great deal of interest from researchers, both in theory and applications; see for instance the survey articles [8, 19, 20, 30]. For the current status of various open problems in this area, the reader is referred to the web-page maintained by Smid [31].

Typical objectives include constructions of low stretch factor geometric spanners that have few edges, bounded degree, low weight and/or diameter, etc. Geometric spanners find their applications in the areas of robotics, computer networks, distributed systems and many others. Various algorithmic and structural results on sparse geometric spanners can be found in [1, 2, 3, 10, 11, 18, 23, 25].

Chew [12] was the first to show that it is always possible to construct a plane 2-spanner with $O(n)$ edges on a set of n points; more recently, Xia [32] proved a slightly sharper upper bound of 1.998 using Delaunay triangulations. Bose et al. [7] showed that there exists a plane t-spanner of degree at most 27 on any set of points in the Euclidean plane where $t \approx 10.02$. The result was subsequently improved in [4, 9, 6, 21, 26] in terms of degree. Recently, Bonichon et al. [5]

reduced the degree to 4 with $t \approx 156.82$. The question whether the degree can be reduced to 3 remains open at the time of this writing; if one does not insist on having a plane spanner, Das et al. [13] showed that degree 3 is achievable. From the other direction, lower bounds on the stretch factors of plane spanners for finite point sets have been investigated in [15, 23, 29].

It is natural to study the existence of low-degree spanners of fundamental regular structures, such as point lattices. Indeed, these have been the focus of interest since the early days of computing. One such intense research area concerns VLSI [24]. Other applications of spanners (not necessarily geometric) are in the areas of computer networks and parallel computing; see for instance [27, 28]. While the authors of [27, 28] do examine grid structures (including planar ones), the resulting stretch factors however are not defined (or measured) in geometric terms. More recently, lattice structures at a larger scale are used in industrial design, modern urban design and outer space design. Indeed, Manhattan-like layout of facilities and road connections are very convenient to plan and deploy, frequently in an automatic manner. Studying the stretch factors that can be achieved in low degree spanners of point sets with a lattice structure appears to be quite useful. The two most common lattices are the square lattice and the hexagonal lattice.

According to an argument due to Das and Heffernan [13],[30, p. 468], the n points in a $\sqrt{n} \times \sqrt{n}$ section of the integer lattice cannot be connected in a path or cycle with stretch factor $o(\sqrt{n})$, $O(1)$ in particular. Similarly, no degree 2 plane spanner of the infinite integer lattice can have stretch factor $O(1)$, hence a minimum degree of 3 is necessary in achieving a constant stretch factor. The same facts hold for the infinite hexagonal lattice.

Our results. Let Λ be the infinite square lattice. We show that the degree 3 and 4 dilation of this lattice are bounded as follows:

(i) $1 + \sqrt{2} \leq \delta_0(\Lambda, 3) \leq (3 + 2\sqrt{2})\, 5^{-1/2}$ (Theorem 2.1, Section 2.3).

(ii) $\delta_0(\Lambda, 4) = \sqrt{2}$ (Theorem 2.2, Section 2.3).

If Λ is the infinite hexagonal lattice, we show that

(i) $\delta_0(\Lambda, 3) = 1 + \sqrt{3}$ (Theorem 2.3, Section 2.4).

(ii) $\delta_0(\Lambda, 4) = 2$ (Theorem 2.4, Section 2.4).

2.2 Preliminaries

By the well known Cauchy--Schwarz inequality for $n = 2$, if $a, b, x, y \in \mathbf{R}^+$, then

$$g(x, y) = \frac{ax + by}{\sqrt{x^2 + y^2}} \leq \sqrt{a^2 + b^2},$$

and moreover, $g(x, y) = \sqrt{a^2 + b^2}$ when $x/y = a/b$. In this chapter, we will use this inequality in an equivalent form:

Fact 2.1. *Let $a, b, \lambda \in \mathbf{R}^+$. Then $f(\lambda) = \dfrac{a\lambda + b}{\sqrt{\lambda^2 + 1}} \leq \sqrt{a^2 + b^2}$, and moreover, $f(\lambda) = \sqrt{a^2 + b^2}$ when $\lambda = a/b$.*

Notations and assumptions. Let P be a planar point set and $G = (V, E)$ be a plane geometric graph on vertex set P. For $p, q \in P$, pq denotes the connecting segment and $|pq|$ denotes its Euclidean length. The degree of a vertex (point) $p \in V$ is denoted by $\deg(p)$. For a specific point set $P = \{p_1, \ldots, p_n\}$, we denote the shortest path between p_s, p_t in G consisting of vertices in the order $p_s, \ldots, p_t$ using $\varrho(p_s, \ldots, p_t)$ and by $|\varrho(p_s, \ldots, p_t)|$ its total Euclidean length. The graphs we construct have the property that no edge contains a point of P in its interior.

2.3 The square lattice

This section is devoted to the degree 3 and 4 dilation of the square lattice. In [16], we showed that the degree 3 dilation of the infinite square lattice is at most $(7 + 5\sqrt{2})\, 29^{-1/2} = 2.6129\ldots$; also see the Appendix for the same results. Here we improve this upper bound to $\delta_0 := (3 + 2\sqrt{2})\, 5^{-1/2} = 2.6065\ldots$ We believe that this upper bound is the best possible, and so in this

section we present two degree 3 spanners for the infinite square lattice that attain this bound. Another possible candidate is presented in Section 1.5.

Theorem 2.1. *Let Λ be the infinite square lattice. Then,*

$$2.4142\ldots = 1+\sqrt{2} \leq \delta_0(\Lambda, 3) \leq (2\sqrt{2}+3)\,5^{-1/2} = 2.6065\ldots$$

Proof. To prove the lower bound, consider any point $p_0 \in \Lambda$ and its eight neighbors $p_1, \ldots, p_8$, as in Fig. 2.1. Since $\texttt{deg}(p_0) \leq 3$, p_0 can be connected to at most three neighbors from

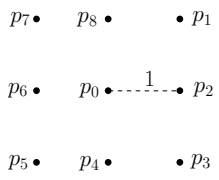

Figure 2.1: Illustrating the lower bound of $1+\sqrt{2}$ for the square lattice.

$\{p_2, p_4, p_6, p_8\}$. We may assume that the edge p_0p_2 is not present; then

$$\delta(p_0, p_2) \geq \frac{|\varrho(p_0, p_i, p_2)|}{|p_0p_2|} \geq 1+\sqrt{2}, \text{ where } i \in \{1, 3, 4, 8\}.$$

To prove the upper bound, we construct a plane degree 3 geometric graph G as illustrated in Fig. 2.2 (left); observe that there are four types of vertices in G. For any two lattice points $p, q \in \Lambda$, we construct a path in G. Set $p = (0, 0)$ as the origin and consider the four quadrants W_i, $i = 1, \ldots, 4$, labeled counterclockwise in the standard fashion; see Fig. 2.2 (right). Points on the dividing lines are assigned arbitrarily to any of the two adjacent quadrants. By the symmetry of G, we can assume that q lies in the first quadrant, thus $q = (x, y)$, where $x, y \geq 0$, while the origin $p = (0, 0)$ can be at any of the four possible types of lattice points.

Consider the path from $p = (0, 0)$ to $q = (x, y)$ via (z, z), where $z = \min(x, y)$, that visits every other lattice point on this diagonal segment as shown in Fig. 2.3, and let $\ell(x, y)$ denote its length. If $x = 0$, the stretch factor is easily seen to be at most $1+\sqrt{2}$. Since a horizontal path connecting two points with the same y-coordinate at distance a is always shorter than any path

connecting two points with the same x-coordinate at the same distance a, it is enough to prove our bound on the stretch factor in the case $y \geq x$ (i.e., $z = x$). We thus subsequently assume that $y \geq x \geq 1$.

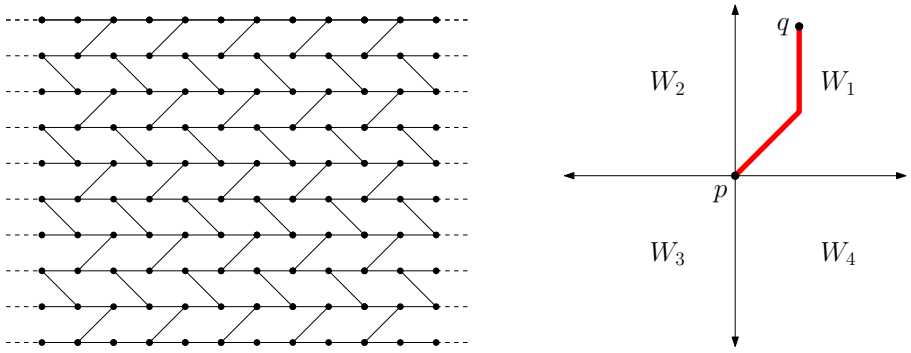

Figure 2.2: Left: a degree 3 plane graph on Λ. Right: a schematic diagram showing the path between p, q (when $x \leq y$). The bold path consist of segments of lengths 1 and $\sqrt{2}$.

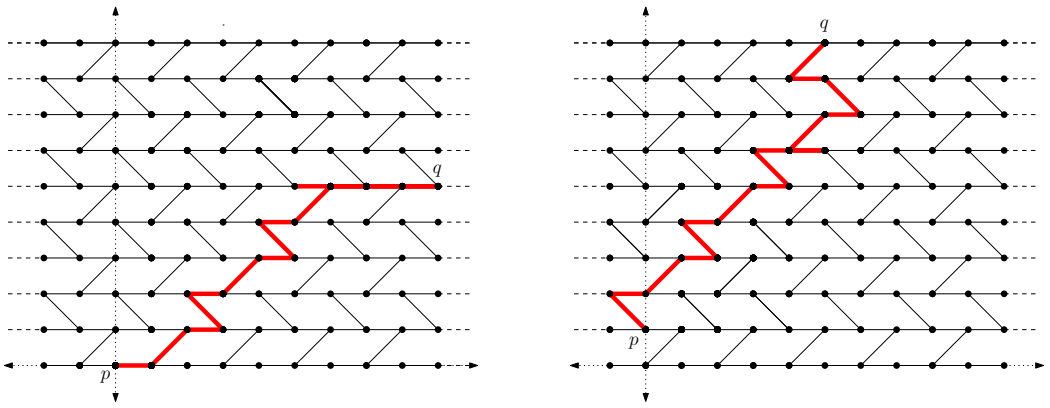

Figure 2.3: Paths connecting p to q in G generated by the procedure outlined in the text. Observe that in both examples a unit horizontal edge is traversed in both directions (but can be shortcut).

Observe that connecting points (a, a) with $(a + 2, a + 2)$, for any $a \geq 0$, requires length $2 + 2\sqrt{2}$, and that connecting points (a, a) with $(a + 1, a + 1)$, for any $a \geq 0$, requires length at most $2 + \sqrt{2}$. It follows that

$$\begin{aligned}\ell(x, y) &\leq \left(2\left\lceil \frac{x}{2} \right\rceil + \sqrt{2}x\right) + (y - x)(1 + \sqrt{2}) \\ &\leq 2\left(\frac{x+1}{2}\right) + \sqrt{2}x + (y - x)(1 + \sqrt{2}) = 1 + y(1 + \sqrt{2}).\end{aligned}$$

Since $|pq| = \sqrt{x^2+y^2}$, the corresponding stretch factor is bounded in terms of x, y as follows

$$\delta(p,q) \leq \gamma(x,y) := \frac{1+y(1+\sqrt{2})}{\sqrt{x^2+y^2}}. \tag{2.1}$$

We now consider the case $x = 1$ separately. Let $\lambda = \frac{1}{y}$, where $y = 1, 2, 3, 4, 5, \ldots$, and so $\lambda = 1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \ldots \in (0,1)$. According to (2.1) we have

$$\gamma(1,y) \leq \frac{1+y(1+\sqrt{2})}{\sqrt{y^2+1}} = \frac{\lambda+1+\sqrt{2}}{\sqrt{\lambda^2+1}} =: f(\lambda).$$

The derivative f' vanishes at $\lambda_0 = \frac{1}{\sqrt{2}+1} = \sqrt{2}-1 = 0.4142\ldots$ On the interval $(0,1)$: f is increasing on the interval $(0,\lambda_0)$ and decreasing on the interval $(\lambda_0,1)$; it attains a unique maximum at $\lambda = \lambda_0$. Since $\lambda_0 \in (\frac{1}{3}, \frac{1}{2})$, we have

$$f(\lambda) \leq \max\left(f\left(\frac{1}{3}\right), f\left(\frac{1}{2}\right)\right) = f\left(\frac{1}{3}\right) = f\left(\frac{1}{2}\right) = \delta_0.$$

It remains to consider the case $x \geq 2$; according to (2.1) we have

$$\begin{aligned}\delta(p,q) &\leq \frac{1+y(1+\sqrt{2})}{\sqrt{x^2+y^2}} \leq \frac{1+y(1+\sqrt{2})}{\sqrt{4+y^2}} \\ &= \frac{(1+\sqrt{2})(y/2)+1/2}{\sqrt{(y/2)^2+1}} \leq \sqrt{(1+\sqrt{2})^2+1/4} = 2.4654\ldots < \delta_0,\end{aligned}$$

where the last inequality follows from Fact 2.1 by setting $\lambda = y/2$.

This completes the case analysis. Observe that the above analysis is tight since there are point pairs with $x = 1, y = 2$ having pairwise stretch factor δ_0. We have thus shown that for any $p, q \in \Lambda$, we have $\delta(p,q) \leq (3+2\sqrt{2})\,5^{-1/2}$, completing the proof of the upper bound, and thereby the proof of Theorem 2.1. □

Another degree 3 spanner with stretch factor $\delta_0 = (3+2\sqrt{2})\,5^{-1/2}$. The graph G is illustrated in Fig. 2.4 (left). For any two lattice points $p, q \in \Lambda$, we construct a path in G. Set

$p = (0,0)$ as the origin and consider the four quadrants W_i, $i = 1, \ldots, 4$, labeled counter-clockwise in the standard fashion; see Fig. 2.4 (right). Points on the dividing lines are assigned arbitrarily to any of the two adjacent quadrants. By the symmetry of G, we can assume that q lies in one of the first two quadrants.

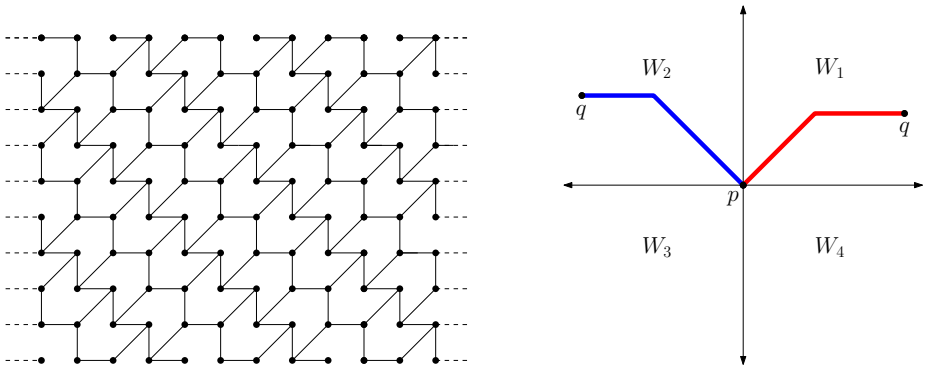

Figure 2.4: Left: a degree 3 spanner on Λ. Right: a schematic diagram showing the path between p, q when q lies in different quadrants of p (when $y \leq x$). The bold paths consist of segments of lengths 1 and $\sqrt{2}$.

Case 1: $q \in W_1$. By the symmetry of G, we may assume in the analysis that $q = (x, y)$, where $0 \leq y \leq x$. Consider the path from p to q via (y, y), that visits every lattice point on this diagonal segment as shown in Fig. 2.5 and let $\ell(x, y)$ denote its length.

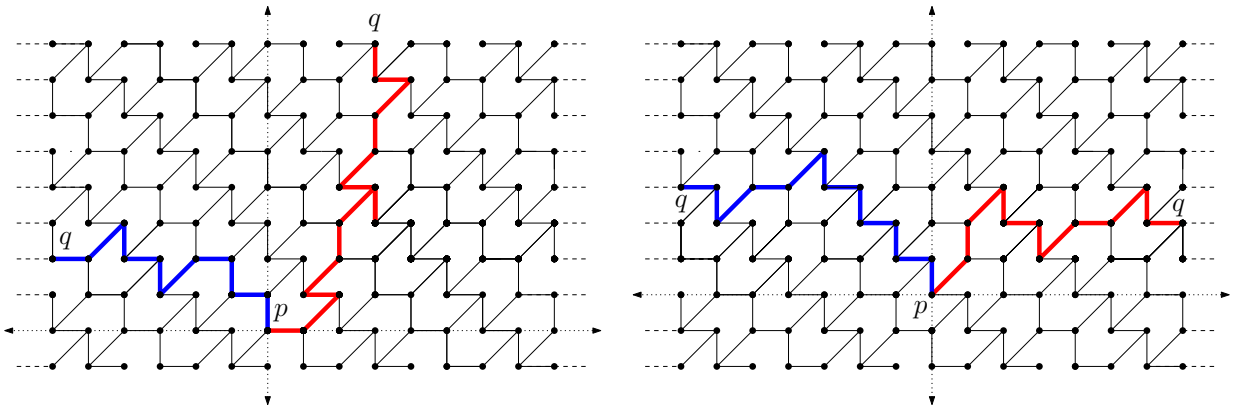

Figure 2.5: Illustration of various paths from p to q depending on the pattern of edges incident to p; for $q \in W_1$ (in red) and for $q \in W_2$ (in blue). Observe that in the red path on the left, a unit vertical edge is traversed in both directions (but can be shortcut).

If $y = 0$, or $x = y$, it is easily checked that the stretch factor is at most $1 + \sqrt{2}$. Assume

subsequently that $x \geq y+1$ and $y \geq 1$. A path of length $(2+2\sqrt{2})\lfloor y/2 \rfloor + (y \bmod 1)(2+\sqrt{2})$ suffices to reach from $p=(0,0)$ to (y,y), and a path of length $\lceil (x-y)/2 \rceil \sqrt{2} + (x-y)$ suffices to reach from (y,y) to $q=(x,y)$. Thus,

$$\ell(x,y) \leq (2+2\sqrt{2})\left\lfloor \frac{y}{2} \right\rfloor + (y \bmod 1)(2+\sqrt{2}) + \left\lceil \frac{x-y}{2} \right\rceil \sqrt{2} + (x-y).$$

That is,

$$\ell(x,y) \leq \begin{cases} (2+2\sqrt{2})\dfrac{y}{2} + (x-y) + \left\lceil \dfrac{x-y}{2} \right\rceil \sqrt{2}, & \text{for even } y \\ (2+2\sqrt{2})\dfrac{y-1}{2} + (2+\sqrt{2}) + (x-y) + \left\lceil \dfrac{x-y}{2} \right\rceil \sqrt{2}, & \text{for odd } y. \end{cases}$$

The distance $|pq|$ equals $\sqrt{x^2+y^2}$ in either case, and so the corresponding stretch factor (bounded in terms of x, y) is

$$\delta(p,q) \leq \gamma(x,y) := \begin{cases} \dfrac{\left(1+\frac{\sqrt{2}}{2}\right)x + \frac{\sqrt{2}}{2}y + \frac{\sqrt{2}}{2}}{\sqrt{x^2+y^2}}, & \text{for even } y \qquad (2.2) \\ \dfrac{\left(1+\frac{\sqrt{2}}{2}\right)x + \frac{\sqrt{2}}{2}y + \left(1+\frac{\sqrt{2}}{2}\right)}{\sqrt{x^2+y^2}}, & \text{for odd } y. \qquad (2.3) \end{cases}$$

Consider first the case of even y. Since the case $y=0$ has been dealt with, we have $y \geq 2$. Setting $\lambda = x/y$ in (2.2) and using Fact 2.1 in the last step yields

$$\begin{aligned} \delta(p,q) \leq \gamma(x,y) = \frac{\left(1+\frac{\sqrt{2}}{2}\right)\lambda + \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2y}}{\sqrt{\lambda^2+1}} &\leq \frac{\left(1+\frac{\sqrt{2}}{2}\right)\lambda + \frac{3\sqrt{2}}{4}}{\sqrt{\lambda^2+1}} \\ &\leq \sqrt{\left(1+\frac{\sqrt{2}}{2}\right)^2 + \frac{9}{8}} < 2.01 < \delta_0. \end{aligned}$$

Consider now the case of odd y. We have $y \geq 1$ and $x \geq y+1 \geq 2$. By (2.3) we have

$$\gamma(x,1) = \frac{\left(1+\frac{\sqrt{2}}{2}\right)x + (1+\sqrt{2})}{\sqrt{x^2+1}} := f(x).$$

We next show that f is decreasing on the interval $[2,\infty)$. Indeed, $f'(x) = f_1(x)/(x^2+1)^{3/2}$,

where

$$f_1(x) = \left(1 + \frac{\sqrt{2}}{2}\right)(x^2+1) - x\left[\left(1 + \frac{\sqrt{2}}{2}\right)x + (1+\sqrt{2})\right]$$
$$= \left(1 + \frac{\sqrt{2}}{2}\right) - (1+\sqrt{2})x < 0, \text{ for } x \geq 2.$$

Consequently,

$$\delta(p,q) \leq \gamma(x,1) = f(x) \leq f(2) = \delta_0,$$

as required. Observe that the above analysis is tight for some point pairs with $x = 2, y = 1$ (that achieve stretch factor δ_0).

Consider now the remaining case $y \geq 3$. Setting $\lambda = x/y$ in (2.3) and using Fact 2.1 in the last step yields

$$\delta(p,q) \leq \gamma(x,y) \leq \frac{\left(1 + \frac{\sqrt{2}}{2}\right)\lambda + \frac{\sqrt{2}}{2} + \left(1 + \frac{\sqrt{2}}{2}\right)\frac{1}{3}}{\sqrt{\lambda^2+1}} = \frac{\left(1 + \frac{\sqrt{2}}{2}\right)\lambda + \frac{1+2\sqrt{2}}{3}}{\sqrt{\lambda^2+1}}$$
$$\leq \sqrt{\left(1 + \frac{\sqrt{2}}{2}\right)^2 + \left(\frac{1+2\sqrt{2}}{3}\right)^2} < 2.14 < \delta_0,$$

as required.

Case 2: $q \in W_2$. We may assume that $q = (-x, y)$, where $x \geq y \geq 0$. Consider the path from p to q via $(-y, y)$, that visits every lattice point on this diagonal segment as shown in Fig. 2.5, and let $\ell(x,y)$ denote its length. The distance $|pq|$ equals $\sqrt{x^2+y^2}$.

If $y = 0$, it is easily checked that the stretch factor is at most $\sqrt{2}$, and so we assume subsequently that $y \geq 1$. The path length $\ell(x,y)$ is bounded from above as

$$\ell(x,y) \leq 2y + (x-y) + \left\lceil \frac{x-y}{2} \right\rceil \sqrt{2} \leq 2y + (x-y) + \frac{x-y+1}{2}\sqrt{2}$$
$$= \left(1 + \frac{\sqrt{2}}{2}\right)x + \left(1 - \frac{\sqrt{2}}{2}\right)y + \frac{\sqrt{2}}{2} \leq \left(1 + \frac{\sqrt{2}}{2}\right)x + y.$$

Setting $\lambda = x/y$ and using Fact 2.1 in the last step yields that the stretch factor is bounded as

$$\delta(p,q) \leq \gamma(x,y) := \frac{\left(1+\frac{\sqrt{2}}{2}\right)x+y}{\sqrt{x^2+y^2}} = \frac{\left(1+\frac{\sqrt{2}}{2}\right)\lambda+1}{\sqrt{\lambda^2+1}} \leq \sqrt{\left(1+\frac{\sqrt{2}}{2}\right)^2+1} < \sqrt{4} = 2 < \delta_0,$$

as required.

Next, we determine the degree 4 dilation of the square lattice.

Theorem 2.2. *Let Λ be the infinite square lattice. Then $\delta_0(\Lambda, 4) = \sqrt{2}$.*

Proof. Trivially, the (unrestricted degree) dilation of four points placed at the four corners of a square is $\sqrt{2}$. Thus, $\delta_0(\Lambda) \geq \sqrt{2}$. To prove the upper bound, construct a 4-regular graph G on Λ by connecting every $(i,j) \in \Lambda$ with its four neighbors $(i+1,j), (i,j+1), (i-1,j), (i,j-1)$. For any two points $p, q \in \Lambda$, the Manhattan path connecting them yields a stretch factor of the form

$$\frac{x+y}{\sqrt{x^2+y^2}} \leq \sqrt{2}, \text{ where } x, y \in \mathbf{N},$$

as required. □

Remark. It can be checked that the upper and lower bounds in Theorem 2.2 hold for every degree $k \geq 4$. Thus, $\delta_0(\Lambda, k) = \sqrt{2}$ for $k \geq 4$.

2.4 The hexagonal lattice

This section is devoted to the degree 3 and 4 dilation of the hexagonal lattice. In [16], we showed that the degree 3 dilation of the infinite hexagonal lattice is between 2 and 3. Also see the Appendix for the degree 3 geometric spanner having stretch factor 3. Here we establish that the exact value is $1+\sqrt{3}$.

Theorem 2.3. *Let Λ be the infinite hexagonal lattice. Then $\delta_0(\Lambda, 3) = 1+\sqrt{3}$.*

Proof. **Lower bound.** Consider a section of the lattice as shown in Fig. 2.6 (left). First, we

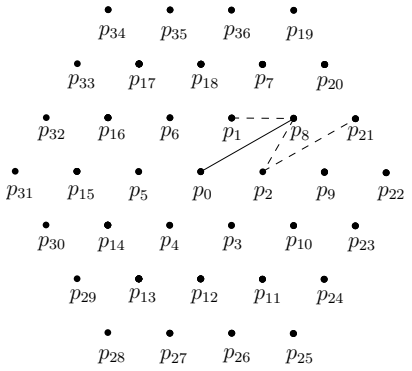

Figure 2.6: If an edge of length $\sqrt{3}$ is present, then the stretch factor of any plane degree 3 graph is $\geq 1 + \sqrt{3}$.

will show that if an edge of length at least $\sqrt{3}$ is present, the stretch factor of any resulting plane degree 3 graph is at least $1 + \sqrt{3}$. Now assume, as we may, that the edge p_0p_8 of length $\sqrt{3}$ is present. Now consider the point pair p_1, p_2. Clearly, $|p_1p_2| = 1$. It is easy to check that between p_1, p_2, there are two shortest detours each of length 2, viz. $\varrho(p_1, p_8, p_2)$ and $\varrho(p_1, p_0, p_2)$. The next largest detours $\varrho(p_1, p_8, p_9, p_2)$ and $\varrho(p_1, p_0, p_3, p_2)$ have length 3 each, in which cases, $\delta(p_1, p_2) \geq 3$. Hence, without loss of any generality, consider $\varrho(p_1, p_8, p_2)$, and assume that the edges p_1p_8 and p_2p_8 are present. Then,

$$\delta(p_8, p_{21}) \geq \frac{|\varrho(p_8, p_2, p_{21})|}{|p_8p_{21}|} \geq 1 + \sqrt{3}.$$

A similar argument can be made for any edge e of length greater than $\sqrt{3}$, since one can always locate two lattice points lying in opposite sides of e; as required in the above analysis. In the remaining part of the proof, assume that no edge of length $\sqrt{3}$ or more is present. In particular, we will only consider unit length edges in our proof. Note that if every point in Λ has degree 1 in the graph, we have a matching on Λ, and hence the graph is disconnected. Thus, let p_0 be any point in Λ with degree at least 2. We have the following two[1] cases:

Case 1: $\mathtt{deg}(p_0) = 2$. There are 3 non-symmetric sub-cases as follows.

[1]As mentioned in Section 2.1, one can argue that degree 3 is needed for achieving a constant stretch factor. Thus, it is enough to analyze the case when $\mathtt{deg}(p_0) = 3$. Nevertheless, we include a complete argument.

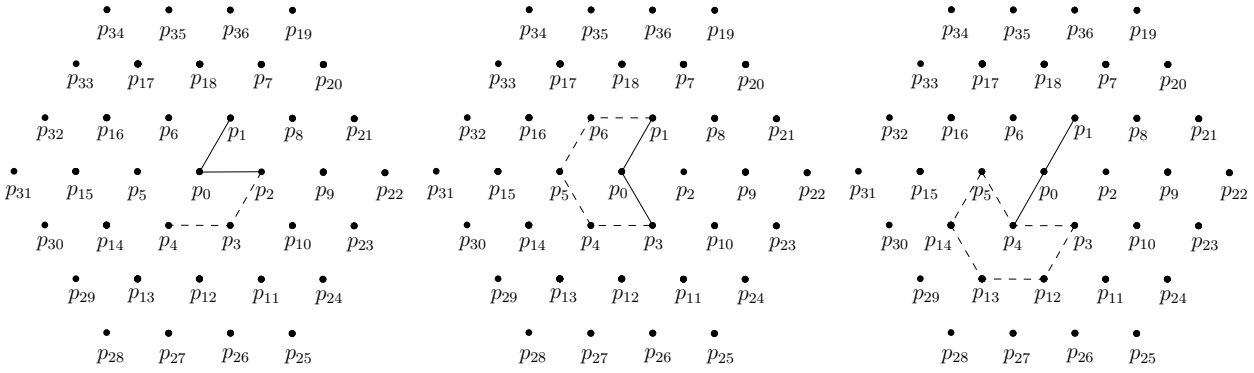

Figure 2.7: Illustration of *Case 1* from the proof of lower bound in Theorem 2.3. Left: *Case 1.1*, Middle: *Case 1.2*, Right: *Case 1.3*.

Case 1.1: Refer to Fig. 2.7 (left). Let the edges p_0p_1, p_0p_2 be present. Then,

$$\delta(p_0, p_4) \geq \frac{|\varrho(p_0, p_2, p_3, p_4)|}{|p_0p_4|} \geq 3.$$

Case 1.2: Refer to Fig. 2.7 (middle). Now, let the edges p_0p_1, p_0p_3 be present. Then,

$$\delta(p_0, p_5) \geq \frac{|\varrho(p_0, p_3, p_4, p_5)|}{|p_0p_5|} = \frac{|\varrho(p_0, p_1, p_6, p_5)|}{|p_0p_5|} \geq 3.$$

Case 1.3: Refer to Fig. 2.7 (right). Let the edges p_0p_1, p_0p_4 be present. Note that if the edge p_3p_4 is absent, $\delta(p_0, p_3) \geq 3$. So, assume that p_3p_4 is present. Similarly let p_4p_5 be present otherwise $\delta(p_0, p_5) \geq 3$. Then, arguing the same way as in *Case 1.2*, $\delta(p_4, p_{13}) \geq 3$.

Case 2: $\deg(p_0) = 3$. There are 3 non-symmetric sub-cases as follows.

Case 2.1: Refer to Fig. 2.8 (left). Let the edges p_0p_1, p_0p_2, p_0p_3 be present. Then, by a similar argument as in *Case 1.2*, $\delta(p_0, p_5) \geq 3$.

Case 2.2: Refer to Fig. 2.8 (middle). Now, let the edges p_0p_3, p_0p_4, p_0p_6 be present. Clearly, if p_1p_6 is absent, $\delta(p_0, p_1) \geq 3$. Thus, assume that p_1p_6 is present. Now consider the pair p_5, p_6. If p_5p_6 is present, then $\delta(p_6, p_{17}) \geq 3$, arguing in a similar way to *Case 1.2*. Thus, assume that p_5p_6 is absent. The shortest detour between p_5, p_6 is $\varrho(p_5, p_{16}, p_6)$ which has length 2. The next largest detour has length 3. So, let the edges p_6p_{16} and p_5p_{16} be present. Now consider

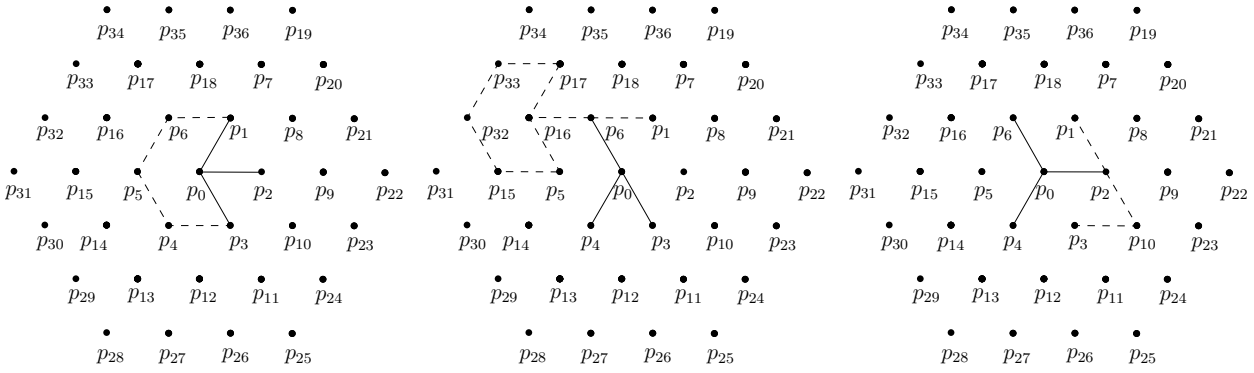

Figure 2.8: Illustration of *Case 2* from the proof of lower bound in Theorem 2.3. Left: *Case 2.1*, Middle: *Case 2.2*, Right: *Case 2.3*.

the pair p_{16}, p_{17}. If $p_{16}p_{17}$ is present, $\delta(p_{16}, p_{32}) \geq 3$ (analysis is similar to *Case 1.2*), otherwise, $\delta(p_6, p_{17}) \geq 3$.

Case 2.3: Refer to Fig. 2.8 (right). Let p_0p_2, p_0p_4, p_0p_6 be present. To achieve $\delta(p_0, p_1) = 2$, at least one of p_1p_2 or p_1p_6 needs to be present (the next largest detour has length 3). Without loss of any generality, assume that p_1p_2 is present. Now consider the pair p_2, p_3. If p_2p_3 is present, then by *Case 2.1*, $\delta(p_2, p_9) \geq 3$. So, assume that p_2p_3 is absent. The minimum length detour is $\varrho(p_2, p_{10}, p_3)$ (next largest detours have length 3 each). Thus, let p_2p_{10} and p_3p_{10} be present. Now, observe that the edges incident to p_2 form the same symmetric pattern as dealt with in *Case 2.2*, where it is shown that the stretch factor of any resulting degree 3 plane graph is at least 3.

Upper bound. We construct a 3-regular graph G achieving $\delta_0(\Lambda, 3) \leq 1 + \sqrt{3}$, as illustrated in Fig. 2.9 (left). For any two lattice points $p, q \in \Lambda$, we construct a path in G. Set p as the origin, and subdivide the plane into six wedges of $60°$ each, centered at p, and labeled counterclockwise W_i, $i = 1, \ldots, 6$, as in Fig. 2.9 (right). Points on the dividing lines are assigned arbitrarily to any of the two adjacent wedges. Let $\theta = \pi/3$, and consider the three unit vectors $\vec{\mu_i} = (\cos i\theta, \sin i\theta)$, for $i = 0, 1, 2$. We distinguish three cases depending on the location of q.

Case 1: $q \in W_1$ (the case $q \in W_4$ is symmetric), i.e., $\vec{q} = u\vec{\mu_0} + v\vec{\mu_1}$, for some $u, v \in \mathbf{N}$. By the symmetry of G, we can assume that $u \geq v \geq 0$ in the analysis. Consider the path from p

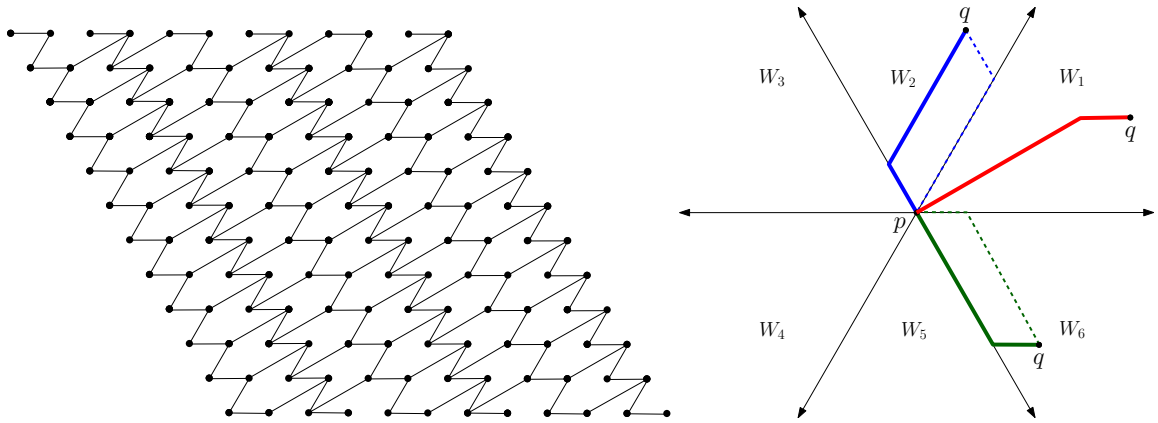

Figure 2.9: Left: a degree 3 plane graph on Λ. Right: a schematic diagram showing the path between p and q when q lies in different wedges determined by p. The bold paths consist of segments of lengths 1 and $\sqrt{3}$. Alternative paths are shown using dotted segments.

to q via $v\vec{\mu_0} + v\vec{\mu_1}$ that visits every lattice point on the diagonal segment, as shown in Fig. 2.10 and let $\ell(u, v)$ denote its length. Observe that connecting $a\vec{\mu_0} + a\vec{\mu_1}$ to $(a+2)\vec{\mu_0} + (a+2)\vec{\mu_1}$ requires a length of $2 + 2\sqrt{3}$. Thus,

$$\ell(u, v) \leq (2 + 2\sqrt{3})\left\lfloor \frac{v}{2} \right\rfloor + (2 + \sqrt{3})(v \bmod 1) + (u - v) + \left\lceil \frac{u - v}{2} \right\rceil \sqrt{3}.$$

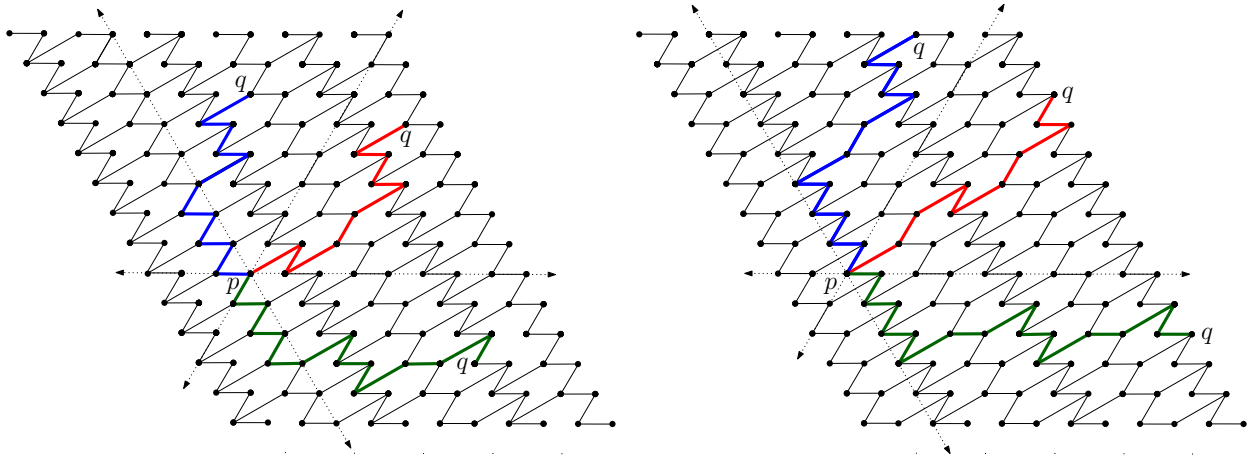

Figure 2.10: Illustration of various paths from p to q depending on the pattern of edges incident to p; for $q \in W_1$ (in red), for $q \in W_2$ (in blue), and for $q \in W_6$ (in green).

For even v, $\ell(u,v)$ is bounded from above by

$$\ell(u,v) \le \left(2+2\sqrt{3}\right)\frac{v}{2} + \left(1+\frac{\sqrt{3}}{2}\right)(u-v) + \frac{\sqrt{3}}{2} = \left(1+\frac{\sqrt{3}}{2}\right)u + \frac{\sqrt{3}}{2}v + \frac{\sqrt{3}}{2}.$$

For odd v, $\ell(u,v)$ is bounded from above by

$$\begin{aligned}\ell(u,v) &\le \left(2+2\sqrt{3}\right)\frac{v-1}{2} + (2+\sqrt{3}) + \left(1+\frac{\sqrt{3}}{2}\right)(u-v) + \frac{\sqrt{3}}{2} \\ &= \left(1+\frac{\sqrt{3}}{2}\right)u + \frac{\sqrt{3}}{2}v + \left(1+\frac{\sqrt{3}}{2}\right).\end{aligned}$$

The distance $|pq|$ equals

$$\sqrt{u^2+v^2-2uv\cos\frac{2\pi}{3}} = \sqrt{u^2+v^2+uv},$$

and so the corresponding stretch factor $\delta(p,q)$ is bounded by a function $\gamma(u,v)$ as follows

$$\delta(p,q) \le \gamma(u,v) := \begin{cases} \dfrac{\left(1+\frac{\sqrt{3}}{2}\right)u + \frac{\sqrt{3}}{2}v + \frac{\sqrt{3}}{2}}{\sqrt{u^2+v^2+uv}}, & \text{for even } v \\ \dfrac{\left(1+\frac{\sqrt{3}}{2}\right)u + \frac{\sqrt{3}}{2}v + \left(1+\frac{\sqrt{3}}{2}\right)}{\sqrt{u^2+v^2+uv}}, & \text{for odd } v. \end{cases}$$

Consider first the case of even v. We have $u \ge v \ge 0$ and $u \ge 1$ (since $u = v = 0$ is not a valid choice). We next show that $\gamma(u,v)$ is a decreasing function of v for $v \ge 0$. Indeed,

$$\frac{\partial\gamma(u,v)}{\partial v} = f(u,v)/[2(u^2+v^2+uv)^{3/2}],$$

where

$$\begin{aligned}f(u,v) &= \sqrt{3}\left(u^2+v^2+uv\right) - (2v+u)\left[\left(1+\frac{\sqrt{3}}{2}\right)u + \frac{\sqrt{3}}{2}v + \frac{\sqrt{3}}{2}\right] \\ &= \left(\frac{\sqrt{3}}{2}-1\right)u^2 - \left(2+\frac{\sqrt{3}}{2}\right)uv - \frac{\sqrt{3}}{2}u - \sqrt{3}v < 0.\end{aligned}$$

Consequently,

$$\delta(p,q) \le \gamma(u,v) \le \gamma(u,0) = \frac{\left(1+\frac{\sqrt{3}}{2}\right)u + \frac{\sqrt{3}}{2}}{u} = 1 + \frac{\sqrt{3}}{2} + \frac{\sqrt{3}}{2u} \le 1+\sqrt{3},$$

as required.

Consider now the case of odd v. We have $u \ge v \ge 1$. Since the expressions of $\gamma(u,v)$ for odd and even v differ by $(u^2+v^2+uv)^{-1/2}$, which is also a decreasing function of v, it follows that $\gamma(u,v)$ for odd v is decreasing on the same interval, in particular on the interval $v \ge 1$. Consequently,

$$\delta(p,q) \le \gamma(u,v) \le \gamma(u,1) = \frac{\left(1+\frac{\sqrt{3}}{2}\right)u + (1+\sqrt{3})}{\sqrt{u^2+u+1}} \le \frac{3}{2} + \frac{2}{\sqrt{3}} < 1+\sqrt{3},$$

as required. To check this last inequality, let

$$h(u) = \frac{\left(1+\frac{\sqrt{3}}{2}\right)u + (1+\sqrt{3})}{\sqrt{u^2+u+1}},$$

and notice that this function is decreasing for $u \ge 1$, thus $h(u) \le h(1) = \frac{3}{2} + \frac{2}{\sqrt{3}}$.

Case 2: $q \in W_2$ (the case $q \in W_5$ is symmetric), i.e., $\vec{q} = u\vec{\mu_2} + v\vec{\mu_1}$, for some $u, v \in \mathbf{N}$. Consider the path from p to q via $u\vec{\mu_2}$ as shown in Fig. 2.10 and let $\ell(u,v)$ denote its length. (Alternatively, the path via $v\vec{\mu_1}$ can be used.) Then $\ell(u,v)$ is bounded from above as follows

$$\ell(u,v) \le 2u + v + \left\lceil \frac{v}{2} \right\rceil \sqrt{3} \le 2u + \left(1+\frac{\sqrt{3}}{2}\right)v + \frac{\sqrt{3}}{2}.$$

As in Case 1, the distance $|pq|$ equals $\sqrt{u^2+v^2-2uv\cos\frac{2\pi}{3}} = \sqrt{u^2+v^2+uv}$, and so the corresponding stretch factor is

$$\delta(p,q) \le \gamma(u,v) := \frac{2u + \left(1+\frac{\sqrt{3}}{2}\right)v + \frac{\sqrt{3}}{2}}{\sqrt{u^2+v^2+uv}}.$$

We can assume that $u \geq 1$ and $v \geq 1$ (else the stretch factor is at most $1+\sqrt{3}$). Further, since the coefficient of u is larger than that of v in the numerator, we can assume that $u \geq v \geq 1$ when maximizing $\gamma(u,v)$. Set now $\lambda = \frac{u}{v} \geq 1$. We have

$$\gamma(u,v) = \frac{2\lambda + \left(1+\frac{\sqrt{3}}{2}\right) + \frac{\sqrt{3}}{2v}}{\sqrt{\lambda^2+\lambda+1}} \leq \frac{2\lambda + (1+\sqrt{3})}{\sqrt{\lambda^2+\lambda+1}} := f(\lambda).$$

It is easy to check that $f(\lambda)$ is decreasing for $\lambda \geq 1$, hence for $u, v \geq 1$ we also have

$$\delta(p,q) \leq \gamma(u,v) \leq f(1) = 1+\sqrt{3},$$

as required.

Case 3: $q \in W_6$ (the case $q \in W_3$ is symmetric), i.e., $\vec{q} = -u\vec{\mu_2} + v\vec{\mu_0}$, for some $u, v \in \mathbf{N}$. By the symmetry of G, this case is symmetric to Case 2.

This completes the case analysis and thereby the proof of the upper bound. □

Remark. In [15] we have shown that a certain 13-point section of the hexagonal lattice with six boundary points removed has degree 3 dilation at least $1+\sqrt{3}$. It is worth noting that this subset cannot be used however to deduce that the degree 3 dilation of the hexagonal lattice is at least $1+\sqrt{3}$. Indeed, the reason is that the absence of the respective boundary points has been explicitly invoked in that argument. This is the reason of why in the proof of the lower bound in Theorem 2.3 we have used a different argument.

Next we determine the degree 4 dilation of the infinite hexagonal lattice.

Theorem 2.4. *Let Λ be the infinite hexagonal lattice. Then $\delta_0(\Lambda, 4) = 2$.*

Proof. We first prove the lower bound. Let p_0 be any point in Λ with its six closest neighbors, say, $p_1, \ldots, p_6$, where $|p_0p_i| = 1$, for $i = 1, \ldots, 6$. Since $\deg(p_0) \leq 4$ in any plane degree 4 geometric spanner on Λ, there exists $i \in \{1, \ldots, 6\}$ such that the edge p_0p_i is absent; we may

assume that $i = 1$. Then

$$\delta(p_0, p_1) \geq \frac{|\varrho(p_0, p_i, p_1)|}{|p_0 p_1|} \geq 2, \text{ where } i \in \{2, 6\}.$$

To prove the upper bound, consider the 4-regular graph G shown in Fig. 2.11; it remains to show that $\delta(G) \leq 2$. For any two lattice points $p, q \in \Lambda$, we construct a path in G. Consider the setup from the proof of Theorem 2.3. Set the lower point p as the origin $(0, 0)$. Let $\theta = \pi/3$,

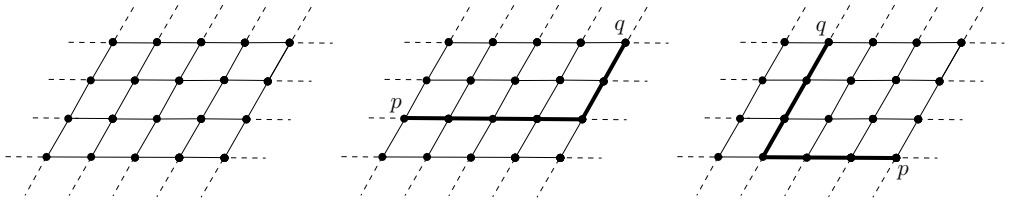

Figure 2.11: Left: a degree 4 plane graph G on Λ. Middle, Right: illustration of various paths from p to q depending on their relative position in Λ.

and consider the two unit vectors $\vec{\mu_i} = (\cos i\theta, \sin i\theta)$, for $i = 0, 1$. Then $\vec{q} = \pm u\vec{\mu_0} + v\vec{\mu_1}$, for some $u, v \in \mathbf{N}$. Since the two points can be connected by a path in G of length $u + v$, and the distance between the points is $\sqrt{u^2 + v^2 \pm uv}$ (depending on their relative position in Λ), the corresponding stretch factor satisfies

$$\delta(p, q) \leq \gamma(u, v) := \frac{u + v}{\sqrt{u^2 + v^2 \pm uv}} \leq 2, \tag{2.4}$$

as required. Indeed, the above inequalities are equivalent to $(u \pm v)^2 \geq 0$, which are obvious. □

Remarks. 1. Another degree 4 spanner for the hexagonal lattice with stretch factor 2 appears in Fig. 2.12; the proof for the stretch factor is left to the reader.

2. Note that $\delta_0(\Lambda, 5) = 2$ since the bounds in Theorem 2.4 also hold for degree 5. Let now $k = 6$. Connecting each lattice point with all its six closest neighbors yields a planar graph with stretch factor $2/\sqrt{3}$. Indeed, as in (2.4), the stretch factor satisfies

$$\delta(p, q) \leq \gamma(u, v) := \frac{u + v}{\sqrt{u^2 + v^2 + uv}} \leq \frac{2}{\sqrt{3}},$$

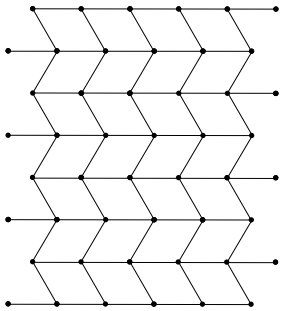

Figure 2.12: A degree 4 spanner on Λ with stretch factor 2.

where the above inequality is equivalent to $(u-v)^2 \geq 0$, which is obvious. Hence $\delta_0(\Lambda, 6) \leq 2/\sqrt{3}$. On the other hand, an argument similar to that in the proof of the inequality $\delta_0(\Lambda, 3) \geq 1+\sqrt{3}$ shows that the presence of any edge longer than 1 would force the stretch factor to be at least 2. We may thus assume that the spanner G contains all unit edges (since no two cross each other); now the length of a shortest path in G connecting any pair of lattice points at distance $\sqrt{3}$ is 2, thus the stretch factor $2/\sqrt{3}$ is also needed. Consequently, $\delta_0(\Lambda, 6) = 2/\sqrt{3}$. It can be easily checked that $\delta_0(\Lambda, k) = \delta_0(\Lambda, 6) = 2/\sqrt{3}$ for every $k \geq 6$.

2.5 Concluding remarks

We have given constructive upper bounds and derived close lower bounds on the degree 3 dilation of the infinite square lattice in the domain of plane geometric spanners. We have also derived exact values for the degree 4 dilation of the square lattice along with the degree 3 and 4 dilation of the infinite hexagonal lattice. It is easy to verify that our bounds also apply for finite sections of these lattices; see [16] or the Appendix for some examples.

It may be worth pointing out that in addition to the low stretch factors achieved, the spanners we construct also have low weight and low geometric dilation[2]; see for instance [14, 17] for basic terms. That is, each of these two parameters is at most a small constant factor times the optimal one attainable.

[2]When the stretch factor (or dilation) is measured over all pairs of points on edges or vertices of a plane graph G (rather than only over pairs of vertices) one arrives at the concept of *geometric dilation* of G.

As shown in Theorem 2.1, the degree 3 dilation of the infinite square lattice is at most $(3+2\sqrt{2})\,5^{-1/2}$. It would be interesting to know whether this upper bound can be improved, and so we put forward the following.

Conjecture 2.1. *Let Λ be the infinite square lattice. Then $\delta_0(\Lambda,3) = (3+2\sqrt{2})\,5^{-1/2} = 2.6065\ldots$*

A lighter degree 3 spanner. The graph G is illustrated in Fig. 2.13. It is easy to check that it is "shorter" than each of the two previous spanners of degree 3 for the square lattice analyzed in Section 2.3.

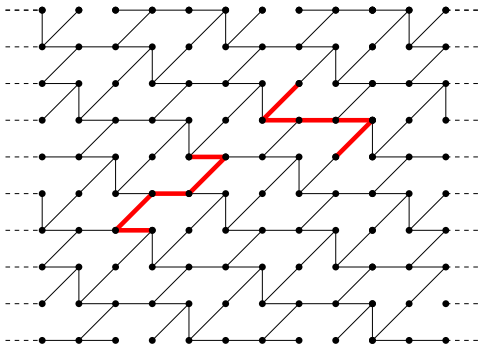

Figure 2.13: A lighter degree 3 spanner on the infinite square lattice. The shortest paths between point pairs with pairwise stretch factor δ_0 are shown in red.

Indeed, the average cost (length) per vertex is in this case smaller (note that some vertices have degree 2):

$$\frac{4}{5}\left(\frac{1}{2}+\frac{1}{2}+\frac{\sqrt{2}}{2}\right)+\frac{1}{5}\left(\frac{\sqrt{2}}{2}+\frac{\sqrt{2}}{2}\right)=\frac{4}{5}+\frac{3}{5}\sqrt{2}=1.6485\ldots,$$

while that for the previous spanners it is $\frac{1}{2}+\frac{1}{2}+\frac{\sqrt{2}}{2}=1.7071\ldots$ In particular, the total length of a square lattice section with n points is $1.6485\ldots n+o(n)$ rather than $1.7071\ldots n+o(n)$. If the stretch factor of G would also be $\delta_0=(3+2\sqrt{2})\,5^{-1/2}$, G would be superior from the length perspective to the two spanners described in Section 2.3. We conjecture that the stretch factor of this lighter degree 3 spanner shown in Fig. 2.13 equals δ_0.

References

[1] P. K. Agarwal, R. Klein, C. Kane, S. Langerman, P. Morin, M. Sharir, and M. Soss, Computing the detour and spanning ratio of paths, trees, and cycles in 2D and 3D, *Discrete Comput. Geom.* **39(1-3)** (2008), 17–37.

[2] I. Althöfer, G. Das, D. P. Dobkin, D. Joseph, and J. Soares, On sparse spanners of weighted graphs, *Discrete Comput. Geom.* **9** (1993), 81–100.

[3] B. Aronov, M. de Berg, O. Cheong, J. Gudmundsson, H. J. Haverkort, and A. Vigneron, Sparse geometric graphs with small dilation, *Comput. Geom.* **40(3)** (2008), 207–219.

[4] N. Bonichon, C. Gavoille, N. Hanusse, and L. Perković, Plane spanners of maximum degree six, In *Proc. Internat. Colloq. Automata, Lang. and Prog.*, Springer, 2010, pp. 19–30.

[5] N. Bonichon, I. Kanj, L. Perković, and G. Xia, There are plane spanners of degree 4 and moderate stretch factor, *Discrete Comput. Geom.* **53(3)** (2015), 514–546.

[6] P. Bose, P. Carmi, and L. Chaitman-Yerushalmi, On bounded degree plane strong geometric spanners, *J. Discrete Algorithms* **15** (2012), 16–31.

[7] P. Bose, J. Gudmundsson, and M. Smid, Constructing plane spanners of bounded degree and low weight, *Algorithmica* **42** (2005), 249–264.

[8] P. Bose and M. Smid, On plane geometric spanners: A survey and open problems, *Comput. Geom.* **46(7)** (2013), 818–830.

[9] P. Bose, M. Smid, and D. Xu, Delaunay and diamond triangulations contain spanners of bounded degree, *Internat. J. Comput. Geom. Appl.* **19(2)** (2009), 119–140.

[10] B. Chandra, G. Das, G. Narasimhan, and J. Soares, New sparseness results on graph spanners, *Internat. J. Comput. Geom. Appl.* **5** (1995), 125–144.

[11] O. Cheong, H. Herman, and M. Lee, Computing a minimum-dilation spanning tree is NP-hard, *Comput. Geom.* **41(3)** (2008), 188–205.

[12] P. Chew, There are planar graphs almost as good as the complete graph, *J. of Computer and System Sci.* **39(2)** (1989), 205–219.

[13] G. Das and P. Heffernan, Constructing degree-3 spanners with other sparseness properties, *Internat. J. Found. Comput. Sci.* **7(2)** (1996), 121–136.

[14] A. Dumitrescu, A. Ebbers-Baumann, A. Grüne, R. Klein, and G. Rote, On the geometric dilation of closed curves, graphs, and point sets, *Comput. Geom.* **36** (2006), 16–38.

[15] A. Dumitrescu and A. Ghosh, Lower bounds on the dilation of plane spanners, *Proc. Internat. Conf. Algor. and Discrete Appl. Math.*, 2016, vol. 9602 of LNCS, pp. 139-151; preprint, Oct. 2015, `arXiv:1509.07181.v2`.

[16] A. Dumitrescu and A. Ghosh, Lattice spanners of low degree, *Proc. Internat. Conf. Algor. and Discrete Appl. Math.*, 2016, vol. 9602 of LNCS, pp. 152-163; preprint, Feb. 2016, `arXiv:1602.04381.v1`.

[17] A. Ebbers-Baumann, A. Grüne, and R. Klein, On the geometric dilation of finite point sets, *Algorithmica* **44** (2006), 137–149.

[18] A. Ebbers-Baumann, R. Klein, E. Langetepe, and A. Lingas, A fast algorithm for approximating the detour of a polygonal chain, *Comput. Geom.* **27** (2004), 123–134.

[19] D. Eppstein, Spanning trees and spanners, in *Handbook of Computational Geometry* (J. R. Sack and J. Urrutia, editors), North-Holland, Amsterdam, 2000, pp. 425–461.

[20] J. Gudmundsson and C. Knauer, Dilation and detour in geometric networks, in *Handbook on Approximation Algorithms and Metaheuristics, Chap. 52* (T. Gonzalez, editor), Chapman & Hall/CRC, Boca Raton, 2007.

[21] I. Kanj and L. Perković, On geometric spanners of Euclidean and unit disk graphs, in *Proc. 25th Annual Sympos. on Theoretical Aspects of Comp. Sci.*, Schloss Dagstuhl Leibniz–Zentrum für Informatik, 2008, pp. 409–420.

[22] M. Keil and C. A. Gutwin, Classes of graphs which approximate the complete Euclidean graph, *Discrete Comput. Geom.* **7** (1992), 13–28.

[23] R. Klein, M. Kutz, and R. Penninger, Most finite point sets in the plane have dilation > 1, *Discrete Comput. Geom.* **53(1)** (2015), 80–106.

[24] T. Leighton, *Complexity Issues in VLSI, Foundations of Computing Series*, MIT Press, Cambridge, MA, 1983.

[25] C. Levcopoulos and A. Lingas, There are planar graphs almost as good as the complete graphs and almost as cheap as minimum spanning trees, *Algorithmica* **8** (1992), 251–256.

[26] X. Y. Li and Y. Wang, Efficient construction of low weight bounded degree planar spanner, *Internat. J. Comput. Geom. Appl.* **14(1–2)** (2004), 69–84.

[27] A. L. Liestman and T. C. Shermer, Grid spanners, *Networks* **23(2)** (1993), 123–133.

[28] A. L. Liestman, T. C. Shermer, and C. R. Stolte, Degree-constrained spanners for multidimensional grids, *Discrete Appl. Math.* **68(1)** (1996), 119–144.

[29] W. Mulzer, Minimum dilation triangulations for the regular n-gon, Masters Thesis, Freie Universität Berlin, 2004.

[30] G. Narasimhan and M. Smid, *Geometric Spanner Networks*, Cambridge University Press, 2007.

[31] M. Smid, Progress on open problems in the book Geometric Spanner Networks, `http://people.scs.carleton.ca/~michiel/SpannerBook/openproblems.html`.

[32] G. Xia, The stretch factor of the Delaunay triangulation is less than 1.998, *SIAM J. Comput.*, **42(4)** (2013), 1620–1659.

Appendix

In this Appendix, we present our earlier upper bounds on the dilation of lattices, both square and hexagonal.

Square lattice. First, we present a degree 3 spanner having stretch factor $\sqrt{4+2\sqrt{3}} = 2.6131\ldots$, and then we show how a small change in the network gives us a degree 3 spanner with a better stretch factor.

For all $(i,j) \in \mathbf{Z}^2$, connect (i,j) with $(i+1,j)$; For all $j \in \mathbf{Z}$ connect (i,j) with $(i+1,j+1)$ if i is odd, as shown in Fig. 2.14.

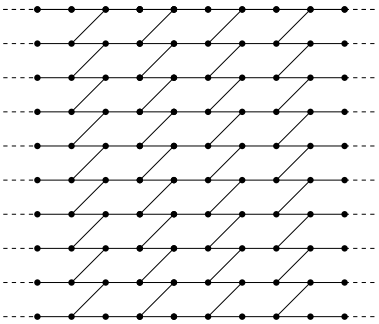

Figure 2.14: A degree 3 plane spanner for the square lattice with stretch factor at most $\sqrt{4+2\sqrt{2}} = 2.6131\ldots$

Let $u = (a,b)$ and $v = (c,d)$ be any two points in Λ. Clearly, if $b = d$, $\delta_G(u,v) = 1$. Now observe that in G we need to cover a distance of at most $\sqrt{2}+1$ to go from (i,j) to $(i,j+1)$ or vice-versa. Thus,

$$\delta_G(u,v) \leq \frac{|a-c| + (\sqrt{2}+1)|b-d|}{\sqrt{(a-c)^2+(b-d)^2}}, \tag{2.5}$$

since the length of the shortest path between u, v is at most the length of the path consisting of a straight-line horizontal sub-path of length $|a-c|$ and a zig-zag sub-path of length $(\sqrt{2}+1)|b-d|$. Now setting $|a-c| = x, |b-d| = y$ and using Fact 1, yields

$$\delta_G(u,v) \leq g(x,y) = \frac{x+(\sqrt{2}+1)y}{\sqrt{x^2+y^2}} \leq \sqrt{1+(\sqrt{2}+1)^2} = \sqrt{4+2\sqrt{2}} = 2.6131\ldots \tag{2.6}$$

To see that the same bound holds for any $m \times n$ section of the lattice, proceed as follows. Connect the points in $\Lambda(m, n)$ as shown in Fig. 2.15. The following connections are made:

STEP 1. For $j = 0$ to $n - 1$ and $i = 0$ to $m - 2$, connect (i, j) with $(i + 1, j)$.

STEP 2. For $j = 0$ to $n - 2$, connect $(0, j)$ with $(0, j + 1)$.

STEP 3. For $j = 0$ to $n - 2$, connect (i, j) with $(i + 1, j + 1)$ if i is odd and $(i + 1, j + 1) \in \Lambda(m, n)$.

STEP 4. If m is even, for $j = 0$ to $n - 2$, connect $(m - 1, j)$ with $(m - 1, j + 1)$.

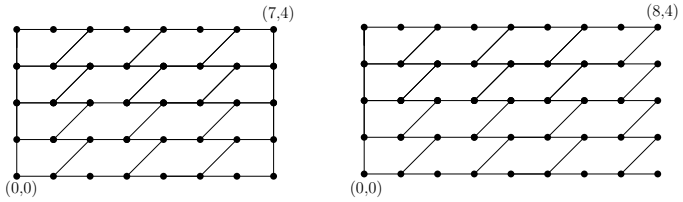

Figure 2.15: Connecting the points in $\Lambda(8, 5)$ (left) and $\Lambda(9, 5)$ (right).

Instances $\Lambda(8, 5)$ and $\Lambda(9, 5)$ are shown in Fig. 2.15. Now the analysis is exactly the same as above.

Interestingly enough, a twist in the spanner construction yields a slightly better stretch factor. The following connections are made in G: For all $(i, j) \in \mathbf{Z}^2$, connect (i, j) with $(i+1, j)$. For all $j \in \mathbf{Z}$ connect (i, j) with $(i+1, j+1)$ if $i \equiv 1 \pmod 4$, and connect (i, j) with $(i-1, j+1)$ if $i \equiv 0 \pmod 4$. See Fig. 2.16.

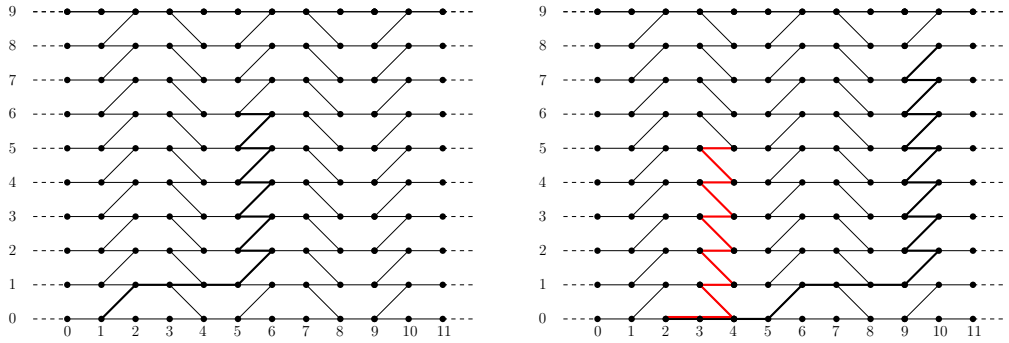

Figure 2.16: A degree 3 spanner with stretch factor $\frac{5\sqrt{2}+7}{\sqrt{29}} = 2.6129\ldots$ Left: a path connecting $(1, 0)$ with $(5, 6)$. Right: a path connecting $(2, 0)$ with $(4, 5)$ (in red) and a path connecting $(2, 0)$ with $(10, 8)$ (in black).

Observe that the upper bound in (2.6) still holds, since a path with the same structure,

namely a straight line horizontal sub-path of length $|a-c|$ followed by a zig-zag sub-path of length $(\sqrt{2}+1)|b-d|$ exists. We next derive a sharper bound: $\delta_0(\Lambda,3) \le (5\sqrt{2}+7)\,29^{-1/2} = 2.6129\ldots$

Assume, as we may, that $a \le c$ and $b \le d$. Put $x = c-a$ and $y = d-b$. We can further assume that $b = 0$, and so $u = (a,0)$.

If $x \ge y$, let $\lambda = x/y \ge 1$. We can write

$$\delta_G(u,v) \le \frac{(\sqrt{2}+1)y + x}{\sqrt{x^2+y^2}} = \frac{\sqrt{2}+1+\lambda}{\sqrt{1+\lambda^2}} =: f(\lambda).$$

Its derivative is $f'(\lambda) = \dfrac{1-(\sqrt{2}+1)\lambda}{\sqrt{1+\lambda^2}} < 0$, for $\lambda \ge 1$; f is a decreasing function on $[1,\infty)$, and thus $f(\lambda) \le f(1) = \sqrt{2}+1$ for this range of λ.

Let now $y \ge x+1$ for the remainder of the proof. If $x = 0$, it is easy to check that $\delta_G(u,v) \le \sqrt{2}+1$.

If $x = 1$, $\lambda = \frac{1}{y}$, where $y = 2,3,4,5,\ldots$, and so $\lambda = \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \ldots \in (0,1)$. The derivative f' vanishes at $\lambda_0 = \frac{1}{\sqrt{2}+1} = \sqrt{2}-1 = 0.4142\ldots$ On the interval $(0,1)$: f is increasing on the interval $(0,\lambda_0)$ and decreasing on the interval $(\lambda_0,1)$; it attains a unique maximum at $\lambda = \lambda_0$. Since $\lambda_0 \in (\frac{1}{3},\frac{1}{2})$, we have

$$f(\lambda) \le \max\left(f\left(\frac{1}{3}\right), f\left(\frac{1}{2}\right)\right) = f\left(\frac{1}{3}\right) = f\left(\frac{1}{2}\right) = \frac{2\sqrt{2}+3}{\sqrt{5}} = 2.6065\ldots$$

If $x = 2$, $\lambda = \frac{2}{y}$, where $y = 3,4,5,6,\ldots$, and so $\lambda = \frac{2}{3}, \frac{2}{4}, \frac{2}{5}, \frac{2}{6}, \ldots \in (0,1)$. Since $\lambda_0 \in (\frac{2}{5},\frac{2}{4})$, we have

$$f(\lambda) \le \max\left(f\left(\frac{2}{5}\right), f\left(\frac{1}{2}\right)\right) = f\left(\frac{2}{5}\right) = \frac{5\sqrt{2}+7}{\sqrt{29}} = 2.6129\ldots$$

If $x = 3$, $\lambda = \frac{3}{y}$, where $y = 4,5,6,7,8,9,\ldots$, and so $\lambda = \frac{3}{4}, \frac{3}{5}, \frac{3}{6}, \frac{3}{7}, \frac{3}{8}, \ldots \in (0,1)$. Since $\lambda_0 \in (\frac{3}{8},\frac{3}{7})$, we have

$$f(\lambda) \le \max\left(f\left(\frac{3}{8}\right), f\left(\frac{3}{7}\right)\right) = f\left(\frac{3}{7}\right) = \frac{5\sqrt{2}+7}{\sqrt{29}} = 2.6129\ldots$$

We have thus shown that for $x \leq 3$ we have $\delta_G(u,v) \leq (5\sqrt{2}+7)\,29^{-1/2}$. Let now $x \geq 4$ for the remainder of the proof. By the symmetry of the spanner construction (recall its periodicity modulo 4), it suffices to consider four cases:

Case 1: $u = (1,0)$. Connect u to v using (segments are listed cumulatively): $\lceil \frac{x}{4} \rceil$ upward-right (diagonal) segments, $x - \lceil \frac{x}{4} \rceil$ unit horizontal segments (going right), and $y - \lceil \frac{x}{4} \rceil$ upward two-segment zig-zags of length $\sqrt{2}+1$ each, as shown in Fig. 2.16 (left). We thus have

$$\begin{aligned} |\pi(u,v)| &\leq \left(y - \left\lceil \frac{x}{4} \right\rceil\right)\left(\sqrt{2}+1\right) + \left\lceil \frac{x}{4} \right\rceil \sqrt{2} + \left(x - \left\lceil \frac{x}{4} \right\rceil\right) \\ &= y\left(\sqrt{2}+1\right) + \left(x - 2\left\lceil \frac{x}{4} \right\rceil\right) \leq y\left(\sqrt{2}+1\right) + \frac{x}{2}. \end{aligned} \tag{2.7}$$

Consequently, by Fact 1 we have

$$\delta_G(u,v) \leq \frac{\sqrt{2}+1+\lambda/2}{\sqrt{1+\lambda^2}} \leq \sqrt{\left(\sqrt{2}+1\right)^2 + \left(\frac{1}{2}\right)^2} = 2.4654\ldots \tag{2.8}$$

Case 2: $u = (2,0)$. Connect u to v using: $\lfloor \frac{x}{4} \rfloor$ upward-right (diagonal) segments, $x - \lfloor \frac{x}{4} \rfloor$ unit horizontal segments (going right), and $y - \lfloor \frac{x}{4} \rfloor$ upward zig-zags of length $\sqrt{2}+1$ each; see Fig. 2.16 (right). Observe that for $x \geq 4$ we have $\lfloor \frac{x}{4} \rfloor \geq \frac{x}{7}$. It follows that

$$\begin{aligned} |\pi(u,v)| &\leq \left(y - \left\lfloor \frac{x}{4} \right\rfloor\right)\left(\sqrt{2}+1\right) + \left\lfloor \frac{x}{4} \right\rfloor \sqrt{2} + \left(x - \left\lfloor \frac{x}{4} \right\rfloor\right) \\ &= y\left(\sqrt{2}+1\right) + \left(x - 2\left\lfloor \frac{x}{4} \right\rfloor\right) \leq y\left(\sqrt{2}+1\right) + \frac{5x}{7}. \end{aligned} \tag{2.9}$$

Consequently, by Fact 1 we have

$$\delta_G(u,v) \leq \frac{\sqrt{2}+1+5\lambda/7}{\sqrt{1+\lambda^2}} \leq \sqrt{\left(\sqrt{2}+1\right)^2 + \left(\frac{5}{7}\right)^2} = 2.5176\ldots \tag{2.10}$$

Case 3: $u = (3,0)$. Connect u to v using: $\lfloor \frac{x+1}{4} \rfloor$ upward-right (diagonal) segments, $x - \lfloor \frac{x+1}{4} \rfloor$ unit horizontal segments (going right), and $y - \lfloor \frac{x+1}{4} \rfloor$ upward zig-zags of length $\sqrt{2}+1$ each;

see Fig. 2.17 (left). Observe that for $x \geq 3$ we have $\lfloor \frac{x+1}{4} \rfloor \geq \frac{x}{7}$. It follows that

$$\begin{aligned} |\pi(u,v)| &\leq \left(y - \left\lfloor \frac{x+1}{4} \right\rfloor\right)\left(\sqrt{2}+1\right) + \left\lfloor \frac{x+1}{4} \right\rfloor \sqrt{2} + \left(x - \left\lfloor \frac{x+1}{4} \right\rfloor\right) \\ &= y\left(\sqrt{2}+1\right) + \left(x - 2\left\lfloor \frac{x+1}{4} \right\rfloor\right) \leq y\left(\sqrt{2}+1\right) + \frac{5x}{7}. \end{aligned} \tag{2.11}$$

Consequently, $\delta_G(u,v) \leq \sqrt{\left(\sqrt{2}+1\right)^2 + \left(\frac{5}{7}\right)^2} = 2.5176\ldots$ follows as in (2.10).

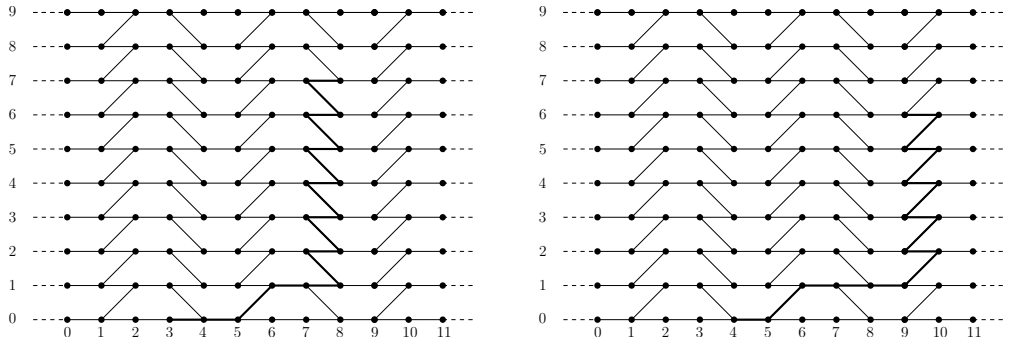

Figure 2.17: Left: a path connecting $(3,0)$ with $(8,7)$. Right: a path connecting $(4,0)$ with $(9,6)$.

Case 4: $u = (4,0)$. Connect u to v using: $\lfloor \frac{x+2}{4} \rfloor$ upward-right (diagonal) segments, $x - \lfloor \frac{x+2}{4} \rfloor$ unit horizontal segments (going right), and $y - \lfloor \frac{x+2}{4} \rfloor$ upward zig-zags of length $\sqrt{2}+1$ each; see Fig. 2.17 (right). Observe that for $x \geq 2$ we have $\lfloor \frac{x+2}{4} \rfloor \geq \frac{x}{7}$. It follows that

$$|\pi(u,v)| \leq y\left(\sqrt{2}+1\right) + \left(x - 2\left\lfloor \frac{x+2}{4} \right\rfloor\right) \leq y\left(\sqrt{2}+1\right) + \frac{5x}{7}. \tag{2.12}$$

Consequently, $\delta_G(u,v) \leq \sqrt{\left(\sqrt{2}+1\right)^2 + \left(\frac{5}{7}\right)^2} = 2.5176\ldots$ follows as in (2.10).

We have thus shown that for any $x, y \geq 0$, we have $\delta_G(u,v) \leq (5\sqrt{2}+7)\,29^{-1/2}$.

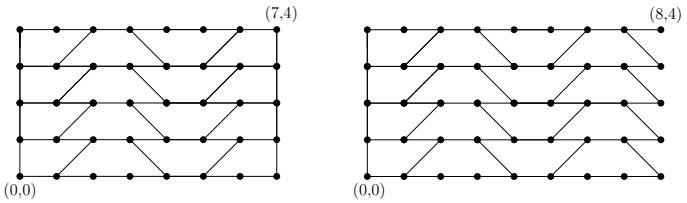

Figure 2.18: Connecting the points in $\Lambda(8,5)$ (left) and $\Lambda(9,5)$ (right).

It is easy to check that the same upper also holds for finite sections, $\Lambda(m,n)$ of Λ; see

Fig. 2.18.

Hexagonal lattice. We present a degree 3 spanner that has stretch factor exactly 3. Classify the points in Λ into two types. A point $u \in \Lambda$ is of Type I if the edge between the points $u = (a, b)$ and $(a - 0.5, b - \sqrt{3}/2)$ is present otherwise it is of Type II. Now let $u = (a, b)$ and $v = (c, d)$ be any two points of Λ. Observe that $|a - c| = m/2, m \in \mathbf{N}$ and $|b - d| = \sqrt{3}n/2, n \in \mathbf{N}$. Clearly, if $b = d$, $\delta_G(u, v) = 1$. Next, in each of the following remaining cases, we show that $\delta(u, v) \leq 3$.

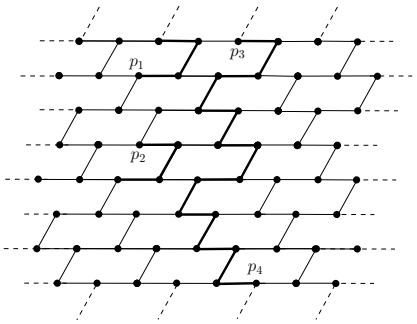

Figure 2.19: A degree 3 plane spanner G on the infinite hexagonal lattice; p_1 is of Type I and p_2 is of Type II.

Case 1: If $|b - d| = \sqrt{3}/2$, then

$$\delta_G(u, v) \leq \frac{2.5 + |a - c|}{\sqrt{(a - c)^2 + \left(\frac{\sqrt{3}}{2}\right)^2}} \leq \frac{2.5 + 0.5}{\sqrt{0.25 + 0.75}} = 3.$$

Here 2.5 is the maximum distance taken to transfer from the line $y = b$ to the line $y = d$. This can be easily verified by considering u either as a Type I point or as a Type II point. Refer to Fig. 2.19.

Case 2: If $a = c$, observe that $|b - d| = k\sqrt{3}, k \in \mathbf{Z}^+$. When $|b - d| = \sqrt{3}$, the shortest path between u and v has length either 3 or 5 (see Fig. 2.19). Thus, $\delta_G(u, v) \leq 5k/k\sqrt{3} = 2.8867\ldots$

Case 3: Now assume that $|a - c| \geq 0.5, |b - d| \geq \sqrt{3}$. We trace out a path from u to v. Observe that since $|b - d|/(\sqrt{3}/2) = n, n \in \mathbf{N}$, the shortest path from the line $y = b$ to the line

$y = d$ starting from u consists of at most $2n$ unit segments. Thus,

$$\delta_G(u, v) \leq \frac{(|a - c| + 0.5) + 4|b - d|/\sqrt{3}}{\sqrt{(a - c)^2 + (b - d)^2}},$$

since the length of the shortest path from u to v is at most the length of the path consisting of a straight line horizontal sub-path of length $|a-c|+0.5$ and a sub-path of length $2|b-d|/(\sqrt{3}/2)$. While tracing out the path α from u to v, the next point in α is chosen in a way such that it is the closest to the vertical line $x = u$. Since the shortest path from the line $y = b$ to the line $y = d$ starting from u may have its endpoint on $y = d$ at most $|a - c| + 0.5$ away from v in terms of x-coordinate, an adjusting factor of 0.5 suffices. Write $x = |a - c|$ and $y = |b - d|$; since $x \geq 0.5$ and $y \geq \sqrt{3}$, by Fact 1 we obtain:

$$g(x, y) = \frac{x + 0.5 + 4y/\sqrt{3}}{\sqrt{x^2 + y^2}} \leq \frac{x + 4y/\sqrt{3}}{\sqrt{x^2 + y^2}} + \frac{0.5}{\sqrt{0.5^2 + 3}} \leq 2.7939\ldots$$

Equality is attained for point pairs such as p_3, p_4 in Fig. 2.19, where $|a - c| = 0.5$ and $|b - d| = \frac{\sqrt{3}}{2}(4n + 3)$, $n \in \mathbf{N}$, with p_3 of Type II and p_4 of Type I.

It is easy to check that the above spanner is "shorter" than the one analyzed in Section 2.4 which has better stretch factor. The average cost (length) per vertex in this case is smaller: $1/2 + 1/2 + 1/2 = 3/2$, while that of the previous spanner (analyzed in Section 2.4) is $1/2 + 1/2 + \sqrt{3}/2 = 1.8660\ldots$

Chapter 3

Cutting out polygon collections with a saw

3.1 Introduction

The problem of efficiently cutting out a simple polygon P drawn on a planar piece of material (such as wood, paper, glass) Q, was introduced by Overmars and Welzl in their seminal paper [21] from 1985. Since then, the problem has attracted the interest of many computational geometers.

A saw cut may split (divide) Q into a number of pieces—those that lie left of the cut and those that lie right of the cut. In some situations, the saw may stop short of splitting Q, in which

case, the material remains as one solid piece. In any case we do not allow a cut to run through the interior of P. Several variants have been studied, primarily depending on the cutting tools used [1, 3, 5, 7, 10, 11, 12, 13, 14, 21, 24]: line cuts, ray cuts and segment cuts; they are described subsequently. In saw cutting, i.e., in all three models above, turns are impossible. The type of tool used in cutting determines the class of polygons that can be cut within that model.

The measures of efficiency commonly considered in polygon cutting are the total length of the cuts and the total number of cuts. Polygon cutting problems are useful in industry applications such as metal sheet cutting, paper cutting, furniture manufacturing and numerous other areas of engineering, where smart cutting techniques with high efficiency may result in the reduction of production costs. For instance, reducing the total length of the cuts may result in lesser power requirement and extend the life of the cutting tool. Similarly, reducing the total number of cuts may save cutting time and extend the life of the cutting tool.

3.1.1 Line cuts, ray cuts, and segment cuts

A *line cut* (also called a *guillotine cut*) is a line that does not go through P and divides the current piece of material containing P (initially Q) into two pieces. For cutting P out of Q by line cuts, P must be convex. An instance of a cutting sequence using line cuts appears in Fig. 3.1. The most studied efficiency measure for line cutting is the total length of the cuts and several approximation algorithms have been obtained [1, 3, 5, 7, 10, 11, 12, 13, 21, 24], including a PTAS proposed by Bereg et al. [3].

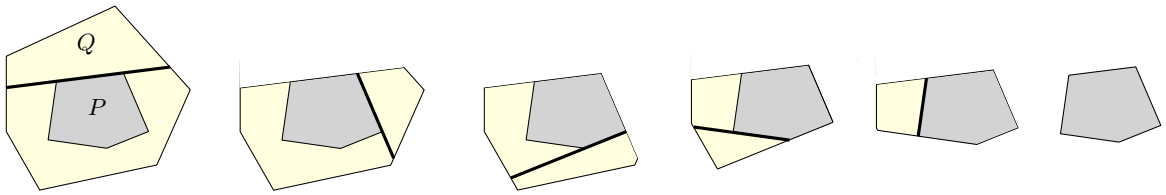

Figure 3.1: Cutting a convex polygon P out of Q using 5 line cuts.

A *ray cut* comes from infinity and can stop at any point outside P, again, not necessarily splitting the piece of material into pieces. Ray cuts are usually used to cut out non-convex poly-

gons; however, not all non-convex polygons can be cut by ray cuts. The following observation gives a necessary and sufficient condition for ray-cuttability, in the case of a single polygon; see Fig. 3.2. Several approximation algorithms can be found in [7, 10, 11, 24].

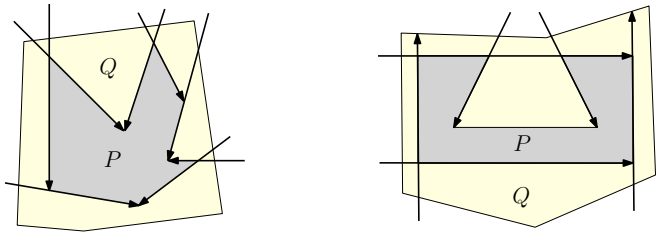

Figure 3.2: Left: a ray-cuttable polygon. Right: a polygon which is not ray-cuttable or segment-cuttable.

Observation 3.1. *A polygon P drawn on a planar material is ray-cuttable if and only if every edge of P that has some material adjacent to it, can be extended to infinity from one of its endpoints without passing through the interior of P.*

A *segment cut* is similar to a ray cut, but is not required to start at infinity, it may start at some finite point. The *segment saw* (also referred to as *circular saw* in [11]) is abstracted as a line segment, which cuts through material when moved along its supporting line. Before executing a segment cut, the saw needs to be placed. A small example of a cutting sequence appears in Fig. 3.3. Recall that saw turns are impossible during a cut; however, if a small free space within Q is available, a segment cut can be initiated there by maneuvering (i.e., rotating) the saw. The space required for maneuvering the saw is proportional to the length of the saw. The problem of cutting a polygon by a segment saw was introduced by Demaine et al. [11] in 2001. The authors gave a characterization of the class of polygons cuttable by a (possibly short) segment saw: a polygon is cuttable in this model, i.e., by a sufficiently short segment saw, if and only if it does not have two adjacent reflex vertices (with interior angle $> \pi$).

Note that ray-cuttability is not equivalent with segment-cuttability. For instance the polygon P in Fig. 3.3 is segment-cuttable but not ray-cuttable; indeed, the condition specified by Observation 3.1 is not fulfilled by two edges of P at the bottom of the pocket.

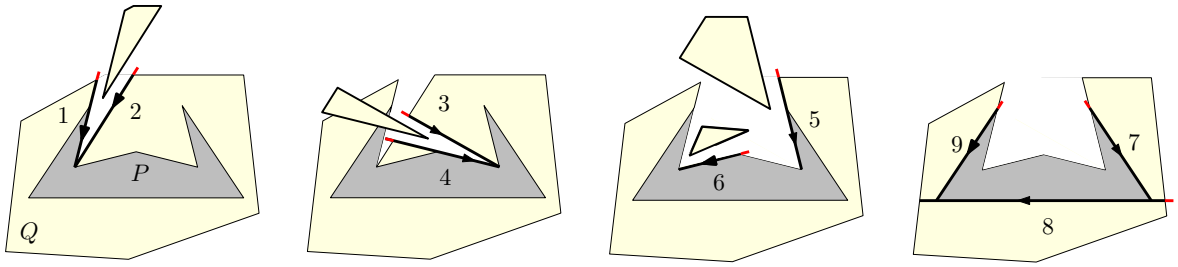

Figure 3.3: A cutting sequence with a segment saw (in red) consisting of 9 cuts. The polygon is not ray-cuttable.

For ease of analysis the length of the saw is assumed to be arbitrarily small, i.e., the segment abstracting the saw is as short as needed. Consequently, a segment cut can be initiated from an arbitrarily small available free space. In this model, several parts may result after a cut is made, and any of them can be removed (lifted) from the original plane. Moreover, free space may appear within the pieces of material from where future segment cuts can be initiated. The cutting process may continue independently on any of the separated pieces of material, if resulting parts contain subcollections of a larger collection to be cut out.

3.1.2 Our results and related work

Demaine et al. [11] presented an algorithm for cutting P out of its convex hull using segment cuts with a total number of cuts and total length of the cuts within constant factors of the respective optima. With regard to the total number of cuts, this number is within 2.5 times the respective optimum. Dumitrescu and Hasan [14] improved the approximation ratio on the total number of cuts from 2.5 to 2. Moreover, the new approximation guarantee is in a stronger sense than that offered by Demaine et al. [11]. While theirs achieves ratio 2.5 for cutting out P from its convex hull, the algorithm in [14] achieves ratio 2 for cutting out P from any enclosing polygon Q.

With regard to the total length measure of the cuts, Demaine et al. [11] reported that their algorithm achieves the same ratio, 2.5, as for the total number of cuts. Here we point out that the approximation factor on the total length of the cuts for their algorithm (as implied by their

proof) is in fact 3. In Section 3.2 we show that with a little care, one can recover the claimed ratio 2.5; it requires a small change in their algorithm to obtain:

Theorem 3.1. *Given a segment-cuttable polygon P, drawn on a planar piece of material Q, P can be cut out of Q by an arbitrarily short segment saw with the total length of the cuts within 2.5 times the optimum.*

We now proceed to our main results regarding collections of polygons. In the conclusion of their paper, Demaine et al. [11] left several open questions. Here we focus on one of them: "What collections of nonoverlapping polygons in the plane can be simultaneously cut out by a circular saw?"[1] The authors remarked that the problem is nontrivial when some of the polygons share edges. We study the case of axis-parallel rectangles in Section 3.3, where it is shown that the answer is always positive. Moreover, Section 3.4 gives evidence that the problem is nontrivial even if the polygons do not share edges.

A collection $\mathcal{P}$ of disjoint polygons drawn on a planar piece of material is *segment-cuttable* (or cuttable by segments) if there exists a sequence of segment cuts after which every polygon in the collection is cut out (along its sides) as a separate piece. Otherwise, we say that the collection is *uncuttable* by segments. Cuttability with other tools (such as arbitrarily short segment saw, rays or lines) are defined similarly. The following observations are in order.

1. If a collection of polygons is cuttable by rays, then it is also cuttable by a segment saw of any length.

2. If a collection of polygons is cuttable by lines, then it is also cuttable by rays.

3. If a collection of polygons is cuttable by a segment saw (segment) s of length $|s|$, then it is also cuttable by any segment saw of smaller length.

It is easy to draw collections of disjoint convex polygons (even with axis-parallel rectangles) that are uncuttable by line cuts. Motivated by this state of affairs, Pach and Tardos have studied

[1] Recall, this means cuttable by some (possibly short) segment saw.

the problem of separating a large subfamily from a given family of pairwise disjoint compact convex sets on a sheet of glass, using line cuts [22]. For the case of axis-parallel rectangles, the authors show how to separate a subcollection with $\Omega(n/\log n)$ members out of given n. From the other direction, there exist instances of n rectangles such that at most cn of them can be separated in this model, where $c < 1$ is a positive constant. Far weaker guarantees, sublinear in n, can be made for arbitrary convex polygons. For other related results see [2, 8, 9, 19, 23]. In Section 3.3 we prove:

Theorem 3.2. *Given a collection $\mathcal{R}$ of n disjoint axis-parallel rectangles drawn on a planar piece of material, $\mathcal{R}$ is cuttable by rays, so in particular by a segment saw of any length. The cutting sequence can be computed in $O(n \log n)$ time and uses at most $4n$ ray cuts, which is optimal in the worst case.*

In Section 3.4 we exhibit some uncuttable collections of disjoint polygons.

Theorem 3.3. *There exist collections of disjoint polygons that are uncuttable by any segment saw. Such collections can be realized with convex or not necessarily convex polygons, and even with rectangles, or triangles.*

On the other hand we have the following positive result (in Section 3.4).

Theorem 3.4. *Given a collection $\mathcal{P}$ of segment-cuttable polygons drawn on a planar piece of material such that no two polygons in $\mathcal{P}$ touch each other, $\mathcal{P}$ is always cuttable by a sufficiently short segment saw.*

In Section 3.5 we prove:

Theorem 3.5. *Consider a collection $\mathcal{P}$ of k disjoint polygons with n vertices in total drawn on a planar piece of material Q. Then there exists an algorithm that computes in $O(n^6)$ time a suitable sequence of ray cuts for cutting the polygons in $\mathcal{P}$ out of Q when $\mathcal{P}$ is ray-cuttable and otherwise reports $\mathcal{P}$ as uncuttable.*

The same algorithm can be adapted to the case of line cuts resulting in a faster running time of $O(n^4)$. We conclude in Section 3.6 with some open problems.

3.2 Cutting out a single polygon using a segment saw [2]

In this section we prove Theorem 3.1. Let OPT and ALG denote the lengths of an optimal cutting sequence and that of a given algorithm being analyzed. We first show that the approximation algorithm of Demaine et al. [11] for cutting out a polygon achieves ratio 3 in the length measure. Moreover, this ratio cannot be improved as long as one uses the trivial lower bound on OPT given by the perimeter of P.

The algorithm cuts the polygon P out of its convex hull, $\mathtt{conv}(P)$, by following the boundary of P (in a chosen fixed direction, clockwise or counterclockwise) and removing the material in the pockets of P (the maximal connected components of $\mathtt{conv}(P) \setminus P$). The pockets of P are thereby cut out sequentially; we refer the reader to [11] for details. Note that reflex vertices in a pocket K correspond to convex vertices of the target polygon P, and vice versa. By the characterization of Demaine et al. [11], P is cuttable by a (short) segment saw if and only if no two reflex vertices of P are consecutive; equivalently, no two convex vertices of any pocket K are consecutive.

Let a_i and b_i (in this order) be the two edges incident to a convex vertex of K, along the chosen direction. By the above characterization, any two terms a_i and b_j are disjoint, i.e., they denote distinct edges of K. Let $i = 1, \ldots, k$ be the sequence of convex vertices along the same direction. Write $A = \sum_{i=1}^{k} |a_i|$ and $B = \sum_{i=1}^{k} |b_i|$. The cutting algorithm (illustrated in [11, Fig. 5, p. 74]) gives a cost arbitrarily close to $2A + 3B$, while obviously $\mathsf{OPT} \geq A + B$ (the trivial lower bound on OPT). If $A \to 0$ and $A \ll B$ then the ratio can be arbitrarily close to 3:

$$\frac{\mathsf{ALG}}{\mathsf{OPT}} \leq \frac{2A + 3B}{A + B} = 2 + \frac{B}{A + B} \to 3.$$

Moreover, such a polygon (with $A \to 0$ and $A \ll B$) can be constructed by choosing small $|a_i|$ and large $|b_i|$ in the pockets and minimizing other parts on the perimeter in comparison with the pockets. See Fig. 3.4.

[2] The corrections to the Theorem 3.1 has been done in collaboration with Masud Hasan, refer to [15].

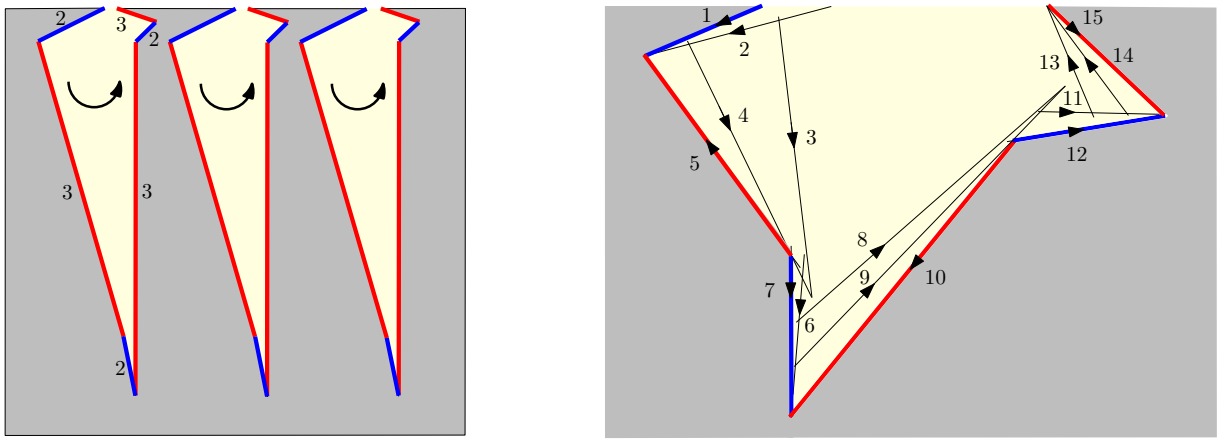

Figure 3.4: Left: a polygon with many thin pockets for which cutting length is close to 3 times the perimeter. The B-parts (which generate cuts of length close to $3B$) are drawn in red lines; the pockets are processed in the order indicated by the arrows. The number of cuts (2 or 3) corresponding to each edge is specified. Right: details of cutting out one of the pockets; for clarity the internal angles and edge-lengths have been altered.

The revised algorithm chooses the best direction for cutting out each pocket K from the two possible, clockwise or counterclockwise, namely the direction for which $A \geq B$. Recall that the cutting algorithm gives a cost arbitrarily close to $2A + 3B$, while $\mathsf{OPT} \geq A + B$. Hence the ratio is arbitrarily close to

$$\frac{\mathsf{ALG}}{\mathsf{OPT}} \leq \frac{2A + 3B}{A + B} = 2 + \frac{B}{A + B} \leq 2.5,$$

as desired.

Saw Length. As remarked by Demaine et al. [11], it would be interesting to compute the length of the largest segment saw that can be used to cut out a cuttable polygon. While this question remains unresolved, the following observation gives an easy upper bound.

Observation 3.2. *Let P be a segment-cuttable polygon drawn on a piece of material Q (in its interior) and E be the set of its edges. Let $l_1(e)$ and $l_2(e)$ be the lengths of the two extensions of $e \in E$ at both endpoints until they intersect the interior of P and let $L(e) = max\{l_1(e), l_2(e)\}$. If an extension does not intersect the interior, we consider its length to be infinity. If L is the maximum length*

of a segment saw that can be used to cut P out of Q, then

$$L \leq \min_{e \in E} L(e).$$

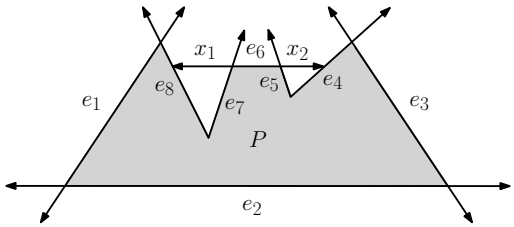

Figure 3.5: A line represents the two end extensions and a ray represents one end extension of an edge. If $x_1 \geq x_2$, $L \leq x_1$ is needed.

Proof. For an illustration refer to Fig. 3.5. To remove the material adjacent to an edge e, we need to place the saw on one of the extensions of e and the maximum length of the saw that can be used for e is the maximum of the lengths of the two extensions, i.e., $L(e)$. Since a fixed length saw is used for every edge of P, we need to take the minimum of all $L(e)$, $e \in E$. □

3.3 Cutting out a collection of axis-parallel rectangles using a segment saw

In this section we prove Theorem 3.2. As it turns out, the problem of cutting out a collection of axis-parallel rectangles by a segment saw is very much related to the problem of separating such a collection by moving the rectangles, one at at time, to infinity, using translations. We start by recalling the following classical result of Guibas and Yao [18] concerning translations of rectangles in a common direction.

Lemma 3.1 (Guibas and Yao [18])**.** *Let $\mathcal{R}$ be any set of n disjoint axis-parallel rectangles in the plane, and θ be any direction. Then there is an ordering $R_1, \ldots, R_n$ of the rectangles such that R_i can be moved continuously to infinity in direction θ without colliding with the rectangles R_j, $1 \leq i < j \leq n$. Such an ordering can be computed in $O(n \log n)$ time.*

The reader can verify that the proof of Lemma 3.1 in [18] implies the following stronger result; see also [16, Lemma 3].

Lemma 3.2 (Guibas and Yao [18])**.** *For any set of n disjoint axis-parallel rectangles in the plane, there is an ordering $R_1, \ldots, R_n$ of the rectangles such that R_i can be moved continuously to infinity in any direction between 0 and $\pi/2$ without colliding with the rectangles R_j, $1 \leq i < j \leq n$. Such an ordering can be computed in $O(n \log n)$ time.*

In Lemma 3.2 the set of directions under discussion make the closed interval $[0, \pi/2]$. It is now convenient to reformulate this lemma in terms of our interest.

Notation Given two points $p, q \in \mathbf{R}^2$, p *dominates* q if the inequalities $x(p) > x(q)$ and $y(p) > y(q)$ hold among their x- and y-coordinates. Given two axis-parallel rectangles R', R'', write $R'' >_x R'$ if there exists a vertical line that separates R' and R'', so that R' lies in the left (closed) halfplane and R'' lies in the right (closed) halfplane determined by the line.

Lemma 3.3. *For any set of n disjoint axis-parallel rectangles in the plane, there is an ordering $R_1, \ldots, R_n$ of the rectangles such that R_i is unblocked in any direction between 0 and $\pi/2$ by any of the rectangles R_j, $1 \leq i < j \leq n$. Such an ordering can be computed in $O(n \log n)$ time.*

To be precise, R_i is *unblocked* in any direction between 0 and $\pi/2$ by any of the rectangles R_j, $i < j \leq n$, if and only if no vertex of such a rectangle dominates the lower left corner of R_i; see Fig. 3.6.

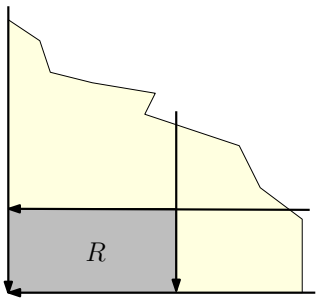

Figure 3.6: R is unblocked hence it is cuttable.

The two-step algorithm from [18] for computing the order is as follows.

1. Sort the rectangles by decreasing order of their y-coordinate of the top side; let $R_1, \ldots, R_n$ be the resulting order.

2. Start with an empty list L and add the rectangles R_i, for $i = 1, \ldots, n$, in this order. Place each new rectangle R in the first (i.e., leftmost) position in L consistent to the constraint that $R >_x S$ for every rectangle S following R in L.

An illustration of the ordering produced appears in Fig. 3.7. An obvious implementation of STEP 2 takes $O(n^2)$ time, however, Guibas and Yao [18] showed that it is possible to use balanced trees in a non-trivial way to reduce the time to $O(n \log n)$. Consequently, the two-step algorithm runs in $O(n \log n)$ time. Using the ordering guaranteed by Lemma 3.3, we get our desired result.

Proof of Theorem 3.2 We use the ordering provided by Lemma 3.3 and cut out rectangles one by one in this order; there are n iterations, $i = 1, \ldots, n$. The following invariant is maintained: in iteration i, rectangles $R_1, \ldots, R_{i-1}$ have been cut out (i.e., each has detached on a separate piece of material), and the current piece of material contains the subcollection $R_i, \ldots, R_n$. Observe that in iteration i, rectangle R_i is unblocked in any direction between 0 and $\pi/2$, hence it is cuttable as follows; refer again to Fig. 3.6.

We execute two ray cuts from infinity (or from the boundary of the material): one vertical (going down) and one horizontal (going left); the two cuts meet at the lower left corner of R_i. Further, each ray cut is extended until it hits the interior of a rectangle or the boundary of the current piece of material. The effect is detaching R_i from the piece of material containing the remaining rectangles R_j, $i < j \leq n$. The piece containing R_i contains no other rectangles, so two more cuts along two sides of R_i suffice to completely separate R_i completely, for a total of at most 4 cuts per rectangle. The process is continued until all rectangles are cut out in this way, with at most $4n$ cuts overall.

Clearly some collections require at least 4 cuts for each rectangle, e.g., if no two rectangle sides are collinear, for a total of at least $4n$ cuts. Hence the number of cuts executed in the

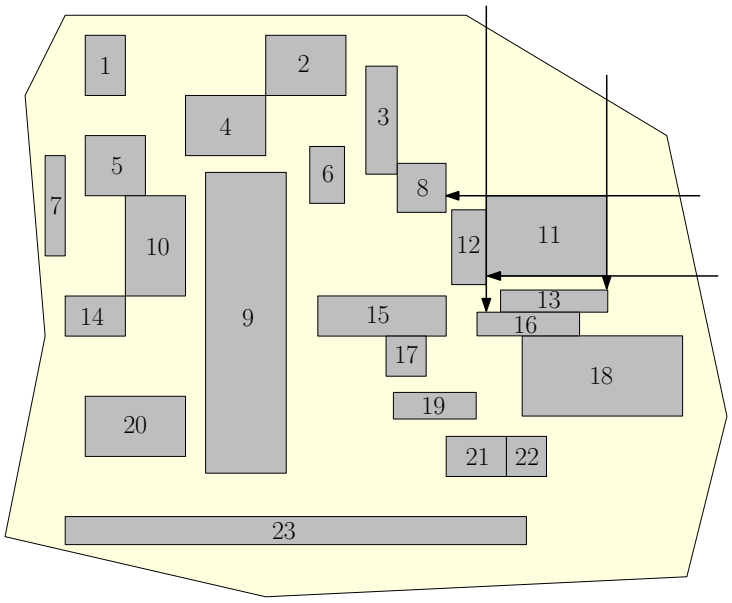

Figure 3.7: STEP 2 of the algorithm. After $R_1, \ldots, R_{23}$ are successively inserted into L, the list is $11, 13, 12, 16, 18, 22, 8, 3, 2, 6, 15, 17, 19, 21, 4, 9, 1, 5, 10, 14, 20, 7, 23$. The first rectangle to be cut out is 11.

algorithm is worst-case optimal.

Remark There are cases when as few as $\Theta(\sqrt{n})$ cuts suffice, for instance when the rectangles are arranged in a square grid formation with their sides aligned.

3.4 Cuttable and uncuttable collections by a segment saw

It is easy to exhibit collections of disjoint polygons that are uncuttable by a segment saw, especially if one uses non-convex polygons or convex polygons with many sides. The problem becomes more interesting when one restricts the number of sides of the polygons or their shape. In this section we prove Theorem 3.3. An uncuttable collection of rectangles (in arbitrary orientations) is shown in Fig. 3.8 (left); this example has $n = 12$ rectangles, 6 drawn on the sides of a regular hexagon in its exterior, and 6 drawn in its interior. A similar pattern can be realized for every $n \geq 10$ (with $\lfloor n/2 \rfloor$ rectangles on the outer boundary of the union and the remaining $\lceil n/2 \rceil$ inside). Observe that every possible cut that can be initiated from outside is blocked by

one of the small rectangles; moreover, all intersecting cuts are incident to some tangency point between two consecutive outer rectangles. Hence none of the rectangles can be separated. In

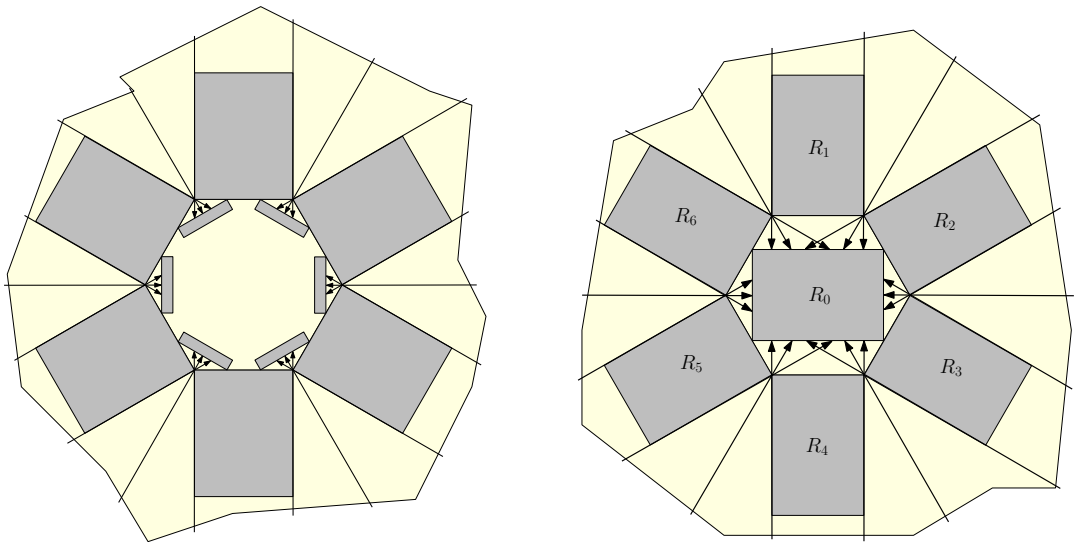

Figure 3.8: Left: an uncuttable collection of $n = 12$ rectangles; arrows represent saw cuts that fail in cutting out any rectangle. Right: a cuttable collection of $n = 7$ rectangles.

contrast, the similar looking construction with $n = 7$ rectangles shown in Fig. 3.8 (right) is cuttable by rays, hence in particular, by a segment saw of any length. This particular example has $n = 7$ rectangles, 6 drawn on the sides of a regular hexagon in its exterior, and one drawn in its interior. Start by separating R_1 by using two ray cuts aligned with the long sides of its left and right neighbor rectangles. Observe that these two rays cross each other, so the piece of material containing the top rectangle can be detached. Once one of the rectangles has been cut out separately, the rest can be easily cut out one by one, and so can the entire collection.

If we slightly modify the placements of the rectangles in Fig. 3.8 (left) so that no two rectangles touch each other, as shown in Fig. 3.9 (left), then the resulting collection becomes cuttable using a segment saw. Indeed, every outer rectangle can be separated out as shown in Fig. 3.9 (right). Next, using a similar approach, the inner rectangles can also be separated out. After separation, it is easy to remove any material adjacent to an edge of a rectangle.

This approach can be applied to any collection of segment-cuttable polygons where no two polygons touch each other. Separation can be achieved by using double cuts as shown in

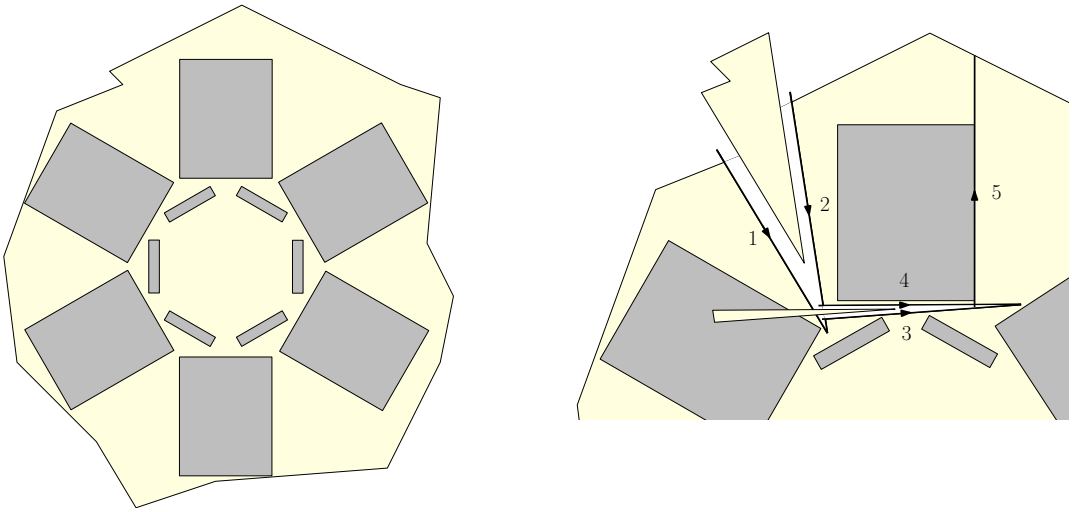

Figure 3.9: Left: a collection of $n = 12$ rectangles where no two rectangles touch each other. The collection is now cuttable by a (sufficiently short) segment saw. Right: a magnified view.

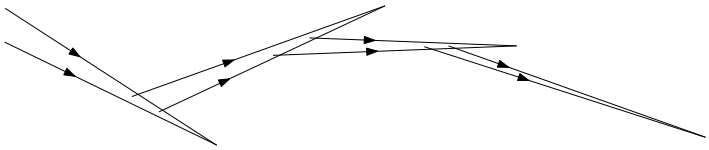

Figure 3.10: Double cuts (pairs of segment cuts of about the same length) can be used to cut along any polygonal line achieving separation of two or more polygons.

Fig. 3.10. After separation, the individual polygons can be cut out using the algorithm from [11]. We have thereby proved Theorem 3.4.

An uncuttable collection of triangles is shown in Fig. 3.11 (left). Such collections can be also realized with a larger number of triangles. For any $k \geq 4$, it is straightforward to draw uncuttable k-gon collections using $k + 1$ polygons $P_0, P_1, \ldots, P_k$. Constructions for $k = 4$ and $k = 6$ are shown in Fig. 3.11 (middle and right).

Remarks It is interesting to observe that separability by translations in a single direction holds for any collection of disjoint convex bodies; see also [4, Theorem 1], [16, Lemma 1], [20, Theorem 8.7.2]. Theorem 3.6 below appears in the work of Fejes Tóth and Heppes [17], but it can be traced back to de Bruijn [6]; the algorithmic aspects of the problem have been studied by Guibas and Yao [18].

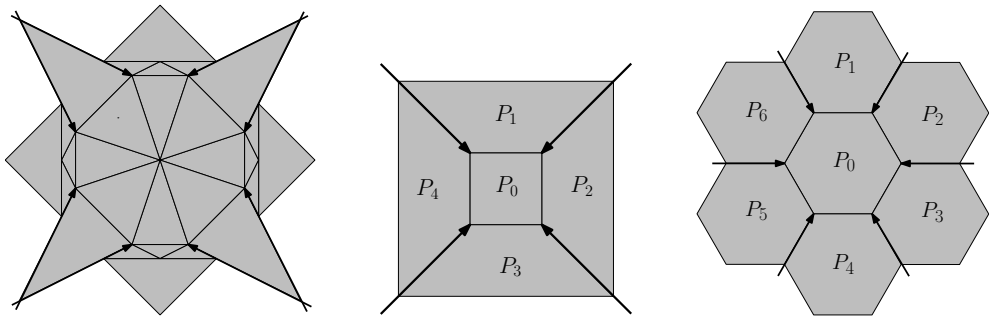

Figure 3.11: Uncuttable k-gon collections. Left: construction with triangles ($k = 3$). Middle: $k = 4$. Right: $k = 6$. Arrows represent the only possible useful cuts in the collection.

Theorem 3.6. [6, 17, 18] *Any set of n convex objects in the plane can be separated via translations all parallel to any given fixed direction, with each object moving only once. If the top and bottom points of each object are given, an ordering of the moves can be computed in $O(n \log n)$ time.*

In general, separability via translation does not imply cuttability. Observe that the collections in Fig. 3.8 (left) and 3.11 are not cuttable by a segment saw although they can be separated via translations along any fixed direction as stated in Theorem 3.6. On the other hand, the broader variant of separability for axis-aligned rectangles stated in Lemma 3.2 (so that R_i can be translated along any direction in the first quadrant) finally allows cuttability by rays for any family of axis-parallel rectangles.

3.5 Cutting out a collection of polygons using ray cuts

In this section we prove Theorem 3.5. Consider a collection $\mathcal{P}$ of k disjoint polygons $P_1, \ldots, P_k$, with n vertices in total, drawn on a planar piece of material Q. Recall that a ray cut comes from infinity and can stop at any point. We may assume that each P_i is ray-cuttable by itself; otherwise the collection is not ray-cuttable. (Note that a collection of convex polygons is not necessarily ray-cuttable; see Fig. 3.11.)

We present a polynomial-time algorithm that computes a suitable cutting sequence to cut the polygons in $\mathcal{P}$ out of Q using ray cuts when $\mathcal{P}$ is ray-cuttable and otherwise reports $\mathcal{P}$ as uncuttable.

A subcollection of polygons $\mathcal{P}' \subset \mathcal{P}$ drawn on a piece of material $Q' \subset Q$ is called *separated* if Q' is already detached from Q after executing a sequence of ray cuts. Observe that a ray cut can produce multiple (separated) subcollections; in that case, we say that the ray cut *achieves some separation*. In order to cut out the polygons in $\mathcal{P}$, it is necessary to separate out every polygon in the given collection.

When two ray cuts r, r' meet at a point p, the non-reflex angle between the ray cuts at p is called their *internal angle* and is denoted by $\theta(r, r')$. The wedge-shaped piece of material enclosed by r, r' when $\theta(r, r')$ is considered is denoted by $W(r, r')$; see Fig. 3.12.

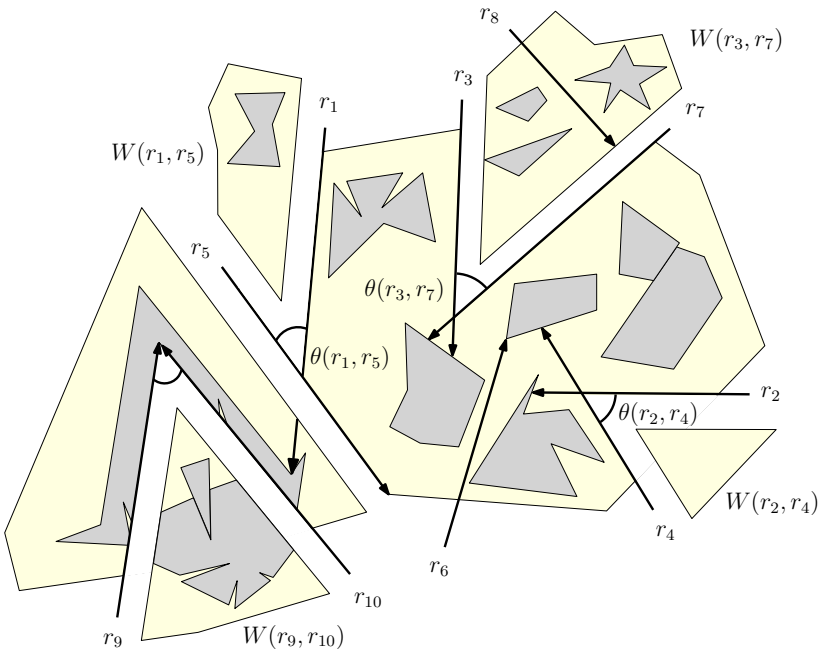

Figure 3.12: Let $\mathcal{S} = r_1, \ldots, r_{10}$ be a sequence of executed ray cuts. (r_3, r_7) and (r_9, r_{10}) are the only separating pairs; r_5 and r_8 are the only separating rays. However, (r_1, r_5) is not a separating pair since r_5 is a separating ray.

By slightly abusing the notation, we refer to the current piece of material as Q and the subcollection present on Q as $\mathcal{P}$. Consider a sequence $\mathcal{S}$ of ray cuts executed on Q. We assume that the detached pieces of Q, with or without polygons, are removed immediately after separation and are handled independently. Furthermore, we extend any ray cut $r \in \mathcal{S}$ until it hits the interior of a polygon in $\mathcal{P}$ or the boundary of Q.

A *separating ray* is a ray cut $r \in \mathcal{S}$ executed on Q such that its endpoint is on the boundary of

Q and each side of r contains a proper polygon subcollection of $\mathcal{P}$. A *separating pair (of rays)* is a pair of meeting ray cuts r, r' in $\mathcal{S}$ executed on Q such that $W(r, r')$ contains a proper polygon subcollection of $\mathcal{P}$ and r, r' are not separating rays. Refer to Fig. 3.12 for an illustration.

The following lemma shows that there is always a separating ray or a separating pair in any sequence $\mathcal{S}$ of ray cuts that achieves some separation. The ray cuts in $\mathcal{S}$ are executed until the first separation occurs. After separation each piece of material is handled independently.

Lemma 3.4. *Let $\mathcal{S} = r_1, \ldots, r_s$ be a sequence of ray cuts executed in this order and achieving some separation. Then either there is a separating ray or a separating pair in $\mathcal{S}$.*

Proof. Consider the subsequence $\mathcal{S}' = r_1, \ldots, r_t$ where $t \leq s$, such that the first separation occurs when r_t is executed. If r_t is a separating ray, we are done. Now assume that r_t is not a separating ray. Since separation is achieved only after r_t is executed, then it must be the case that r_t forms a separating pair with some previously executed ray cut $r_i \in \mathcal{S}'$, with $i < t$, i.e., $W(r_i, r_t) \cap \mathcal{P}$ is not empty. □

A ray is called *canonical* if it passes through at least two polygon vertices (not necessarily of the same polygon). Clearly, there are $O(n^2)$ canonical rays in the given configuration. Any valid ray cutting sequence will repeatedly partition the given polygon collection using separating pairs or separating rays until each polygon has been separated out. A cutting sequence need not to contain canonical ray cuts. However, the following lemma shows that any valid separation can be achieved by using canonical ray cuts only. This observation is key for designing a polynomial-time ray cutting algorithm, based on finding appropriate canonical ray cuts for separation.

Lemma 3.5. *Let (r_1, r_2) be a separating pair. Then there exists a pair of canonical ray cuts (r_1'', r_2'') that achieves the same effect as (r_1, r_2) in terms of separability.*

Proof. We transform the pair (r_1, r_2) into the pair (r_1'', r_2'') so that the two rays intersect at all times during the transformation; refer to Fig. 3.13. Let p be the intersection point of r_1, r_2 such that the pair separates $\mathcal{P}' \subset \mathcal{P}$. Translate r_1 towards $\mathcal{P}'$ until it touches a polygon vertex, say

a, and refer to this ray as r_1'. Similarly, translate r_2 towards $\mathcal{P}'$ and obtain the ray r_2' passing through c. Then (r_1', r_2') is still a separating pair, where r_1', r_2' intersect at p'.

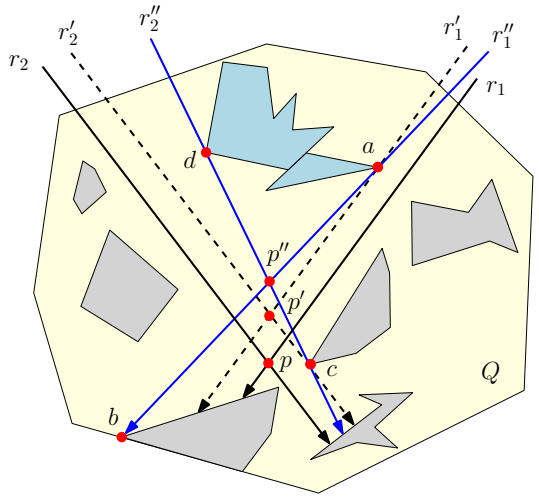

Figure 3.13: Separating the set of polygons $\mathcal{P}' \subset \mathcal{P}$ (in light blue) using the canonical pair (r_1'', r_2'').

For a given a ray r, let $\pi_{\text{left}}(r)$ denote the open half plane to the left of r and $\pi_{\text{right}}(r)$ denote the open half plane to the right of r. If r_1' passes through a only, we have the following two cases where r_1'' is obtained by rotating r_1' around a until it passes through a second polygon vertex b.

1. If $a \in \pi_{\text{left}}(r_2')$, rotate r_1' clockwise around a.

2. If $a \in \pi_{\text{right}}(r_2')$ or $a \in r_2'$, rotate r_1' counter-clockwise around a.

Similarly if r_2' passes through c only, we obtain r_2'' passing through vertices c and d by rotating r_2' around c in the following analogous way.

1. If $c \in \pi_{\text{left}}(r_1')$ or $c \in r_1'$, rotate r_2' clockwise around c.

2. If $c \in \pi_{\text{right}}(r_1')$, rotate r_1' counter-clockwise around c.

The canonical rays r_1'', r_2'' meet at p'' and can also separate P'. □

A rather straightforward argument also yields:

Lemma 3.6. *Let r be a separating ray. Then there exists a canonical separating ray that achieves the same partition.*

Algorithm for cutting According to Lemmas 3.4,3.5,3.6, the existence of a canonical separating ray is first determined. If at least one exists, an arbitrary ray is chosen and the corresponding cut is executed. If no canonical separating ray exists, the existence of a canonical separating pair is determined. If at least one exists, an arbitrary pair is chosen and the corresponding cuts are executed. After separation the algorithm continues independently and recursively on each of the newly formed pieces; it terminates when every polygon has been separated from the rest. If no canonical separating ray or canonical separating pair exists at some point during the algorithm, $\mathcal{P}$ is reported as uncuttable.

As mentioned earlier, each ray cut is extended until it hits the interior of a polygon or the boundary of the current piece of material. (In some instances separation can be only achieved with separating rays, e.g., for a collection of stacked congruent axis-aligned rectangles.)

After each polygon in $\mathcal{P}$ has been separated out, it is cut out using ray cuts. This step can be achieved by considering one end extensions of the edges that have some material attached to them, to form the necessary ray cuts (refer to Observation 3.1).

Analysis There are $O(n^2)$ canonical ray cuts and $O(n^4)$ canonical pairs of ray cuts in any given configuration. Hence we have $O(n^4)$ ways to cut in total. Verifying whether a canonical pair is a separating pair and a canonical ray is a separating ray can be done in $O(n)$ time. In the worst case, we need to execute $O(n)$ canonical separating pairs and rays. After each execution, the bookkeeping of separation and creation of new pieces can be performed in $O(n)$ time. Thus, separation of the polygons can be achieved in $O(n^6)$ time. After the separation step, in $O(n)$ time we can remove the pieces of material adjacent to the edges of the polygons. Hence the overall running time of the algorithm is $O(n^6)$. This completes the proof of Theorem 3.5. □

Line Cuts The above algorithm can be easily adapted for cutting out polygon collections using line cuts instead of ray cuts. Similar to the approach described above, we will use only canonical separating lines and since there are $O(n^2)$ of them, the algorithm would run in $O(n^4)$ time.

3.6 Open problems

1. The obvious remaining open problem is devising an algorithm, which, given a collection of disjoint polygons in the plane determines whether it is cuttable by a segment saw, and computes a suitable cutting sequence if it is. We conjecture that the problem admits a polynomial-time algorithm.

2. Can the cutting algorithm presented in Section 3.5 (or its analysis) be substantially improved? Is there a substantially faster algorithm?

References

[1] S. I. Ahmed, M. A. Islam, and M. Hasan, Cutting a cornered convex polygon out of a circle, *Journal of Computers* **5(1)** (2010), 4–11.

[2] N. Alon, M. Katchalski, and W. R. Pulleyblank, Cutting disjoint disks by straight lines, *Discrete Comput. Geom.* **4** (1989), 239–243.

[3] S. Bereg, O. Dăescu, and M. Jiang, A PTAS for cutting out polygons with lines, *Algorithmica* **53** (2009), 157–171.

[4] S. Bereg, A. Dumitrescu, and J. Pach, Sliding disks in the plane, *Internat. J. Comput. Geom. Appl.* **15(8)** (2008), 373–387.

[5] J. Bhadury and R. Chandrasekaran, Stock cutting to minimize cutting length, *European J. Oper. Res.* **88** (1996), 69–87.

[6] N. G. de Bruijn, Aufgaben 17 and 18 (in Dutch), Nieuw Archief voor Wiskunde **2** (1954), 67.

[7] R. Chandrasekaran, O. Dăescu, and J. Luo, Cutting out polygons, *Proceedings of the 17th Canadian Conference on Computational Geometry (CCCG)* (2005), pp. 183–186.

[8] J. Czyzowicz, E. Rivera-Campo, and J. Urrutia, Separating convex sets in the plane, *Discrete Comput. Geom.* **7** (1992), 189–195.

[9] J. Czyzowicz, E. Rivera-Campo, and J. Urrutia, Separation of convex sets, *Discrete Appl. Math.* **51(3)** (1994), 325–328.

[10] O. Dăescu and J. Luo, Cutting out polygons with lines and rays, *Internat. J. Comput. Geom. Appl.* **16(2-3)** (2006), 227–248.

[11] E. D. Demaine, M. L. Demaine, and C. S. Kaplan, Polygons cuttable by a circular saw, *Comput. Geom.* **20** (2001), 69–84.

[12] A. Dumitrescu, An approximation algorithm for cutting out convex polygons, *Comput. Geom.* **29** (2004), 223–231.

[13] A. Dumitrescu, The cost of cutting out convex n-gons, *Discrete Appl. Math.* **143** (2004), 353–358.

[14] A. Dumitrescu and M. Hasan, Cutting out polygons with a circular saw, *Internat. J. Comput. Geom. Appl.* **23(2)** (2013), 127–139.

[15] A. Dumitrescu, A. Ghosh, and M. Hasan, On Collections of Polygons Cuttable with a Segment Saw, *Proceedings of the International Conference on Algorithms and Discrete Applied Mathematics*, (CALDAM 2015), IIT Kanpur, India, February 2015, vol. 8959 of LNCS, pp. 58-68.

[16] A. Dumitrescu and M. Jiang, On reconfiguration of disks in the plane and other related problems, *Comput. Geom.* **46(3)** (2013), 191–202.

[17] L. Fejes Tóth and A. Heppes, Über stabile Körpersysteme (in German), *Compositio Mathematica* **15** (1963), 119–126.

[18] L. Guibas and F. F. Yao, On translating a set of rectangles, in *Computational Geometry*, F. Preparata (ed.), pp. 61–67, Vol. 1 of *Advances in Computing Research*, JAI Press, London, 1983.

[19] K. Hope and M. Katchalski, Separating plane convex sets, *Mathematica Scandinavica* **66** (1990), 44–46.

[20] J. O'Rourke, *Computational Geometry in C*, Cambridge University Press, 2nd edition, 1998.

[21] M. H. Overmars and E. Welzl, The complexity of cutting paper, *Proceedings of the 1st Annual ACM Symposium on Computational Geometry* (1985), pp. 316–321.

[22] J. Pach and G. Tardos, Cutting glass, *Discrete Comput. Geom.*, **24** (2000), 481–495.

[23] J. Pach and G. Tardos, Separating convex sets by straight lines, *Discrete Math.* **241(1-3)** (2001), 427–433.

[24] X. Tan, Approximation algorithms for cutting out polygons with lines and rays, *Proceedings of the 11th Annual International Computing and Combinatorics Conference*, Vol. 3595 of *LNCS*, Springer, (2005), pp. 534–543.

Chapter 4

Fence patrolling by mobile agents

4.1 Introduction

A set of k mobile agents with (possibly distinct) maximum speeds v_i ($i = 1, \ldots, k$) are in charge of *patrolling* a given region of interest. Patrolling problems find applications in the field of robotics where surveillance of a region is necessary. An interesting one-dimensional variant has been introduced by Czyzowicz et al. [7], where the agents move along a rectifiable Jordan curve representing a *fence*. The fence is either a *closed* curve (the boundary of a compact region in the plane), or an *open* curve (the boundary between two regions). For simplicity (and without loss of generality) it can be assumed that the open curve is a line segment and the closed curve is a circle. The movement of the agents over the time interval $[0, \infty)$ is described by a *patrolling schedule* (or *guarding schedule*), where the speed of the ith agent, a_i ($i = 1, \ldots, k$), may vary

between zero and its maximum value v_i in any of the two moving directions (left or right).

Given a closed or open fence of length ℓ and maximum speeds $v_1, \ldots, v_k > 0$ of k agents, the goal is to find a patrolling schedule that minimizes the *idle time* I, defined as the longest time interval in $[0, \infty)$ during which a point on the fence remains unvisited, taken over all points. A straightforward volume argument [7] yields the lower bound $I \geq \ell / \sum_{i=1}^{k} v_i$ for an (open or closed) fence of length ℓ. A *patrolling algorithm* computes a patrolling schedule for a given fence and set of speeds $v_1, \ldots, v_k > 0$.

For an open fence (line segment), Czyzowicz et al. [7] proposed a simple partitioning strategy, algorithm $\mathcal{A}_1$, where each agent moves back and forth perpetually in a segment whose length is proportional with its speed.

Algorithm $\mathcal{A}_1$. For a segment of length ℓ and k agents with maximum speeds $v_1, \ldots, v_k$, the algorithm partitions the segment into k pieces of lengths $\ell v_i / \sum_{j=1}^{k} v_j$, and schedules the ith agent to patrol the ith interval with speed v_i. Refer to Fig. 4.1 for an illustration.

Algorithm $\mathcal{A}_1$ has been proved to be optimal for uniform speeds [7], i.e., when all maximum speeds are equal. Algorithm $\mathcal{A}_1$ achieves an idle time $2\ell / \sum_{i=1}^{k} v_i$ on a segment of length ℓ, and so $\mathcal{A}_1$ is a 2-approximation algorithm for the shortest idle time. It has been conjectured [7, Conjecture 1] that $\mathcal{A}_1$ is optimal for arbitrary speeds, however this was disproved by Kawamura and Kobayashi [12]: they selected speeds $v_1, \ldots, v_6$ and constructed a schedule for 6 agents that achieves an idle time of $\frac{41}{42}\left(2\ell / \sum_{i=1}^{k} v_i\right)$.

A patrolling algorithm $\mathcal{A}$ is *universal* if it can be executed with any number of agents k and any speed setting $v_1, \ldots, v_k > 0$ for the agents. For example, $\mathcal{A}_1$ above is universal, however certain algorithms (e.g., algorithm $\mathcal{A}_3$ in Section 4.3 or the algorithm in Section 4.4) can only be executed with certain speed settings or number of agents, i.e., they are not universal.

For the closed fence (circle), no universal algorithm has been proposed to be optimal. For uniform speeds (i.e., $v_1 = \ldots = v_k = v$), it is not difficult to see that placing the agents uniformly around the circle and letting them move in the same direction yields the shortest

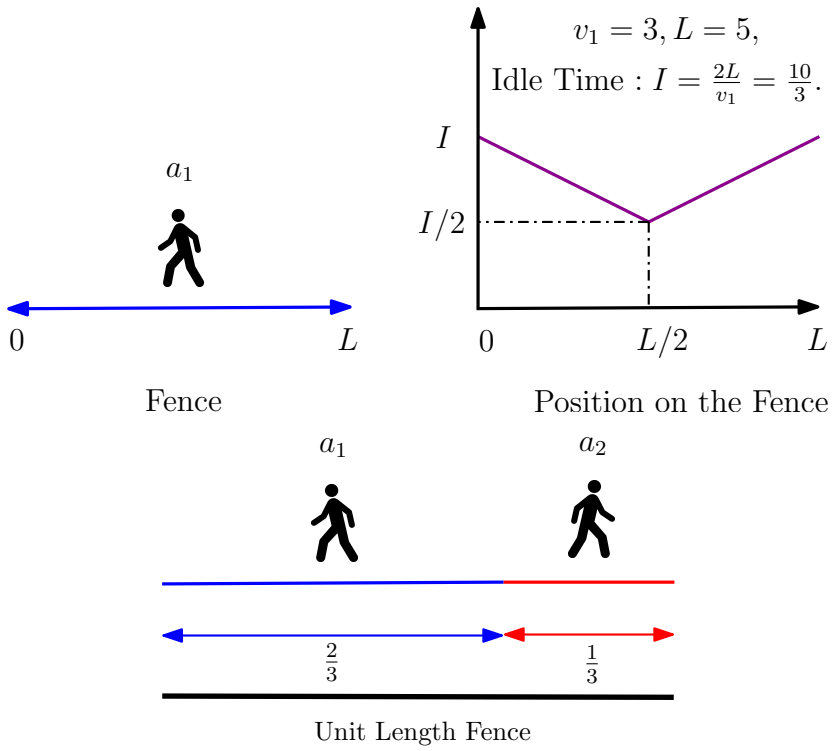

Figure 4.1: Top: illustration of the algorithm $\mathcal{A}_1$ with one agent and fence length L. Bottom: patrolling an unit length fence with two agents. Here, $v_1 = 2, v_2 = 1, I = 2/3$.

idle time. Indeed, the idle time in this case is $\ell/(kv) = \ell/\sum_{i=1}^{k} v_i$, matching the lower bound mentioned earlier.

For the variant in which all agents are *required* to move in the same direction along a circle of unit length (say clockwise), Czyzowicz et al. [7, Conjecture 2] conjectured that the following algorithm $\mathcal{A}_2$ always yields an optimal schedule.

Algorithm $\mathcal{A}_2$. Label the agents so that $v_1 \geq v_2 \geq \ldots \geq v_k > 0$. Let r, $1 \leq r \leq k$, be an index such that $\max_{1 \leq i \leq k} iv_i = rv_r$. Place the agents at equal distances of $1/r$ around the circle, so that each moves clockwise at the same speed v_r. Discard the remaining agents, if any. Since all agents move in the same direction, we also refer to $\mathcal{A}_2$ as the "runners" algorithm. It achieves an idle time of $1/\max_{1 \leq i \leq k} iv_i$ [7, Theorem 2]. Observe that $\mathcal{A}_2$ is also universal. Refer to Fig. 4.2 for an illustration with three agents.

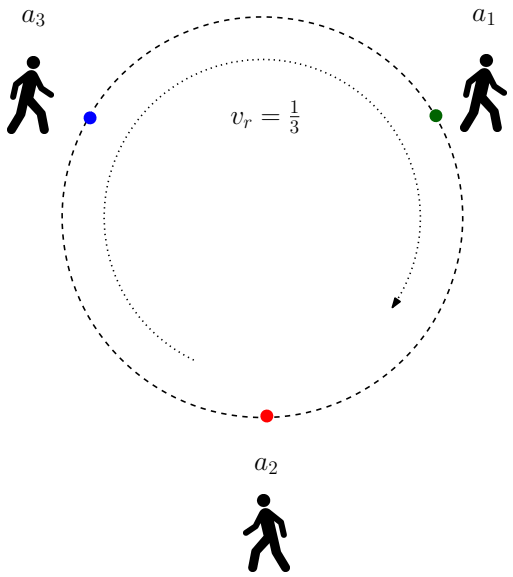

Figure 4.2: Demonstration of $\mathcal{A}_2$ using 3 equidistant agents moving clockwise : $v_1 = 1, v_2 = \frac{1}{2}, v_3 = \frac{1}{3}$, $v_r = \frac{1}{3}$, $I = \frac{1}{max_{1 \leq i \leq k} i v_i} = \frac{1}{max\left\{1 \cdot 1,\ 2 \cdot \frac{1}{2},\ 3 \cdot \frac{1}{3}\right\}} = 1$.

Related problems. Multi-agent patrolling is a variation of the problem of multi-robot coverage [4, 5], studied extensively in the robotics community. A variety of models has been considered for patrolling, including deterministic and randomized, as well as centralized and distributed strategies, under various objectives [1, 10]. *Idleness*, as a measure of efficiency for a patrolling strategy, was introduced by Machado et al. [13] in a graph setting; see also the article by Chevaleyre [4].

The closed fence patrolling problem is reminiscent of the classical *lonely runners conjecture*, introduced by Wills [14] and Cusick [6], independently, in number theory and discrete geometry. Assume that k agents run clockwise along a circle of length 1, starting from the same point at time $t = 0$. They have distinct but constant speeds (the speeds cannot vary, unlike in the model considered in this chapter). A runner is called *lonely* when he/she is at distance of at least $\frac{1}{k}$ from any other runner (along the circle). The conjecture asserts that each runner a_i is lonely at some time $t_i \in (0, \infty)$. The conjecture has only been confirmed for up to $k = 7$ runners [2, 3]. A recent survey [8] lists a few other related problems.

Notation and terminology. A *unit* circle is a circle of unit length. We parameterize a line segment and a circle of length ℓ by the interval $[0, \ell]$. A *schedule* of k agents consists of k functions $f_i : [0, \infty] \to [0, \ell]$, for $i = 1, \ldots, k$, where $f_i(t)$ is the position of agent i at time t. Each function f_i is continuous (for a closed fence, the endpoints of the interval $[0, \ell]$ are identified), it is piecewise differentiable, and its derivative (speed) is bounded by $|f_i'| \leq v_i$.

A schedule is called *periodic* with period $T > 0$ if $f_i(t) = f_i(t + T)$ for all $i = 1, \ldots, k$ and $t \geq 0$. The *idle time* I of a schedule is the maximum length of an open time interval (t_1, t_2) such that there is a point $x \in [0, \ell]$ where $f_i(t) \neq x$ for all $i = 1, \ldots, k$ and $t \in (t_1, t_2)$. Given a fence length ℓ, a fence type (closed or open), and maximum speeds $v_1, \ldots, v_k$, $\texttt{idle}(\mathcal{A})$ denotes the idle time of a schedule produced by algorithm $\mathcal{A}$ for these parameters.

We use *position-time diagrams* to plot the agent trajectories with respect to time. One axis represents the position $f_i(t)$ of the agents along the fence and the other axis represents time. In Fig. 4.3, for instance, the horizontal axis represents the position of the agents along the fence and the vertical axis represents time.

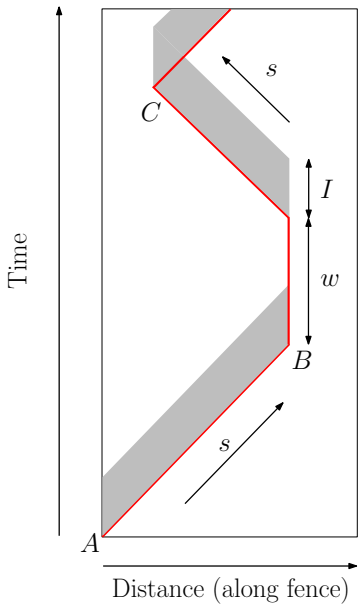

Figure 4.3: Agent moving with speed s from A to B, waiting at B for time w and then moving from B to C with speed s.

A schedule with idle time I is equivalent to a covering problem in such a diagram (see

Fig. 4.3). For a straight-line (i.e., constant speed) trajectory between points (x_1, y_1) and (x_2, y_2) in the diagram, construct a shaded parallelogram with vertices, (x_1, y_1), $(x_1, y_1 + I)$, (x_2, y_2), $(x_2, y_2 + I)$, where I denotes the desired idle time and the shaded region represents the covered region. In particular, if an agent stays put in a time-interval, the parallelogram degenerates to a vertical segment. A schedule for the agents ensures idle time I if and only if the entire area of the diagram in the time interval $[I, \infty)$ is covered.

The efficiency of a patrolling algorithm $\mathcal{A}$ is measured by the ratio

$$\varrho = \frac{\texttt{idle}(\mathcal{A})}{\texttt{idle}(\mathcal{A}_1)}$$

between the idle times of $\mathcal{A}$ and the partition-based algorithm $\mathcal{A}_1$. Lower values of ϱ indicate better (more efficient) algorithms. Recall however that certain algorithms can only be executed with certain speed settings or number of agents.

Our results.

1. Consider the unidirectional unit circle (where all agents are required to move in the same direction).

   (i) We disprove a conjecture by Czyzowicz et al. [7, Conjecture 2] regarding the optimality of algorithm $\mathcal{A}_2$. Specifically, we construct a schedule for 32 agents with harmonic speeds $v_i = 1/i$, $i = 1, \ldots, 32$, that has an idle time strictly less than 1. In contrast, algorithm $\mathcal{A}_2$ yields a unit idle time for harmonic speeds ($\texttt{idle}(\mathcal{A}_2) = 1$), hence it is suboptimal. See Theorem 4.1, Section 4.2.

   (ii) For every $\tau \in (0, 1]$ and $t \geq \tau$, there exists a positive integer $k = k(t) \leq e^{4t/\tau^2}$ and a schedule for the system of k agents with harmonic speeds $v_i = 1/i$, $i = 1, \ldots, k$, that ensures an idle time at most τ during the time interval $[0, t]$. See Theorem 4.2, Section 4.2.

2. Consider the open fence patrolling. For every integer $x \geq 2$, there exist $k = 4x + 1$ agents with $\sum_{i=1}^{k} v_i = 16x + 1$ and a guarding schedule for a segment of length $25x/3$. Alterna-

tively, for every integer $x \geq 2$ there exist $k = 4x+1$ agents with suitable speeds $v_1, \ldots, v_k$, and a guarding schedule for a unit segment that achieves idle time at most $\frac{48x+3}{50x} \frac{2}{\sum_{i=1}^{k} v_i}$. In particular, for every $\varepsilon > 0$, there exist k agents with suitable speeds $v_1, \ldots, v_k$, and a guarding schedule for a unit segment that achieves idle time at most $\left(\frac{24}{25} + \varepsilon\right) \frac{2}{\sum_{i=1}^{k} v_i}$. This improves the previous bound of $\frac{41}{42} \frac{2}{\sum_{i=1}^{k} v_i}$ by Kawamura and Kobayashi [12]. See Theorem 4.3, Section 4.4.

3. Consider the bidirectional unit circle.

   (i) For every $k \geq 4$, there exist maximum speeds $v_1, \ldots, v_k > 0$ and a patrolling algorithm $\mathcal{A}_3$ with a shorter idle time than that achieved by both $\mathcal{A}_1$ and $\mathcal{A}_2$. In particular, for large k, the idle time of $\mathcal{A}_3$ with these speeds is about $2/3$ of that achieved by $\mathcal{A}_1$ and $\mathcal{A}_2$. See Proposition 4.1, Section 4.3.

   (ii) For every $k \geq 2$, there exist maximum speeds $v_1, \ldots, v_k > 0$ and an optimal schedule for patrolling the circle that does not use up to $k-1$ of the agents $a_2, \ldots, a_k$. In contrast, for a segment, any optimal schedule must use all agents. See Proposition 4.2, Section 4.3.

   (iii) There exist settings in which if all k agents are used by a patrolling algorithm, then some agent(s) need overtake (pass) other agent(s). This partially answers a question left open by Czyzowicz et al. [7, Section 3]. See the remark at the end of Section 4.3.

Note. For the case open fence patrolling (refer to Result 2), Kawamura and Soejima [11] subsequently improved the ratio from $24/25$ to $3/4$.

General observations. 1. *Strategy scalability.* Suppose we have a patrolling strategy with k agents for a fence (open or closed) of length l with ratio ϱ (relative to the partition strategy). Then, we can *scale* this strategy for every $l' \neq l$ using k agents as follows. Let $l'/l = c$, then $v'_i = cv_i, 1 \leq i \leq k$, where v'_i is the scaled speed of a_i. The waiting times used in the strategy at specific positions for agents need not to be scaled. One can check that the ratio ϱ remains unchanged.

2. *Strategy extension.* Suppose we have a patrolling strategy with k agents for a fence (open or closed) of length l with ratio $\varrho > 1$ (relative to the partition strategy). Then for any $k' > k$, there exists a a patrolling strategy with k' agents for a fence of length $l' > l$ with ratio $\varrho' > 1$: use $m = k' - k$ additional agents with $\sum_{i=k+1}^{k'} v_i = 2(l' - l)$ to patrol $l' - l$ using the partition strategy. Now if $\varrho = \frac{a}{b} > 1$, then one can check that $\varrho' = \frac{a+2(l'-l)}{b+2(l'-l)} > 1$. It follows from the results of Kawamura and Kobayashi [12] and the above observation that the partition based algorithm is not optimal for a segment for any $k \geq 6$, and k suitable speeds.

4.2 Unidirectional circle patrolling[1]

A counterexample for the optimality of algorithm $\mathcal{A}_2$. We show that algorithm $\mathcal{A}_2$ by Czyzowicz et al. [7] for unidirectional circle patrolling is not always optimal. We consider agents with *harmonic speeds* $v_i = 1/i$, $i \in \mathbf{N}$. Obviously, for this setting we have $\texttt{idle}(\mathcal{A}_2) = 1$, which is already achieved by the agent a_1 with the highest (here unit) speed. We design a periodic schedule (patrolling algorithm) for $k = 32$ agents with idle time $I < 1$. In this schedule, agent a_1 moves continuously with unit speed, and it remains to schedule agents $a_2, \ldots, a_{32}$ such that every point is visited at least one more time in the unit length open time interval between two consecutive visits of a_1. We start with a weaker claim shown in Lemma 4.1, for *closed* intervals but using only 6 agents. Then, using the lemma we arrive at our main result in Theorem 4.1.

Lemma 4.1. *Consider the unit circle, where all agents are required to move in the same direction. For $k = 6$ agents of harmonic speeds $v_i = 1/i$, $i = 1, \ldots, 6$, there is a schedule where agent a_1 moves continuously with speed 1, and every point on the circle is visited by some other agent in every closed unit time interval between two consecutive visits of a_1.*

Theorem 4.1. *Consider the unit circle, where all agents are required to move in the same direction. For 32 agents of harmonic speeds $v_i = 1/i$, $i = 1, \ldots, 32$, there is a periodic schedule with idle time strictly less than 1.*

[1] Theorem 4.1 is mainly the work of Cs. D. Tóth, refer to [9].

Finite time patrolling. Interestingly enough, we can achieve any prescribed idle time below 1 for an arbitrarily long time in this setting, provided we choose the number of agents k large enough.

Theorem 4.2. *Consider the unit circle, where all agents are required to move in the same direction. For every $0 < \tau \leq 1$ and $t \geq \tau$, there exists $k = k(t) \leq e^{4t/\tau^2}$ and a schedule for the system of k agents with maximum speeds $v_i = 1/i$, $i = 1, \ldots, k$, that ensures an idle time at most τ during the time interval $[0, t]$.*

Proof. We construct a schedule with an idle time at most τ. Let agent a_1 start at time 0 and move clockwise at maximum (unit) speed, i.e., $f_1(t) = t \mod 1$ denotes the position on the unit circle of agent a_1 at time t. Assume without loss of generality that t is a multiple of τ, i.e., $t = m\tau$, where m is a natural number. Divide the time interval $[0, t]$ into $2m$ subintervals of length $\tau/2$. For $j = 1, \ldots, 2m$, $[(j-1)\tau/2, j\tau/2]$ is the jth interval.

For each j, cover the unit circle C so that every point of C is visited at least once by some agent. This ensures that each point of the circle is visited at least once in the time interval $[0, \tau/2]$ and no two consecutive visits to any one point are separated in time by more than τ thereafter until time t, as required.

To achieve the covering condition in each interval j, we use the first agent (a_1, of unit speed), and as many other unused agents as needed. The 'origin' on C is reset to the current position of a_1 at time $(j-1)\tau/2$, i.e., the beginning of the current time interval. So the fastest agent is used (continuously) in all $2m$ time intervals. Agent a_1 can cover a distance of $\tau/2$ during one interval. From its endpoint, at time $(j-1)\tau/2$, start the unused agent with the smallest index, say $i_1(j)$; this agent can cover a distance of $\frac{\tau}{2}\frac{1}{i_1(j)}$ during the interval. Continue in the same way using new agents, all starting at time $(j-1)\tau/2$, until the entire circle C is covered; let the index of the last agent used be $i_2(j)$. The covering condition can be written as:

$$\frac{\tau}{2}\left(1 + \sum_{i=i_1(j)}^{i_2(j)} \frac{1}{i}\right) \geq 1, \text{ or equivalently, } 1 + \sum_{i=i_1(j)}^{i_2(j)} \frac{1}{i} \geq \frac{2}{\tau}. \tag{4.1}$$

For example[2], if $\tau = 2/3$: $j = 1$ requires agents a_1 through a_{11}, since $H_{11} \geq 3$, but $H_{10} < 3$; $j = 2$ requires agents a_1 and agents a_{12} through a_{85}, since $1+(H_{85}-H_{11}) \geq 3$, but $1+(H_{84}-H_{11}) < 3$.

We now bound from above the total number k of distinct agents used. Observe that the covering condition (4.1) may lead to overshooting the target. Because the harmonic series has decreasing terms, the overshooting error cannot exceed the term $\frac{1}{i_2(1)+1}$ for $\tau = 1$, namely $1/5$ (the overshooting for $\tau = 1$ is only $\frac{1}{3} - \frac{1}{4} = \frac{1}{12} < \frac{1}{5}$). So inequality (4.1) becomes

$$\frac{2}{\tau} \leq 1 + \sum_{i=i_1(j)}^{i_2(j)} \frac{1}{i} \leq \frac{2}{\tau} + \frac{1}{5}. \tag{4.2}$$

Recall that $t = m\tau$. By adding inequality (4.2) over all $2m$ time intervals yields (in equivalent forms)

$$H_k - 1 + \frac{8m}{5} \leq \frac{4m}{\tau}, \text{ or } H_k \leq \frac{4t}{\tau^2} + 1 - \frac{8t}{5\tau}. \tag{4.3}$$

For $t \geq \tau$ we have $1 \leq \frac{8t}{5\tau}$. Since $\ln k \leq H_k$, it follows from (4.3) that

$$\ln k \leq \frac{4t}{\tau^2}, \text{ or } k \leq e^{4t/\tau^2},$$

as required. □

4.3 Bidirectional circle patrolling

The "train" algorithm for closed fence patrolling. Czyzowicz et al. [7, Theorem 5] showed that for $k = 3$ there exist maximum speeds v_1, v_2, v_3 and a schedule that achieves a shorter idle time than both algorithm $\mathcal{A}_1$ and $\mathcal{A}_2$, namely $35/36$ versus $12/11$ and 1. We extend their result for all $k \geq 4$. We propose a new patrolling algorithm, $\mathcal{A}_3$, for maximum speeds $v_1 > v_2 \geq \ldots \geq v_k > 0$. That is, $\mathcal{A}_3$ assumes that one of the agents is faster than all others; we then show that for all $k \geq 4$ there exist k maximum speeds for which $\mathcal{A}_3$ outperforms both $\mathcal{A}_1$ and $\mathcal{A}_2$.

Place the $k-1$ agents $a_2, \ldots, a_k$ at equal distances, x on the unit circle, and let them move all

[2] $H_n = \sum_{i=1}^{n} 1/i$ denotes the nth *harmonic number*.

clockwise perpetually at the same speed v_k; we say that $a_2, \ldots, a_k$ make a "train". Let a_1 move back and forth (i.e., clockwise and counterclockwise) perpetually on the moving arc of length $1-(k-2)x$, i.e., between the start and the end of the train. Refer to Fig. 4.4.

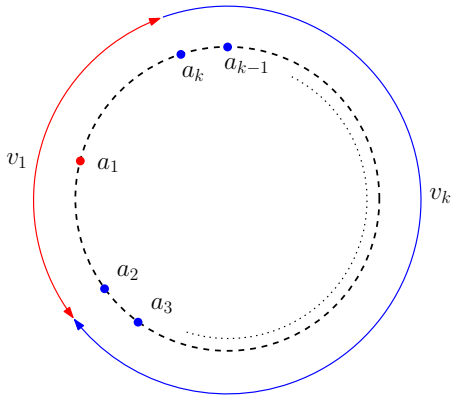

Figure 4.4: Train algorithm: the train $a_2, \ldots, a_k$ moving unidirectionally with speed v_k and the bidirectional agent a_1 with speed v_1.

Proposition 4.1. *For every $k \geq 4$, there exist maximum speeds $v_1 > v_2 \geq \ldots \geq v_k$ such that algorithm $\mathcal{A}_3$ achieves a shorter idle time than both $\mathcal{A}_1$ and $\mathcal{A}_2$. In particular, for large k, the idle time achieved by the train algorithm is about $2/3$ of the idle times achieved by $\mathcal{A}_1$ and $\mathcal{A}_2$.*

Proof. Consider the speed setting $v_1 = a$, $v_2 = \ldots = v_k = b$, where $a > b > 0$, and $\max_{1 \leq i \leq k} iv_i = kb$ (i.e., $a \leq kb$). Put $y = 1-(k-2)x$. To determine the idle time, x/b, write:

$$[1-(k-2)x]\left(\frac{1}{a-b}+\frac{1}{a+b}\right) = \frac{x}{b}, \text{ or equivalently, } \frac{2ay}{a^2-b^2} = \frac{1-y}{(k-2)b}.$$

Solving for x/b yields

$$\texttt{idle}(\mathcal{A}_3) = \frac{2a}{a^2-b^2+2(k-2)ab}.$$

For our speed setting, we also have

$$\texttt{idle}(\mathcal{A}_1) = \frac{2}{a+(k-1)b}, \text{ and } \texttt{idle}(\mathcal{A}_2) = \frac{1}{kb}.$$

Write $t = a/b$. It can be checked that for $k \geq 4$, $\texttt{idle}(\mathcal{A}_3) \leq \texttt{idle}(\mathcal{A}_1)$ and $\texttt{idle}(\mathcal{A}_3) \leq \texttt{idle}(\mathcal{A}_2)$ when $a^2 - b^2 - 4ab \geq 0$, i.e., $t \geq 2 + \sqrt{5}$. In particular, for $a = 1$, and $b = 1/k$ (note that $a \leq kb$), we have

$$\texttt{idle}(\mathcal{A}_3) = \frac{2}{1 - 1/k^2 + 2(k-2)/k} \underset{k\to\infty}{\longrightarrow} \frac{2}{3},$$

while

$$\texttt{idle}(\mathcal{A}_1) = \frac{2}{1 + (k-1)/k} \underset{k\to\infty}{\longrightarrow} 1, \text{ and } \texttt{idle}(\mathcal{A}_2) = \frac{1}{k(1/k)} = 1. \qquad \square$$

Useless agents for circle patrolling. Czyzowicz et al. [7] showed that for $k = 2$ there are maximum speeds for which an optimal schedule does not use one of the agents. Here we extend this result for all $k \geq 2$:

Proposition 4.2. (i) *For every $k \geq 2$, there exist maximum speeds $v_1, \ldots, v_k > 0$ and an optimal schedule for patrolling the circle with these speeds that does not use up to $k-1$ of the agents $a_2, \ldots, a_k$.* (ii) *In contrast, for a segment, any optimal schedule must use all agents.*

Proof. (i) Let $v_1 = 1$ and $v_2 = \ldots = v_k = \varepsilon/k$, for a small positive $\varepsilon \leq 1/300$, and C be a unit circle. Obviously by using agent a_1 alone (moving perpetually clockwise) we can achieve unit idle time. Assume for contradiction that there exists a schedule achieving an idle time less than 1. Let $f_1(t)$ denote the position of agent a_1 at time t. Assume without loss of generality that $f_1(0) = 0$ and consider the time interval $[0, 2]$. For $2 \leq i \leq k$, let J_i be the interval of points visited by agent a_i during the time interval $[0, 2]$, and put $J = \cup_{i=2}^{k} J_i$. We have $|J_i| \leq 2\varepsilon/k$, thus $|J| \leq 2\varepsilon$. We make the following observations:

1. $f_1(1) \in [-2\varepsilon, 2\varepsilon]$. Indeed, if $f_1(1) \notin [-2\varepsilon, 2\varepsilon]$, then either some point in $[-2\varepsilon, 2\varepsilon]$ is not visited by any agent during the time interval $[0, 1]$, or some point in $C \setminus [-2\varepsilon, 2\varepsilon]$ is not visited by any agent during the time interval $[0, 1]$.

2. a_1 has done almost a complete (say, clockwise) rotation along C during the time interval

$[0,1]$, i.e., it starts at $0 \in [-2\varepsilon, 2\varepsilon]$ and ends in $[-2\varepsilon, 2\varepsilon]$, otherwise some point in $C \setminus [-2\varepsilon, 2\varepsilon]$ is not visited during the time interval $[0,1]$.

3. $f_1(2) \in [-4\varepsilon, 4\varepsilon]$, by a similar argument.

4. a_1 has done almost a complete rotation along C during the time interval $[1,2]$, i.e., it starts in $[-2\varepsilon, 2\varepsilon]$ and ends in $[-4\varepsilon, 4\varepsilon]$. Moreover this rotation must be in the same clockwise sense as the previous one, since otherwise there would exist points not visited for at least one unit of time.

Pick three points $x_1, x_2, x_3 \in C \setminus J$ close to $1/4$, $2/4$, and $3/4$, respectively, i.e., $|x_i - i/4| \leq 1/100$, for $i = 1,2,3$. By Observations 2 and 4, these three points must be visited by a_1 in the first two rotations during the time interval $[0,2]$ in the order $x_1, x_2, x_3, x_1, x_2, x_3$. Since a_1 has unit speed, successive visits to x_1 are separated in time by at least one time unit, contradicting the assumption that the idle time of the schedule is less than 1.

(ii) Given $v_1 \geq v_2 \geq \ldots \geq v_k > 0$, assume for contradiction that there is an optimal guarding schedule with unit idle time for a segment s of maximum length that does not use agent a_j (with maximum speed v_j), for some $1 \leq j \leq k$. Extend s at one end by a subsegment of length $v_j/2$ and assign a_j to this subsegment to move back and forth from one end to the other, perpetually. We now have a guarding schedule with unit idle time for a segment longer than s, which is a contradiction. □

Overtaking other agents. Consider an optimal schedule for circle patrolling (with unit idle time) for the agents in the proof of Proposition 4.2, with $v_1 = 1$ and $v_2 = \ldots = v_k = \varepsilon/k$, in which all agents move clockwise at their maximum speeds. Obviously a_1 will overtake all other agents during the time interval $[0,2]$. Thus there exist settings in which if all k agents are used by a patrolling algorithm, then some agent(s) need to overtake (pass) other agent(s). Observe however that overtaking can be easily avoided in this setting by not making use of any of the agents $a_2, \ldots, a_k$.

4.4 An improved idle time for open fence patrolling

Kawamura and Kobayashi [12] showed that algorithm $\mathcal{A}_1$ by Czyzowicz et al. [7] does not always produce an optimal schedule for open fence patrolling. They presented two counterexamples: their first example uses 6 agents and achieves an idle time of $\frac{41}{42}\,\texttt{idle}(\mathcal{A}_1)$; their second example uses 9 agents and achieves an idle time of $\frac{99}{100}\,\texttt{idle}(\mathcal{A}_1)$. By replicating the strategy from the second example with a number of agents larger than 9, i.e., iteratively using blocks of agents, we improve the ratio to $24/25+\varepsilon$ for any $\varepsilon > 0$. We need two technical lemmas to verify this claim.

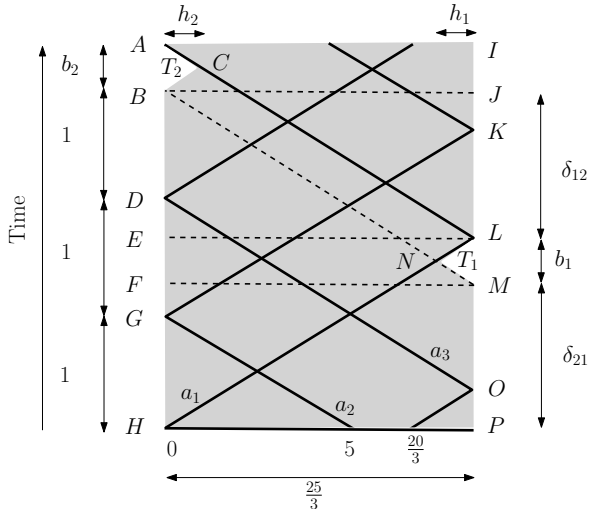

Figure 4.5: Three agents each with a speed of 5 patrolling a fence of length $25/3$; their start positions are 0, 5, and $20/3$, respectively. Figure is *not* to scale.

Lemma 4.2. *Consider a segment of length $\ell = \frac{25}{3}$ such that three agents a_1, a_2, a_3 are patrolling perpetually each with speed of 5 and generating an alternating sequence of uncovered triangles $T_2, T_1, T_2, T_1, \ldots$, as shown in the position-time diagram in Fig. 4.5. Denote the vertical distances between consecutive occurrences of T_1 and T_2 by δ_{12} and between consecutive occurrences of T_2 and T_1 by δ_{21}. Denote the bases of T_1 and T_2 by b_1 and b_2 respectively, and the heights of T_1 and T_2 by h_1 and h_2 respectively. Then*

(i) $\frac{10}{3}$ *is a period of the schedule.*

(ii) *T_1 and T_2 are congruent; further, $b_1 = b_2 = \frac{1}{3}$, $\delta_{12} = \delta_{21} = \frac{4}{3}$, and $h_1 = h_2 = \frac{5}{6}$.*

Proof. (i) Observe that a_1, a_2 and a_3 reach the left endpoint of the segment at times $2(25/3)/5 = 10/3$, $5/5 = 1$, and $(25/3 + 5/3)/5 = 2$, respectively. During the time interval $[0, 10/3]$, each agent traverses the distance 2ℓ and the positions and directions of the agents at time $t = 10/3$ are the same as those at time $t = 0$. Hence $10/3$ is a period for their schedule.

(ii) Since $AL \parallel BM$ and $AB \parallel LM$, we have $b_1 = b_2$. Since L is the midpoint of IP, we have $\delta_{12} + b_2 = \delta_{21} + b_1$, thus $\delta_{12} = \delta_{21}$. Since all the agents have same speed, 5, all the trajectory line segments in the position-time diagram have the same slope, $1/5$. Hence $\angle BAC = \angle ABC = \angle MLN = \angle LMN$. Thus, T_1 is similar to T_2. Since $b_1 = b_2$, T_1 is congruent to T_2, and consequently $h_1 = h_2$.

Put $b = b_1$, $h = h_1$, and $\delta = \delta_{12}$. Recall from (i) that $|AH| = 10/3$. By construction, we have $|BD| = 1$, thus $|BH| = |BD| + |DG| + |GH| = 1 + 1 + 1 = 3$. We also have $|AH| = b + |BH|$, thus $b = 10/3 - 3 = 1/3$. Since L is the midpoint of IP, we have $\delta + b = 5/3$, thus $\delta = 5/3 - b = 4/3$.

Let $x(N)$ denote the x-coordinate of point N; then $x(N) + h = 25/3$. To compute $x(N)$ we compute the intersection of the two segments HL and BM. We have $H = (0, 0)$, $L = (25/3, 5/3)$, $B = (0, 3)$, and $M = (25/3, 4/3)$. The equations of HL and BM are $HL : x = 5y$ and $BM : x + 5y = 15$, and solving for x yields $x = 15/2$, and consequently $h = 25/3 - 15/2 = 5/6$. □

Lemma 4.3. (i) *Let s_1 be the speed of an agent needed to cover an uncovered isosceles triangle T_i; refer to Fig. 4.6 (left). Then $s_1 = \frac{h}{1-b/2}$, where $b < 1$ and h are the base and height of T_i, respectively.*

(ii) *Let s_2 be the speed of an agent needed to cover an alternate sequence of congruent isosceles triangles T_1, T_2 with bases on same vertical line; refer to Fig. 4.6 (right). Then $s_2 = \frac{h}{3b/2+y-1}$ where y is the vertical distance between the triangles, $b < 1$ is the base and h is the height of the congruent triangles.*

Proof. (i) In Fig. 4.6 (left), $\tan\alpha = 1/s_1$, $|UZ| = b/2$, hence $|VZ| = 1 - b/2$. Also, $\frac{|VZ|}{|WV|} =$

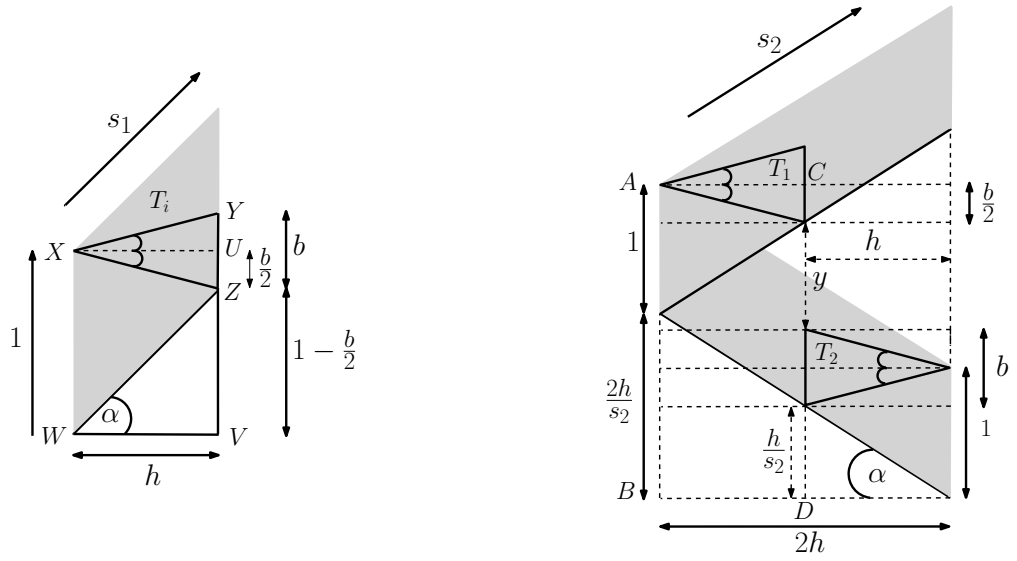

Figure 4.6: Left: agent covering an uncovered triangle T_i. Right: agent covering an alternate sequence of congruent triangles T_1, T_2, with collinear bases.

$\tan\alpha = \frac{1-b/2}{h} = \frac{1}{s_1}$, which yields $s_1 = \frac{h}{1-b/2}$.

(ii) In Fig. 4.6 (right), $|AB| = 1+\frac{2h}{s_2}$. Also, $|CD| = \frac{b}{2}+y+b+\frac{h}{s_2}$. Equating $1+\frac{2h}{s_2} = \frac{3b}{2}+y+\frac{h}{s_2}$ and solving for s_2, we get $s_2 = \frac{h}{3b/2+y-1}$. □

Theorem 4.3. *For every integer $x \geq 2$, there exist $k = 4x+1$ agents with $\sum_{i=1}^{k} v_i = 16x+1$ and a guarding schedule for a segment of length $25x/3$. Alternatively, for every integer $x \geq 2$ there exist $k = 4x+1$ agents with suitable speeds $v_1, \ldots, v_k$, and a guarding schedule for a unit segment that achieves idle time at most $\frac{48x+3}{50x}\,\frac{2}{\sum_{i=1}^{k} v_i}$. In particular, for every $\varepsilon > 0$, there exist k agents with suitable speeds $v_1, \ldots, v_k$, and a guarding schedule for a unit segment that achieves idle time at most $\left(\frac{24}{25}+\varepsilon\right)\,\frac{2}{\sum_{i=1}^{k} v_i}$.*

Proof. Refer to Fig. 4.7. We use a long fence divided into x blocks; each block is of length $25/3$. Each block has 3 agents each of speed 5 running in zig-zag fashion. Consecutive blocks share one agent of speed 1 which covers the uncovered triangles from the trajectories of the zig-zag agents in the position-time diagram. The first and the last block use two agents of speed 1 not shared by any other block. The setting of these speeds is explained below.

From Lemma 4.2(ii), we conclude that all the uncovered triangles generated by the agents of

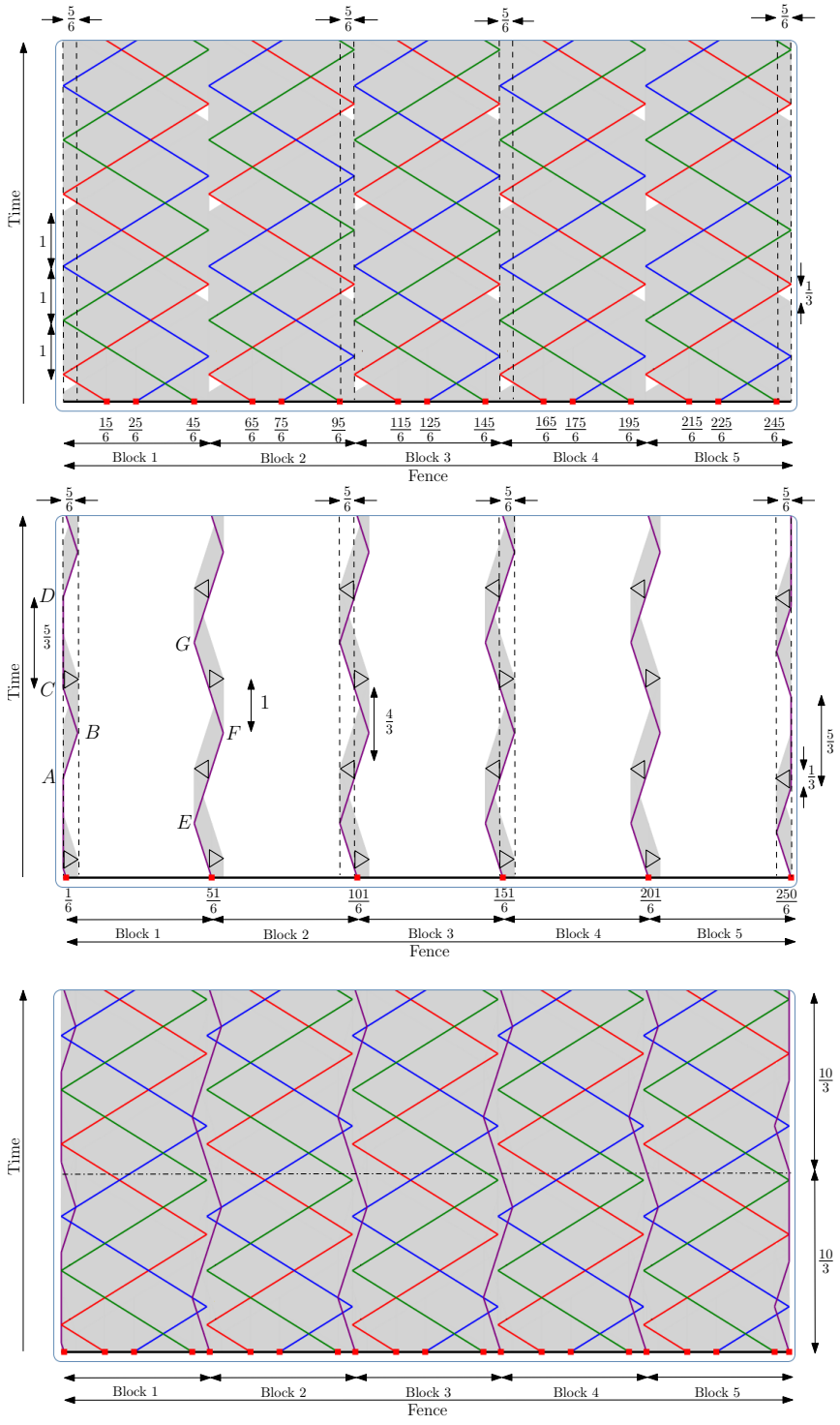

Figure 4.7: Top: iterative construction with 5 blocks; each block has three agents with speed 5. Middle: 6 agents with speed 1. Bottom: patrolling strategy for 5 blocks using 21 agents for two time periods (starting at $t = 1/3$ relative to Fig. 4.5); the block length is $25/3$ and the period is $10/3$.

speed 5 are congruent and their base is $b = 1/3$ and their height is $h = 5/6$. By Lemma 4.3(i), we can set the speeds of the agents not shared by consecutive blocks to $s_1 = \frac{5/6}{1-1/6} = 1$. Also, in our strategy, Lemma 4.2(ii) yields $y = \delta = 4/3$. Hence, by Lemma 4.3(ii), we can set the speeds of the agents shared by consecutive blocks to $s_2 = \frac{5/6}{1/2+4/3-1} = 1$.

In our strategy, we have 3 types of agents: agents running with speed 5 as in Fig. 4.7 (top), unit speed agents not shared by 2 consecutive blocks and unit speed agents shared by two consecutive blocks as in Fig. 4.7 (middle). By Lemma 4.2(i), the agents of first type have period $10/3$. In Fig. 4.7 (middle), there are two agents of second type and both have a similar trajectory. Thus, it is enough to verify for the leftmost unit speed agent. It takes $5/6$ time from A to B and again $5/6$ time from B to C. Next, it waits for $5/3$ time at C. Hence after $5/6 + 5/6 + 5/3 = 10/3$ time, its position and direction at D is same as that at A. Hence, its time period is $10/3$. For the agents of third type, refer to Fig. 4.7 (middle): it takes $10/6$ time from E to F and $10/6$ time from F to G. Thus, arguing as above, its time period is $10/3$. Hence, overall, the time period of the strategy is $10/3$.

For x blocks, we use $3x+(x+1) = 4x+1$ agents. The sum of all speeds is $5(3x)+1(x+1) = 16x + 1$ and the total fence length is $\frac{25x}{3}$. The resulting ratio is $\varrho = \frac{16x+1}{2}/\frac{25x}{3} = \frac{48x+3}{50x}$. For example, when $x = 2$ we reobtain the bound of Kawamura and Kobayashi [12] (from their 2nd example), when $x = 39, \varrho = \frac{100}{104}$ and further on, $\varrho \underset{x\to\infty}{\longrightarrow} \frac{24}{25}$. Thus an idle time of at most $\left(\frac{24}{25} + \varepsilon\right) \frac{2}{\sum_{i=1}^{k} v_i}$ can be achieved for every given $\varepsilon > 0$, as required. □

4.5 Framework for deriving a tighter bound for ϱ

In this section, we consider the patrolling schedules for open fences. We present an idea how to possibly derive a tighter non trivial lower bound for ϱ. Let $\eta = 1/\varrho$. Recall that by a volumetric argument it can be shown that $\eta \leq 2$ for open fence patrolling. We believe that $\eta < 2$, or equivalently there is no strategy which is twice as efficient as the partition-based strategy for open fences when unit idle time is considered. The following discussion establishes a possible

framework for achieving a tighter concrete upper bound for η, less than 2. Here, we focus our attention to the periodic schedules only. However, we think that our argument can be extended to any arbitrary schedule.

Let $v_1, \ldots, v_k$ be the maximum speeds of the k agents. Fix a periodic schedule S for the k agents with time period T, that patrols an open fence of length L. Our approach is based on a covering argument in the position-time diagrams. For S, consider the rectangular region R in the diagram with one side having length L, parallel to the fence and the other having length T, parallel to the time axis. Obviously, $\texttt{area}(R) = LT$. Now we consider the trajectories of the agents in one time period T. Observe that in the time interval T, an agent with maximum speed v_i can generate a shaded region of maximum area Tv_i inside R. Thus,

$$LT \leq T \sum_{i=1}^{k} v_i - \alpha, \tag{4.4}$$

where α is the total area of the overlapping shaded regions contributed by the k agents. Overlapping of shaded regions can occur when an agent takes turn or at least two different agents meet. Using (4.4), the following relation between ϱ and α can be obtained.

$$\eta = \frac{2L}{\sum_{i=1}^{k} v_i} = \frac{2LT}{T\sum_{i=1}^{k} v_i} \leq \frac{2(T\sum_{i=1}^{k} v_i - \alpha)}{T\sum_{i=1}^{k} v_i} = 2 - 2\frac{\alpha}{T\sum_{i=1}^{k} v_i} < 2. \tag{4.5}$$

If there is a positive constant c such that for any open fence patrolling schedule, $\frac{\alpha}{T\sum_{i=1}^{k} v_i} \geq c$, a tighter upper bound for η can be obtained, i.e., $\eta \leq 2(1-c)$. Also, it follows from (4.5) that η is inversely proportional to α. Simply put, an efficient patrolling schedule has low α. Furthermore, we note that in any schedule based on the partition-based strategy, any subsection of the rectangular region R is overlapping.

Note that if every agent has the same maximum speed, the partition-based strategy is optimal; see [12]. Assume that we have many agents of high speed and many agents of low speed. We believe that in any patrolling schedule there is always a certain positive percentage of overlap between the shaded regions of the high speed and low speed agents. In other words, if $\mathcal{O}$ be the

area of the said overlap, we believe that $\mathcal{O}/LT \geq p$ for some constant $0 < p < 1$. This may be useful to obtain a bound for α as discussed above. For instance, the aforesaid observation can be verified for the infinite family of strategies presented in [11]; refer to Fig. 4.8 for a member strategy. For more details, we refer the reader to their paper. The same observation can also be verified for the strategies presented in this chapter.

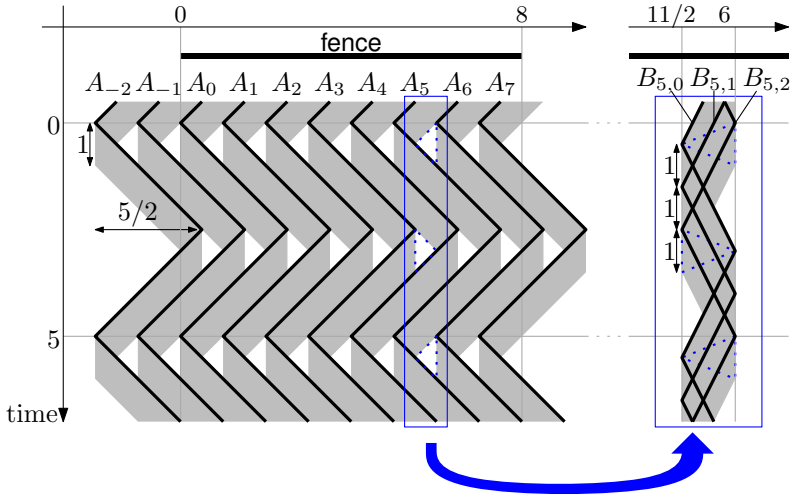

Figure 4.8: A patrolling strategy when fence length is 8. A_is are the unit speed (high speed) agents and B_{ij}s are the low speed agents each having maximum speed $1/5$. Total number of agents used is 34. Figure by Kawamura and Soejima, source: `arxiv.org/abs/1411.6853`.

Acknowledgements. We sincerely thank Akitoshi Kawamura for generously sharing some technical details concerning the patrolling schedules in [12]. We also express our satisfaction with the software package *JSXGraph, Dynamic Mathematics with JavaScript*, used in our experiments.

References

[1] N. Agmon, S. Kraus, and G. A. Kaminka, Multi-robot perimeter patrol in adversarial settings, in *Proc. Int. Conf. Robotics and Automation (ICRA 2008)*, IEEE, 2008, pp. 2339–2345.

[2] J. Barajas and O. Serra, The lonely runner with seven runners, *Electron. J. Combin.* **15** (2008), R48.

[3] T. Bohman, R. Holzman, and D. Kleitman, Six lonely runners, *Electron. J. Combin.* **8** (2001), R3.

[4] Y. Chevaleyre, Theoretical analysis of the multi-agent patrolling problem, in *Proc. Int. Conf. Intelligent Agent Technology (IAT 2004)*, IEEE, 2004, pp. 302–308.

[5] H. Choset, Coverage for robotics – a survey of recent results, *Ann. Math. Artificial Intelligence* **31** (2001), 113–126.

[6] T. W. Cusick, View-obstruction problems, *Aequationes Math.* **9** (1973), 165–170.

[7] J. Czyzowicz, L. Gasieniec, A. Kosowski, and E. Kranakis, Boundary patrolling by mobile agents with distinct maximal speeds, in *Proc. 19th European Sympos. Algorithms (ESA 2011)*, LNCS 6942, Springer, 2011, pp. 701–712.

[8] A. Dumitrescu and Cs. D. Tóth, Computational Geometry Column 59, *SIGACT News Bulletin* **45(2)** (2014), 68–72.

[9] A. Dumitrescu, A. Ghosh, and Cs. D. Tóth, On fence patrolling by mobile agents, *Electron. J. Combin.* **21(3)** (2014), P3.4.

[10] Y. Elmaliach, N. Agmon, and G. A. Kaminka, Multi-robot area patrol under frequency constraints, in *Proc. Int. Conf. Robotics and Automation (ICRA 2007)*, IEEE, 2007, pp. 385–390.

[11] A. Kawamura and M. Soejima, Simple strategies versus optimal schedules in multi-agent patrolling, in Proc. 9th International Conference on Algorithms and Complexity (CIAC 2015), LNCS 9079, Springer, 2015, pp. 261–273.

[12] A. Kawamura and Y. Kobayashi, Fence patrolling by mobile agents with distinct speeds, in *Proc. 23rd Int. Sympos. Algorithms and Computation (ISAAC 2012)*, LNCS 7676, Springer, 2012, pp. 598–608.

[13] A. Machado, G. Ramalho, J. D. Zucker, and A. Drogoul, Multi-agent patrolling: an empirical analysis of alternative architectures, in *Multi-Agent-Based Simulation II (J.S. Sichman et al., eds.)*, LNCS 2581, Springer, 2002, pp. 155–170.

[14] J. M. Wills, Zwei Sätze über inhomogene diophantische Approximation von Irrationalzahlen, *Monatsch. Math.* **71** (1967), 263–269.

Curriculum Vitae

Anirban Ghosh

Education

- **Fall 2011 – Spring 2016**: Doctor of Philosophy in Engineering, Department of Computer Science, University of Wisconsin-Milwaukee.

  Research Assistant: Fall 2011–Spring 2014,

  Teaching Assistant: Fall 2014–Spring 2016.

- **2011**: Master of Technology (M.Tech) in *Computer Science*, Indian Statistical Institute, Kolkata, India.

- **2008**: Master of Science (M.Sc.) in *Computer and Information Science*, University of Calcutta, India.

- **2006**: Bachelor of Science (B.Sc.) (Honors) in *Computer Science*, University of Calcutta, India.

Dissertation Title

Algorithmic and Combinatorial Results on Fence Patrolling, Polygon Cutting and Geometric Spanners.

Research Interests

Computational Geometry, Theory of Algorithms, Experimental Algorithmics, Multi-agent systems, Parallel Computing.