Davenport–Schinzel Sequences and Their Geometric Applications*

Pankaj K. Agarwal† Micha Sharir‡

March 29, 2015

Abstract

An (n, s) Davenport–Schinzel sequence, for positive integers n and s, is a sequence composed of n distinct symbols with the properties that no two adjacent elements are equal, and that it does not contain, as a (possibly non-contiguous) subsequence, any alternation $a \cdots b \cdots a \cdots b \cdots$ of length $s + 2$ between two distinct symbols a and b. The close relationship between Davenport–Schinzel sequences and the combinatorial structure of lower envelopes of collections of functions make the sequences very attractive because a variety of geometric problems can be formulated in terms of lower envelopes. A near-linear bound on the maximum length of Davenport–Schinzel sequences enable us to derive sharp bounds on the combinatorial structure underlying various geometric problems, which in turn yields efficient algorithms for these problems.

*Both authors have been supported by a grant from the U.S.-Israeli Binational Science Foundation. Pankaj Agarwal has also been supported by a National Science Foundation Grant CCR-93–01259, by an Army Research Office MURI grant DAAH04-96-1-0013, by a Sloan fellowship, and by an NYI award and matching funds from Xerox Corporation. Micha Sharir has also been supported by NSF Grants CCR-91-22103 and CCR-93-11127, by a Max-Planck Research Award, and the Israel Science Fund administered by the Israeli Academy of Sciences, and the G.I.F., the German-Israeli Foundation for Scientific Research and Development.

†Department of Computer Science, Box 90129, Duke University, Durham, NC 27708-0129

‡School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel, and Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA

1 Introduction

Davenport–Schinzel sequences, introduced by H. Davenport and A. Schinzel in the 1960s, are interesting and powerful combinatorial structures that arise in the analysis and construction of the lower (or upper) envelope of collections of univariate functions, and therefore have applications in a variety of geometric problems that can be reduced to computing such an envelope. In addition, Davenport–Schinzel sequences play a central role in many related geometric problems involving arrangements of curves and surfaces. For these reasons, they have become one of the major tools in the analysis of combinatorial and algorithmic problems in geometry.

Definition 1.1 Let n and s be two positive integers. A sequence $U = \langle u_1, \ldots, u_m \rangle$ of integers is an (n, s) *Davenport–Schinzel sequence* (a *$DS(n, s)$-sequence* for short) if it satisfies the following conditions:

(i) $1 \leq u_i \leq n$ for each $i \leq m$,

(ii) $u_i \neq u_{i+1}$ for each $i < m$, and

(iii) there do not exist $s + 2$ indices $1 \leq i_1 < i_2 < \cdots < i_{s+2} \leq m$ such that

$$u_{i_1} = u_{i_3} = u_{i_5} = \cdots = a, \quad u_{i_2} = u_{i_4} = u_{i_6} = \cdots = b,$$

and $a \neq b$.

In other words, the third condition forbids the presence of long alternations of any pair of distinct symbols in a Davenport–Schinzel sequence. We refer to s as the *order* of U, to n as the *number of symbols* composing U, and to $|U| = m$ as the *length* of the sequence U. Define

$$\lambda_s(n) = \max \{\, |U| \mid U \text{ is a } DS(n, s)\text{-sequence} \,\}.$$

Curiously, the original papers by Davenport and Schinzel [47, 48] were entitled *On a combinatorial problem connected with differential equations,* because they were motivated by a particular application that involved the pointwise maximum of a collection of independent solutions of a linear differential equation. This, however, is only a special case of more general lower or upper envelopes. Davenport and Schinzel did establish in [47, 48] the connection between envelopes and these sequences, and obtained several non-trivial bounds on the length of the sequences, which were later strengthened by Szemerédi [146]. The potential of *DS*-sequences to geometric problems, however, remained unnoticed until Atallah rediscovered and applied them to several problems in dynamic computational geometry [21]. It is easy to show that $\lambda_1(n) = n$ and $\lambda_2(n) = 2n-1$ (see Theorem 3.1). Hart and Sharir [77] proved that $\lambda_3(n) = \Theta(n\alpha(n))$, where $\alpha(n)$ is the inverse Ackermann function (see below for details), and later Agarwal et al. [10] (see also Sharir [134, 135]) proved sharp bounds

on $\lambda_s(n)$ for $s > 3$. These somewhat surprising bounds show that $\lambda_s(n)$ is nearly linear in n for any fixed s. Davenport–Schinzel sequences have become a useful and powerful tool for solving numerous problems in discrete and computational geometry, usually by showing that the geometric structure being analyzed has smaller combinatorial complexity than what more naive methods would have implied. Many such geometric applications have been obtained in the past decade, and we review some of these applications below. The recent book by the authors [139] gives a more detailed description of the theory of *DS*-sequences and of their geometric applications.

As noted above, and will be shown in more detail below, Davenport–Schinzel sequences provide a complete combinatorial characterization of the lower envelope of a collection of *univariate* functions. In many geometric problems, though, one faces the more difficult problem of calculating or analyzing the envelope of a collection of multivariate functions. Even for bivariate functions this problem appears to be considerably harder than the univariate case. Nevertheless, recent progress has been made on the multivariate case, leading to almost-tight bounds on the complexity of envelopes in higher dimensions [74, 138]. Higher-dimensional lower envelopes and related combinatorial structures will be reviewed by the authors in Chapter **??** on arrangements.

The material reviewed in this chapter is a mixture of the basic combinatorial analysis of Davenport–Schinzel sequences and of their geometric applications, both combinatorial and algorithmic. Section 2 shows the connection between *DS*-sequences and lower envelopes. Sections 3–5 discuss the analysis of the maximum length of (n, s) Davenport–Schinzel sequences. Section 6 presents basic combinatorial geometric applications of Davenport–Schinzel sequences to two-dimensional arrangements of lines, segments, and arcs, and studies the role that these sequences play in various structures in such arrangements, including envelopes, individual faces, zones, and levels. Finally, Section 7 surveys a miscellany of other geometric applications of Davenport–Schinzel sequences. The material given in this survey is, to a large extent, an abridged version of the material presented in the recent book [139], and we refer the reader to that book for more details.

2 Davenport–Schinzel Sequences and Lower Envelopes

2.1 Lower envelopes of totally defined functions

Let $\mathcal{F} = \{f_1, \ldots, f_n\}$ be a collection of n real-valued, continuous totally defined functions so that the graphs of every pair of distinct functions intersect in at most s points (this is the case for polynomials of fixed degree, Chebychev systems, etc). The *lower envelope* of $\mathcal{F}$ is defined as

$$E_{\mathcal{F}}(x) = \min_{1 \le i \le n} f_i(x),$$

i.e., $E_{\mathcal{F}}$ is the *pointwise minimum* of the functions f_i; see Figure 1. Let $I_1, \ldots, I_m$ be the maximal connected intervals on the x-axis so that they cover the entire x-axis and, for each

$k \leq m$, the same function f_{u_k} appears on $E_{\mathcal{F}}$ for all points in I_k (i.e., $E_{\mathcal{F}}(x) = f_{u_k}(x)$ for all $x \in I_k$). In other words, m is the number of (maximal) connected portions of the graphs of the f_i's that constitute the graph of $E_{\mathcal{F}}$. The endpoints of the intervals I_k are called the *breakpoints* of $E_{\mathcal{F}}$. Assuming that $I_1, \ldots, I_m$ are sorted from left to right, put

$$U(\mathcal{F}) = \langle u_1, \ldots, u_m \rangle.$$

$U(\mathcal{F})$ is called the *lower-envelope sequence* of $\mathcal{F}$; see Figure 1. The *minimization diagram* of $\mathcal{F}$, denoted by $M_{\mathcal{F}}$, is the partition of the x-axis induced by the intervals $I_1, \ldots, I_m$. The endpoints of these intervals are called the *breakpoints* of $M_{\mathcal{F}}$. For convenience, we add $-\infty, +\infty$ as the breakpoints of $M_{\mathcal{F}}$.

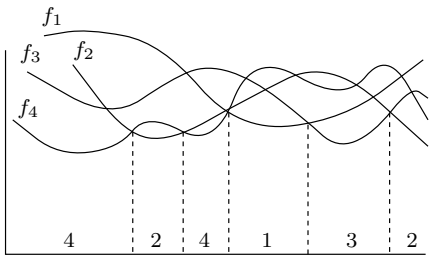

FIGURE 1. A lower-envelope sequence.

The *upper envelope* of $\mathcal{F}$ is defined, in a fully symmetric manner, to be

$$E^*_{\mathcal{F}}(x) = \max_{1 \leq i \leq n} f_i(x),$$

and the *maximization digram* $M^*_{\mathcal{F}}$ is defined as the corresponding partition of the real line, as in the case of lower envelopes. In this chapter we mostly consider lower envelopes. This choice is arbitrary, and all the results, of course, apply equally well to upper envelopes.

Theorem 2.1 ([21, 48]) *$U(\mathcal{F})$ is a $DS(n, s)$-sequence. Conversely, for any given $DS(n, s)$-sequence U, one can construct a set $\mathcal{F} = \{f_1, \ldots, f_n\}$ of continuous, totally defined, univariate functions, each pair of whose graphs intersect in at most s points, such that $U(\mathcal{F}) = U$.*

Proof (Sketch): For the first part, note that, by definition, the lower-envelope sequence $U = U(\mathcal{F})$ does not contain a pair of adjacent equal elements. For simplicity, assume that the graphs of functions in $\mathcal{F}$ intersect transversally at each intersection point. The proof can easily be extended to the case when the graphs of two functions touch each other. Suppose U contains $s+2$ indices $i_1 < i_2 < \cdots < i_{s+2}$ so that $u_{i_1} = u_{i_3} = \cdots = a$ and $u_{i_2} = u_{i_4} = \cdots = b$ for $a \neq b$. By definition of the lower-envelope sequence, we must have $f_a(x) < f_b(x)$ for $x \in (int(I_{i_1}) \cup int(I_{i_3}) \cup \cdots)$ and $f_a(x) > f_b(x)$ for $x \in (int(I_{i_2}) \cup int(I_{i_4}) \cup \cdots)$, where $int(J)$ denotes the interior of the interval J. Since f_a and f_b are continuous, there must

exist $s+1$ distinct points $x_1, \ldots, x_{s+1}$ so that x_r lies between the intervals I_{i_r} and $I_{i_{r+1}}$ and $f_a(x_r) = f_b(x_r)$, for $r = 1, \ldots, s+1$. This, however, contradicts the fact that the graphs of f_a and f_b intersect in at most s points.

For the converse statement, let $U = \langle u_1, \ldots, u_m \rangle$ be a given $DS(n, s)$-sequence. Without loss of generality, suppose the symbols $1, 2, \ldots, n$, of which U is composed, are ordered so that the leftmost appearance of symbol i in U precedes the leftmost appearance of symbol j in U if and only if $i < j$. We now define the required collection of functions $\mathcal{F} = \{f_1, \ldots, f_n\}$ as follows. We choose $m-1$ distinct "transition points" $x_2 < x_3 < \ldots < x_m$ on the x-axis, and $n+m-1$ distinct horizontal "levels," say, at $y = 1, 2, \ldots, n, -1, -2, \ldots, -(m-1)$. For each symbol $1 \le a \le n$ the graph of the corresponding function f_a is always horizontal at one of these levels, except at short intervals near some of the transition points, where it can drop very steeply from one level to a lower one. At each transition point exactly one function changes its level. More specifically:

(i) Before x_2, the function f_a is at the level $y = a$, for $a = 1, \ldots, n$.

(ii) At the transition point x_i, let $a = u_i$; then f_a drops down from its current level to the highest still "unused" level. See Figure 2 for an illustration.

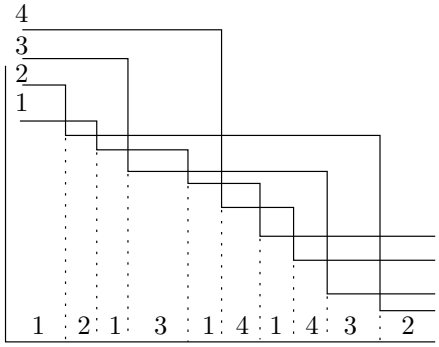

FIGURE 2. Realization of the $DS(4, 3)$-sequence $\langle 1, 2, 1, 3, 1, 4, 1, 4, 3, 2 \rangle$.

It is clear from this construction that $U(\mathcal{F}) = U$, and it can be shown that each pair of functions intersect in at most s points. This completes the proof of the theorem. □

Corollary 2.2 *For any collection $\mathcal{F} = \{f_1, \ldots, f_n\}$ of n continuous, totally defined, univariate functions, each pair of whose graphs intersect in at most s points, the length of the lower-envelope sequence $U(\mathcal{F})$ is at most $\lambda_s(n)$, and this bound can be attained by such a collection $\mathcal{F}$.*

Corollary 2.3 *Let $\mathcal{F} = \{f_1, \ldots, f_n\}$ and $\mathcal{G} = \{g_1, \ldots, g_n\}$ be two collections of n continuous, totally defined, univariate functions, such that the graphs of any pair of functions of*

$\mathcal{F}$, or of any pair of functions of $\mathcal{G}$, intersect in at most s points, and the graphs of any pair of functions in $\mathcal{F} \times \mathcal{G}$ intersect in a (possibly larger) constant number of points. Then the number of intersection points of graphs of functions in $\mathcal{F} \cup \mathcal{G}$ that lie on the boundary of the region lying between the upper envelope of $\mathcal{G}$ and the lower envelope of $\mathcal{F}$ (see Figure 3), i.e., the region

$$\Pi_{\mathcal{F},\mathcal{G}} = \{(x, y) \mid E^*_{\mathcal{G}}(x) \leq y \leq E_{\mathcal{F}}(x)\},$$

is $O(\lambda_s(n))$.

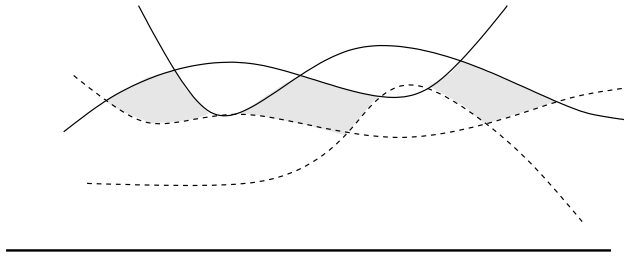

FIGURE 3. The region between $E^*_{\mathcal{G}}$ and $E_{\mathcal{F}}$ is shown shaded; the graphs of the functions in $\mathcal{F}$ (resp. in $\mathcal{G}$) are drawn solid (resp. dashed).

Proof: Let $L = (b_1, \ldots, b_t)$ be the sequence of breakpoints of $M_{\mathcal{F}}$ and $M^*_{\mathcal{G}}$, sorted from left to right. By definition, $t \leq 2\lambda_s(n)$. In each interval (b_i, b_{i+1}), the envelopes $E_{\mathcal{F}}$, $E^*_{\mathcal{G}}$ are attained by a unique pair of functions $f^{(i)} \in \mathcal{F}$, $g^{(i)} \in \mathcal{G}$. Hence, there are $O(1)$ intersection points on the boundary of $\Pi_{\mathcal{F},\mathcal{G}}$ whose x-coordinates lie in (b_i, b_{i+1}). This completes the proof of the corollary. □

2.2 Lower envelopes of partially defined functions

It is useful to note that a similar equivalence exists between Davenport–Schinzel sequences and lower envelopes of *partially defined* functions. Specifically, let $f_1, \ldots, f_n$ be a collection of partially defined and continuous functions, so that the domain of definition of each function f_i is an interval I_i, and suppose further that the graphs of each pair of these functions intersect in at most s points. The lower envelope of $\mathcal{F}$ is now defined as

$$E_{\mathcal{F}}(x) = \min f_i(x),$$

where the minimum is taken over those functions that are defined at x. One can then define the minimization diagram $M_{\mathcal{F}}$ and the lower-envelope sequence $U(\mathcal{F})$ in much the same way as for totally defined functions; see Figure 4. In this case the following theorem holds.

Theorem 2.4 ([77]) *The lower-envelope sequence $U(\mathcal{F})$ is a $DS(n, s+2)$-sequence. Conversely, for any $DS(n, s+2)$-sequence U one can construct a collection $\mathcal{F} = \{f_1, \ldots, f_n\}$*

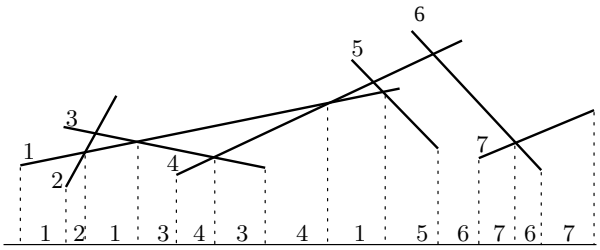

FIGURE 4. The lower envelope of a collection of (nonvertical) segments.

of partially-defined, continuous functions, each defined over an interval, and each pair of which intersect in at most s points, such that $U(\mathcal{F}) = U$.

Hence, we can conclude

Theorem 2.5 ([77]) *Let $\mathcal{F}$ be a collection of n partially-defined, continuous, univariate functions, with at most s intersection points between the graphs of any pair. Then the length of the lower-envelope sequence $U(\mathcal{F})$ is at most $\lambda_{s+2}(n)$.*

The functions constructed in Theorems 2.1 and 2.4, to realize arbitrary $DS(n, s)$-sequences, have fairly irregular structure. A problem that arises naturally in this context is whether any $DS(n, s)$-sequence can be realized as the lower envelope sequence of a collection of n partial or total functions of some canonical form. For example, can any $(n, 3)$ Davenport–Schinzel sequence be realized as the lower envelope sequence of a collection of n line segments (see Figure 4)? Some partially affirmative results on geometric relatization of $DS(n, s)$-sequences will be mentioned below, although the problem is still wide open.

2.3 Constructing lower envelopes

We conclude this section by presenting a simple, efficient divide-and-conquer algorithm for computing the minimization diagram of a set $\mathcal{F}$ of n continuous, totally defined, univariate functions, each pair of whose graphs intersect at most s times, for some constant parameter s. Here we assume a model of computation that allows us to compute the intersections between any pair of functions in $\mathcal{F}$ in $O(1)$ time.

We partition $\mathcal{F}$ into two subsets $\mathcal{F}_1, \mathcal{F}_2$, each of size at most $\lceil n/2 \rceil$, compute the minimization diagrams $M_{\mathcal{F}_1}, M_{\mathcal{F}_2}$ recursively, and merge the two diagrams to obtain $M_{\mathcal{F}}$. We merge the lists of breakpoints of $M_{\mathcal{F}_1}$ and of $M_{\mathcal{F}_2}$ into a single list $V = (v_1 = -\infty, v_2, \ldots, v_t = +\infty)$, sorted from left to right. Notice that, for any $1 \le i < t$, there is a unique pair of functions $f_i^{(1)} \in \mathcal{F}_1$, $f_i^{(2)} \in \mathcal{F}_2$, that attain the respective envelopes $E_{\mathcal{F}_1}$, $E_{\mathcal{F}_2}$ over (v_i, v_{i+1}). We compute the real roots $r_1, \ldots, r_k$ $(k \le s)$ of the function $f_i^{(1)} - f_i^{(2)}$

that lie in the interval (v_i, v_{i+1}), and add them to the list V. Let $V' = (v'_1, \ldots, v'_{t'})$ denote the new list of points. It is clear that, for each i, a unique function $f_i \in \mathcal{F}$ attains $E_{\mathcal{F}}$ over the interval (v'_i, v'_{i+1}). We associate f_i with this interval. If the same function is associated with two adjacent intervals (v'_{i-1}, v'_i) and (v'_i, v'_{i+1}), we delete the breakpoint v'_i from V'. The resulting list represents the minimization diagram $M_{\mathcal{F}}$ of $\mathcal{F}$. The total time spent in the merge step is

$$O(|V'|) = O(|V|) = O(|M_{\mathcal{F}_1}| + |M_{\mathcal{F}_2}|) = O(\lambda_s(n)).$$

Hence, the overall running time of the algorithm is $O(\lambda_s(n) \log n)$.

If the functions in $\mathcal{F}$ are partially defined, an easy modification of the above algorithm constructs $M_{\mathcal{F}}$ in time $O(\lambda_{s+2}(n) \log n)$. In this case, however, $M_{\mathcal{F}}$ can be computed in time $O(\lambda_{s+1}(n) \log n)$, using a more clever algorithm due to Hershberger [78].

Theorem 2.6 ([21, 78]) *The lower envelope of a set $\mathcal{F}$ of n continuous, totally defined, univariate functions, each pair of whose graphs intersect in at most s points, can be constructed, in an appropriate model of computation, in $O(\lambda_s(n) \log n)$ time. If the functions in $\mathcal{F}$ are partially defined, then $E_{\mathcal{F}}$ can be computed in $O(\lambda_{s+1}(n) \log n)$ time. In particular, the lower envelope of a set of n segments in the plane can be computed in optimal $O(n \log n)$ time.*

3 Simple Bounds and Variants

One of the main goals in the analysis of *DS*-sequences is to estimate the value of $\lambda_s(n)$. In this section we review some of the earlier results that established nontrivial bounds on $\lambda_s(n)$. These bounds are somewhat weaker than the best known bounds, but have simpler proofs. We begin our analysis by disposing of the two simple cases $s = 1$ and $s = 2$.

Theorem 3.1 ([48]) (a) $\lambda_1(n) = n$. (b) $\lambda_2(n) = 2n - 1$.

Proof (Sketch): (a) Let U be a $DS(n, 1)$-sequence. U cannot contain any subsequence of the form $\langle a \cdots b \cdots a \rangle$, for $a \neq b$, and any two adjacent elements of U are distinct, therefore all elements of U are distinct, which implies that $|U| \leq n$. The bound is tight, because $U = \langle 1\ 2\ 3 \cdots n \rangle$ is a $DS(n, 1)$-sequence.

(b) The proof proceeds by induction on n. The case $n = 1$ is obvious. Suppose the claim holds for $n - 1$, and let U be any $DS(n, 2)$-sequence. Without loss of generality, we can assume that the leftmost occurrence of i in U is before the leftmost occurrence of j if and only if $i < j$. It can then be shown that there is only one occurrence of n in U, or else a forbidden subsequence of the form $\langle x \cdots n \cdots x \cdots n \rangle$ would arise. Remove this single appearance of n from U, and if the two symbols adjacent to n are equal, remove also one of

them from U. The resulting sequence is clearly a $DS(n-1,2)$-sequence, and is one or two elements shorter than U. The induction hypothesis then implies $|U| \leq 2n-3+2 = 2n-1$. Since the sequence $\langle 1\ 2\ 3 \cdots n-1\ n\ n-1 \cdots 3\ 2\ 1 \rangle$ is clearly a $DS(n,2)$-sequence of length $2n-1$, the bound is tight. □

A cyclic sequence U is called a $DS(n,2)$-*cycle* if no two adjacent symbols are equal and if U does not contain a subcycle of the form $\langle a \cdots b \cdots a \cdots b \rangle$, for any $a \neq b$. Notice that the maximum length of a $DS(2,2)$-cycle is 2. The same argument as in Theorem 3.1(b) can be used to prove the following.

Corollary 3.2 *The maximum length of a $DS(n,2)$-cycle consisting of n symbols is $2n-2$.*

As we will see later, obtaining a sharp bounds on the maximum length of a $DS(n,s)$-sequence, for $s \geq 3$, is not as simple. Let us first give a simple proof of the following bound:

Theorem 3.3 ([48]) $\lambda_3(n) = O(n \log n)$.

Proof (Sketch): Let U be a $DS(n,3)$-sequence of length $\lambda_3(n)$. There must exist a symbol x that appears in U at most $\lambda_3(n)/n$ times. For any appearance of x which is neither the leftmost nor the rightmost, the symbols immediately preceding and succeeding x must be different, or else we would have obtained a forbidden subsequence of the form $\langle x \cdots yxy \cdots x \rangle$. Hence, if we erase from U all appearances of x, and, if necessary, at most two other elements, near the first and last appearances of x, we obtain a $DS(n-1,3)$-sequence, so this analysis implies the recurrence

$$\lambda_3(n) \leq \lambda_3(n-1) + \frac{\lambda_3(n)}{n} + 2\ ,$$

or

$$\frac{\lambda_3(n)}{n} \leq \frac{\lambda_3(n-1)}{n-1} + \frac{2}{n-1}\ ,$$

from which the claim follows easily. □

This bound was later improved by Davenport [47] to $O(n \log n / \log\log n)$. For any given n and s, a trivial upper bound on $\lambda_s(n)$ is $sn(n-1)/2+1$ (use Corollary 2.2 and the observation that the total number of intersections between the functions in $\mathcal{F}$ is at most $s\binom{n}{2}$). Roselle and Stanton [128] proved that, for $s > n$, one has

$$\lambda_s(n) \geq sn(n-1)/2 - cn^3,$$

where $c < 1$ is a constant. Davenport and Schinzel [48] proved that, for any fixed s, there is a constant C_s depending on s, such that $\lambda_s(n) \leq n \cdot 2^{C_s\sqrt{\log n}}$. The problem was

also studied in several early papers [51, 105, 109, 119, 127, 129, 144, 145], but the next significant improvement on the bound of $\lambda_s(n)$ was made by Szemerédi [146], who proved that $\lambda_s(n) \leq A_s n \log^* n$, for each $s \geq 3$ and for appropriate positive constants A_s (doubly exponential in s). The currently best known bounds on $\lambda_s(n)$ for $s \geq 3$, stated below, are by Hart and Sharir [77] and Agarwal et al. [10].

$$\begin{aligned}
\lambda_3(n) &= \Theta(n\alpha(n)), \\
\lambda_4(n) &= \Theta(n \cdot 2^{\alpha(n)}), \\
\lambda_{2s+2}(n) &= n \cdot 2^{\Theta(\alpha^s(n))}, \qquad \text{for } s \geq 2, \\
\lambda_{2s+3}(n) &= n \cdot 2^{O(\alpha^s(n) \log \alpha(n))}, \quad \text{for } s \geq 1;
\end{aligned}$$

more precise forms of these bounds are given in Theorems 4.3, 4.5, 5.1, and 5.2 below.

We conclude this section by mentioning some generalizations of $DS(n, s)$-sequences. Let $U = \langle u_1, u_2, \ldots, u_m \rangle$ be a $DS(n, s)$-sequence. For $1 \leq j \leq m$, let $\mu(j)$ denote the number of symbols whose leftmost occurrences in U occur at an index $\leq j$ and whose rightmost occurrences occur at an index $> j$. We define the *depth* of U to be the maximum value of $\mu(j)$, for $j \leq m$. Define a $DS(n, s, t)$-sequence to be a $DS(n, s)$-sequence whose depth is at most t, and let $\lambda_{s,t}(n)$ denote the maximum length of a $DS(n, s, t)$-sequence. Huttenlocher et al. [80] proved that $\lambda_{s,t}(n) \leq \lceil n/t \rceil \lambda_s(2t)$ (see also Har-Peled [75]). This result has the following interesting consequence:

Corollary 3.4 ([80]) *Let $\mathcal{F} = \{f_1, \ldots, f_t\}$ be a collection of t continuous, real-valued, piecewise-linear functions (i.e., the graph of each f_i is an x-monotone polygonal chain). Let n be the total number of edges in the graphs of the functions of $\mathcal{F}$. Then the lower envelope of $\mathcal{F}$ has at most $\lambda_{3,t}(n) \leq \lceil n/t \rceil \lambda_3(2t) = O(n\alpha(t))$ breakpoints.*

Adamec et al. [2] have studied some generalizations of Davenport–Schinzel sequences. In particular, they bound the length of sequences not containing more general forbidden subsequences, for example, subsequences consisting of more than two symbols. They also showed that the maximum length of a sequence not containing any forbidden subsequence $\langle a^{i_1} b^{i_2} a^{i_3} b^{i_4} \rangle$, where i_1, i_2, i_3, i_4 are some fixed positive constants, is linear. See also [92, 93, 94, 95, 96] for related results.

4 Sharp Upper Bounds on $\lambda_s(n)$

In the previous section we mentioned some weak upper bounds on $\lambda_s(n)$. The problem of bounding $\lambda_s(n)$ lay dormant for about 10 years after Szemerédi's result [146], until Hart and Sharir [77] proved a tight bound of $\Theta(n\alpha(n))$ on $\lambda_3(n)$; here $\alpha(n)$ is the inverse Ackermann function, defined below. Later, Sharir [134] extended the analysis of Hart and Sharir to prove that $\lambda_s(n) = n \cdot \alpha(n)^{O(\alpha(n)^{s-3})}$, for $s > 3$. Applying a more careful analysis, Agarwal

et al. [10] improved the bounds further, and obtained sharp, nearly tight bounds on $\lambda_s(n)$, for any fixed s. The best known upper bounds on $\lambda_s(n)$ are summarized in Theorem 4.3 (for $s = 3$) and Theorem 4.5 (for larger values of s). Since the proofs of these theorems are quite technical, we will sketch the proof of Theorem 4.3, and only briefly mention how the proof extends to the case $s > 3$.

4.1 Ackermann's function—A review

In this subsection we recall the definition of *Ackermann's function* and its functional inverse, which appears in the upper and lower bounds for $\lambda_s(n)$. Ackermann's function (also called "generalized exponentials") is an extremely fast growing function defined over the integers in the following recursive manner [1].

Let $\mathbb{N}$ denote the set of positive integers. Given a function g from a set into itself, denote by $g^{(s)}$ the composition $g \circ g \circ \ldots \circ g$ of g with itself s times, for $s \in \mathbb{N}$. Define inductively a sequence $\{A_k\}_{k=1}^{\infty}$ of functions from $\mathbb{N}$ into itself as follows:

$$\begin{array}{lcll} A_1(n) & = & 2n & n \geq 1, \\ A_k(1) & = & 2 & k \geq 2, \\ A_k(n) & = & A_{k-1}(A_k(n-1)) & n \geq 2, k \geq 2. \end{array}$$

Finally, define *Ackermann's function* itself as $A(n) = A_n(n)$. The function A grows very quickly; its first few values are: $A(1) = 2$, $A(2) = 4$, $A(3) = 16$, and $A(4)$ is an exponential "tower" of 65536 2s. See [90, 118, 123] for a discussion on Ackermann's and other rapidly growing functions.

Let α_k and α denote the functional inverses of A_k and A, respectively. That is,

$$\alpha_k(n) = \min\{s \geq 1 \mid A_k(s) \geq n\} \quad \text{and} \quad \alpha(n) = \min\{s \geq 1 \mid A(s) \geq n\}.$$

The functions α_k are easily seen to satisfy the following recursive formula:

$$\alpha_k(n) = \min\{s \geq 1 : \alpha_{k-1}^{(s)}(n) = 1\}; \tag{4.1}$$

that is, $\alpha_k(n)$ is the number of iterations of α_{k-1} needed to go from n to 1. In particular, (4.1) implies that, for $n \in \mathbb{N}$,

$$\alpha_1(n) = \lceil n/2 \rceil, \quad \alpha_2(n) = \lceil \log n \rceil, \quad \text{and} \quad \alpha_3(n) = \log^* n.$$

For each k, the function α_k is nondecreasing and unbounded. The same holds for α too, which grows more slowly than any of the α_k. Note that $\alpha(n) \leq 4$ for all $n \leq A(4)$, which is an exponential tower with 65536 2s, thus $\alpha(n) \leq 4$ for all practical values of n. We will need the following two easily established properties of $\alpha_k(n)$:

$$\alpha_{\alpha(n)}(n) = \alpha(n) \quad \text{and,} \quad \text{for } n > 4, \quad \alpha_{\alpha(n)+1}(n) \leq 4. \tag{4.2}$$

4.2 The upper bound for $\lambda_3(n)$

Let U be a $DS(n,3)$-sequence. A *chain* in U is a contiguous subsequence in which each symbol appears at most once. One can show that any such U can be decomposed into at most $2n-1$ pairwise disjoint chains, by splitting U just before the leftmost and rightmost appearances of each symbol. Let $\Psi(m,n)$ denote the maximum length of a $DS(n,3)$-sequence that can be decomposed into at most m chains.

Lemma 4.1 *Let $m, n \geq 1$, and let $b > 1$ be a divisor of m. Then there exist integers $n^\star, n_1, n_2, \ldots, n_b \geq 0$ such that*

$$n^\star + \sum_{i=1}^{b} n_i = n,$$

and

$$\Psi(m,n) \leq 4m + 4n^\star + \Psi(b, n^\star) + \sum_{i=1}^{b} \Psi\left(\frac{m}{b}, n_i\right). \tag{4.3}$$

Proof: Let U be a $DS(n,3)$-sequence, consisting of at most m chains $c_1, \ldots, c_m$, of length $\Psi(m,n)$, and let $b > 1$ be a divisor of m. Partition the sequence U into b *blocks* (contiguous subsequences) $L_1, \ldots, L_b$, so that the block L_i consists of $p = m/b$ chains $c_{(i-1)p+1}, c_{(i-1)p+2}, \ldots, c_{ip}$. Call a symbol a *internal* to block L_i if all the occurrences of a in U are within L_i. A symbol is called *external* if it is not internal to any block. Suppose that there are n_i internal symbols in block L_i and $n^\star$ external symbols; thus $n^\star + \sum_{i=1}^{b} n_i = n$.

We estimate the total number of occurrences in U of symbols that are internal to L_i, as follows. Erase all external symbols from L_i. Next scan L_i from left to right and erase each element that has become equal to the element immediately preceding it. This leaves us with a sequence $L_i^\star$, which is clearly a $DS(n_i,3)$-sequence consisting of at most m/b chains, and thus its length is at most $\Psi(m/b, n_i)$. Moreover, if two equal internal elements in L_i have become adjacent after erasing the external symbols, then these two elements must have belonged to two distinct chains, thus the total number of deletions of internal symbols is at most $(m/b) - 1$. Hence, summing over all blocks, we conclude that the total contribution of internal symbols to $|U|$ is at most

$$m - b + \sum_{i=1}^{b} \Psi\left(\frac{m}{b}, n_i\right).$$

Next, to estimate the contribution of external symbols to $|U|$, we argue as follows. For each L_i, call an external symbol a a *middle* symbol if none of its occurrences in L_i is the first or the last occurrence of a in U. Otherwise we call a a *non-middle* symbol. We will consider the contribution of middle and non-middle external symbols separately.

Consider first the middle symbols. To estimate their contribution to the length of L_i, we erase all internal and non-middle symbols from L_i, and also erase a middle symbol if it has become equal to the symbol immediately preceding it. As above, at most $(m/b)-1$ deletions of external middle symbols will be performed. Let $L_i^\star$ be the resulting subsequence, and suppose that it is composed of p_i distinct symbols. It is easily seen that $L_i^\star$ is a $DS(p_i, 1)$-sequence, so its length is at most p_i. Hence, summing over all blocks, the total contribution of external middle symbols is at most $m - b + \sum_{i=1}^{b} p_i$. But $\sum_{i=1}^{b} p_i$ is the length of the sequence obtained by concatenating all the subsequences $L_i^\star$. This concatenation can contain at most b pairs of adjacent equal elements, and if we erase each element that is equal to its predecessor, we obtain a sequence $U^\star$ which is clearly a $DS(n^\star, 3)$-sequence composed of b chains (namely the subsequences $L_i^\star$). The length of $U^\star$ is thus at most $\Psi(b, n^\star)$. Hence, the contribution of middle external elements to the length of U is at most $m + \Psi(b, n^\star)$.

Consider next the contribution of non-middle symbols. A symbol is called *starting* (resp. ending) in block L_i if does not occur in any block before (resp. after) L_i. To estimate the contribution of starting symbols to the length of L_i we erase from L_i all symbols occurring there except for starting symbols, and, if necessary, also erase each occurrence of a *starting* symbol that has become equal to the element immediately preceding it. As above, at most $(m/b) - 1$ deletions of external starting symbols will be performed. Let $L_i^\#$ be the resulting subsequence, and suppose that it is composed of p_i distinct symbols.

Note first that each external symbol can appear as a starting symbol in exactly one block, thus $\sum_{i=1}^{b} p_i = n^\star$. It is easily seen that $L_i^\#$ is a $DS(p_i, 2)$-sequence, so the length of $L_i^\#$ is at most $2p_i - 1$, and, summing over all blocks, we conclude that the contribution of all external starting symbols to the length of U is at most

$$m - b + \sum_{i=1}^{b} (2p_i - 1) = m - 2b + 2n^\star.$$

In a completely symmetric manner, the contribution of external *ending* symbols to the length of U is also at most $m - 2b + 2n^\star$. Summing up all these contributions we finally obtain the asserted inequality (4.3). □

Next, we solve the recurrence derived in the previous lemma.

Lemma 4.2 *For all $m, n \geq 1$, and for $k \geq 2$,*

$$\Psi(m, n) \leq (8k - 8)m\alpha_k(m) + (4k - 2)n. \tag{4.4}$$

Proof (Sketch): For the sake of simplicity, we will only show that for $n, s \geq 1$, $k \geq 2$, and m dividing $A_k(s)$,

$$\Psi(m, n) \leq (4k - 4)ms + (4k - 2)n. \tag{4.5}$$

If $m = A_k(s)$, then $s = \alpha_k(m)$, and (4.5) implies the assertion of the lemma for these values of m also. The case of an arbitrary m is then easy to handle; see [77, 139] for details.

We will use (4.3) repeatedly to obtain the series of upper bounds on Ψ, stated in (4.5) for $k = 2, 3, \ldots$. At each step we choose b in an appropriate manner, and estimate $\Psi(b, n^\star)$ using the bound obtained in the preceding step. This yields a new recurrence relation on Ψ, which we solve to obtain a better upper bound on Ψ.

Specifically, we proceed by double induction on k and s. For $k = 2$, m divides $A_2(s) = 2^s$, so m is a power of 2. Choose $b = 2$ in (4.3); it is easily checked that $\Psi(b, n^\star) = \Psi(2, n^\star) = 2n^\star$ for all $n^\star$, so (4.3) becomes

$$\Psi(m, n) \le 4m + 6n^\star + \Psi\left(\frac{m}{2}, n_1\right) + \Psi\left(\frac{m}{2}, n_2\right).$$

The solution to this recurrence relation, for m a power of 2 and $n = n^\star + n_1 + n_2$ arbitrary, is easily verified to be

$$\Psi(m, n) \;\le\; 4m \log m + 6n.$$

The case $k > 2$ and $s = 1$ is now a consequence of this bound (because m divides $A_k(1) = 2$ in this case).

Suppose next that $k > 2$ and $s > 1$, and that the induction hypothesis is true for all $k' < k$ and $s' \ge 1$, and for $k' = k$ and all $s' < s$. Let $m = A_k(s)$, and $t = A_k(s-1)$, and choose $b = m/t$, which is an integer dividing $m = A_k(s) = A_{k-1}(t)$. Hence, by the induction hypothesis for $k-1$ and t, we have

$$\Psi(b, n^\star) \;\le\; (4k-8)bt + (4k-6)n^\star \;=\; (4k-8)m + (4k-6)n^\star.$$

Then (4.3) becomes

$$\Psi(m, n) \le (4k-8)m + (4k-6)n^\star + 4m + 4n^\star + \sum_{i=1}^{b} \Psi(t, n_i).$$

Using the induction hypothesis once more (for k and $s-1$), we obtain

$$\begin{aligned} \Psi(m, n) &\le (4k-4)m + (4k-2)n^\star + \sum_{i=1}^{b}((4k-4)t(s-1) + (4k-2)n_i) \\ &= (4k-4)ms + (4k-2)n, \end{aligned}$$

because $n^\star + \sum_{i=1}^{b} n_i = n$.

The case where m only divides $A_k(s)$ is handled by taking a concatenation of $p = A_k(s)/m$ copies of a sequence whose length is $\Psi(m, n)$, using pairwise-disjoint sets of symbols for the copies. The concatenated sequence is composed of pn symbols and has at most pm chains, so

$$p\Psi(m, n) \le \Psi(pm, pn) \le (4k-4)pms + (4k-2)pn\,,$$

from which (4.5) follows.

This completes the proof of the asserted bound. □

Theorem 4.3 ([77]) $\lambda_3(n) = O(n\alpha(n))$.

Proof: By putting $k = \alpha(m) + 1$ in (4.4) and using (4.2), we obtain

$$\Psi(m, n) \leq 32m\alpha(m) + (4\alpha(m) + 2)n.$$

As noted in the beginning of this subsection, $\lambda_3(n) \leq \Psi(2n-1, n)$. Since $\alpha(2n-1) \leq \alpha(n) + 1$, the theorem follows. □

Applying a more careful analysis, Klazar [95] has shown that

$$\lambda_3(n) \leq 4n\alpha(n) + O(n\sqrt{\alpha(n)}),$$

provided that n is sufficiently large.

An immediate corollary of Theorem 4.3 is that the lower envelope of n segments in the plane has $O(n\alpha(n))$ breakpoints.

4.3 Upper bounds on $\lambda_s(n)$

We now briefly mention how the upper bounds on $\lambda_s(n)$, for $s > 3$, are derived in [10]. Let $\Psi_s^t(m, n)$ denote the maximum length of a $DS(n, s)$-sequence composed of at most m contiguous subsequences, each of which is a $DS(n, t)$-sequence. As above, Agarwal et al. [10] obtain a recurrence relation for $\Psi_s(m, n)$, the length of a $DS(n, s)$-sequences composed of at most m chains, but the recurrence is now written in terms of Ψ_s and Ψ_s^{s-2}. Let S be a given $DS(n, s)$-sequence composed of at most m chains. The analysis in [10] divides S into b blocks and counts the contributions of internal, middle, and non-middle symbols separately, in a manner similar to that given above. This leads to the following lemma.

Lemma 4.4 *Let $m, n \geq 1$ and $1 < b < m$ be integers. For any partitioning $m = \sum_{i=1}^{b} m_i$, with $m_1, \ldots, m_b \geq 1$, there exist integers $n^\star$, $n_1, n_2, \ldots, n_b \geq 0$ such that*

$$n^\star + \sum_{i=1}^{b} n_i = n$$

and

$$\Psi_s(m, n) \leq \Psi_s^{s-2}(b, n^\star) + 2\Psi_{s-1}(m, n^\star) + 4m + \sum_{i=1}^{b} \Psi_s(m_i, n_i). \tag{4.6}$$

If we choose $b = 2$, the solution of the recurrence is $O(n \log^{s-2} n)$. However, extending the proof of Lemma 4.2, but using a rather involved analysis, one can obtain the following bounds on $\lambda_s(n)$.

Theorem 4.5 ([10]) (i) $\lambda_4(n) = O(n \cdot 2^{\alpha(n)})$.

(ii) *For $s > 1$, there exists a polynomial $C_s(q)$ of degree at most $s-1$, such that*

$$\begin{aligned} \lambda_{2s+1}(n) &\leq n \cdot 2^{\alpha^{s-1}(n)\log\alpha(n)+C_s(\alpha(n))}, \\ \lambda_{2s+2}(n) &\leq n \cdot 2^{\alpha^{s}(n)+C_s(\alpha(n))}. \end{aligned}$$

5 Lower Bounds on $\lambda_s(n)$

An even more surprising result in the theory of Davenport–Schinzel sequences is that the bounds stated in Theorems 4.3 and 4.5 are optimal for $s = 3$ and 4, and are very close to optimal for $s > 4$. The first superlinear bound on $\lambda_s(n)$ was obtained by Hart and Sharir [77], who proved that $\lambda_3(n) = \Omega(n\alpha(n))$. Their original proof transforms $DS(n, 3)$-sequences into certain path compression schemes on rooted trees. A more direct proof for the lower bound on $\lambda_3(n)$ was given by Wiernik and Sharir [149] — they describe an explicit recursive scheme for constructing a $DS(n, 3)$-sequence of length $\Omega(n\alpha(n))$. See also [97] for another proof of the same lower bound. We sketch Wiernik and Sharir's construction, omitting many details, which can be found in [139, 149].

Let $\{C_k(m)\}_{k\geq 1}$ be a sequence of functions from $\mathbb{N}$ to itself, defined by

$$\begin{aligned} C_1(m) &= 1 && m \geq 1, \\ C_k(1) &= 2C_{k-1}(2) && k \geq 2, \\ C_k(m) &= C_k(m-1) \cdot C_{k-1}(C_k(m-1)) && k \geq 2, m \geq 2. \end{aligned}$$

It can be shown that, for all $k \geq 4, m \geq 1$,

$$A_{k-1}(m) \leq C_k(m) \leq A_k(m+3). \tag{5.1}$$

In what follows, let $\mu = C_k(m-1), \nu = C_{k-1}(C_k(m-1))$, and $\gamma = \mu \cdot \nu$.

For each $k, m \geq 1$, we construct a sequence $S_k(m)$ that satisfies the following two properties:

(P1) $S_k(m)$ is composed of $N_k(m) = m \cdot C_k(m)$ distinct symbols. These symbols are named (d, l), for $d = 1, \ldots, m$, $l = 1, \ldots, \gamma$, and are ordered in lexicographical order, so that $(d, l) < (d', l')$ if $l < l'$ or $l = l'$ and $d < d'$.

(P2) $S_k(m)$ contains γ *fans* of size m, where each fan is a contiguous subsequence of the form $\langle (1, l)\,(2, l) \cdots (m, l)\rangle$, for $l = 1, \ldots, \gamma$.

Since fans are pairwise disjoint, by definition, the naming scheme of the symbols of $S_k(m)$ can be interpreted as assigning to each symbol the index l of the fan in which it appears, and its index d within that fan. The construction of $S_k(m)$ proceeds by double induction on k and m, as follows.

1. $k = 1$: The sequence is a single fan of size m: $S_1(m) = \langle (1,1)\,(2,1) \cdots (m,1) \rangle$. Properties (P1) and (P2) clearly hold here ($C_1(m) = 1$).

2. $k = 2$: The sequence contains a pair of disjoint fans of size m, with a block of elements following each of these fans. Specifically,

$$\begin{aligned} S_2(m) \;\; = \;\; & \langle (1,1)\,(2,1) \cdots (m-1,1)\,(m,1)\,(m-1,1) \cdots (1,1) \\ & (1,2)\,(2,2) \cdots (m-1,2)\,(m,2)\,(m-1,2) \cdots (1,2) \rangle. \end{aligned}$$

   Indeed, $S_2(m)$ contains $C_2(m) = 2$ fans and is composed of $2m$ dsitinct symbols.

3. $k \geq 3, m = 1$: The sequence is identical to the sequence for $k' = k-1$ and $m' = 2$, except for renaming of its symbols and fans: $S_{k-1}(2)$ contains $C_{k-1}(2) = \frac{1}{2}C_k(1)$ fans, each of which consists of two symbols; the symbol renaming in $S_k(1)$ causes each of these two elements to become a 1-element fan. Properties (P1) and (P2) clearly hold.

4. The general case $k \geq 3, m > 1$:

   (i) Generate inductively the sequence $S' = S_k(m-1)$; by induction, it contains μ fans of size $m-1$ each and is composed of $(m-1) \cdot \mu$ symbols.

   (ii) Create ν copies of S' whose sets of symbols are pairwise disjoint. For each $j \leq \nu$, rename the symbols in the jth copy S'_j of S' as (d,i,j) where $1 \leq d \leq m-1$ is the index of the symbol in the fan of S'_j containing it, and $1 \leq i \leq \mu$ is the index of this fan in S'_j.

   (iii) Generate inductively the sequence $S^\star = S_{k-1}(\mu)$ whose set of symbols is disjoint from that of any S'_j; by induction, it contains ν fans of size μ each. Rename the symbols of $S^\star$ as (m,i,j) (where i is the index of that symbol within its fan, and j is the index of that fan in $S^\star$). Duplicate the last element (m,μ,j) in each of the ν fans of $S^\star$.

   (iv) For each $1 \leq i \leq \mu$, $1 \leq j \leq \nu$, extend the ith fan of S'_j by duplicating its last element $(m-1,i,j)$, and by inserting the corresponding symbol (m,i,j) of $S^\star$ between these duplicated appearances of $(m-1,i,j)$. This process extends the $(m-1)$-fans of S'_j into m-fans and adds a new element after each extended fan.

   (v) Finally construct the desired sequence $S_k(m)$ by merging the ν copies S'_j of S' with the sequence $S^\star$. This is done by replacing, for each $1 \leq j \leq \nu$, the jth fan of $S^\star$ by the corresponding copy S'_j of S', as modified in (iv) above. Note that the duplicated copy of the last element in each fan of $S^\star$ (formed in step (iii) above) appears now after the copy S'_j that replaces this fan; see Figure 5 for an illustration of this process.

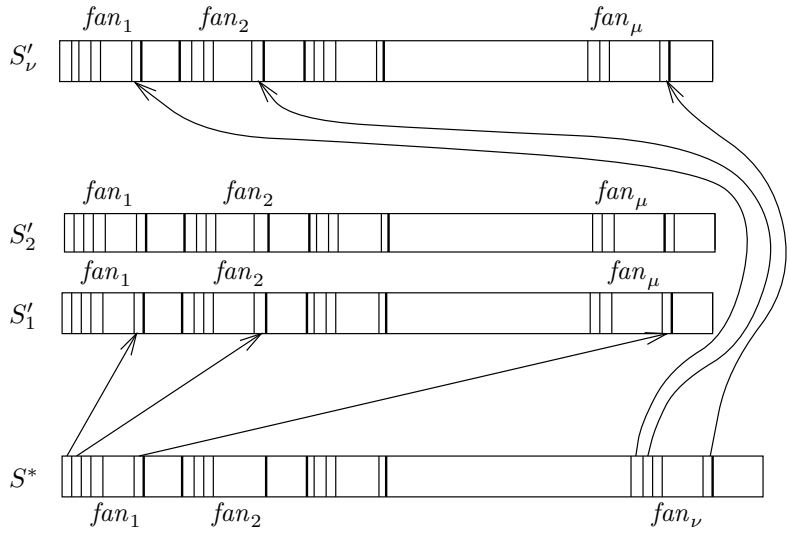

FIGURE 5. Lower bound construction: merging the subsequences.

It is easily checked that $S_k(m)$ consists of

$$N_k(m) = \nu(m-1)\mu + \mu C_{k-1}(\mu) = mC_k(m)$$

symbols, and it can also be shown that $S_k(m)$ is a $DS(N_k(m), 3)$-sequence satisfying properties (P1) and (P2). If we let $\sigma_k(m)$ denote the length of $S_k(m)$, then

$$\begin{aligned} \sigma_1(m) &= m, \\ \sigma_2(m) &= 4m-2, \\ \sigma_k(1) &= \sigma_{k-1}(2), \\ \sigma_k(m) &= \nu\sigma_k(m-1) + \sigma_{k-1}(\mu) + \nu(\mu+1). \end{aligned}$$

The third term in the last equation is due to the duplication of the rightmost symbol of each fan of S^* and of each S'_j (see Steps 4 (iii)-(iv)). Using a double induction on k and m, one can prove that

$$\sigma_k(m) > (km-2)C_k(m) + 1.$$

Theorem 5.1 ([77, 149]) $\lambda_3(n) = \Omega(n\alpha(n))$.

Proof: Choose $m_k = C_{k+1}(k-3)$. Then

$$n_k = N_k(m_k) = C_{k+1}(k-2) \le A_{k+1}(k+1)$$

where the last inequality follows from (5.1). Therefore $\alpha(n_k) \le k+1$, and hence

$$\lambda_3(n_k) \ge \sigma_k(m_k) \ge kn_k - 2C_k(m_k) \ge (k-2)n_k \ge n_k(\alpha(n_k) - 3).$$

As shown in [139], this bound can be extended to any integer n, to prove that $\lambda_3(n) = \Omega(n\alpha(n))$. □

Generalizing the above construction and using induction on s — basically replacing each chain of the sequence $S_k(m)$ by a $DS(n, s-2)$-sequence, which, in turn, is constructed recursively — Sharir [135] proved that $\lambda_{2s+1}(n) = \Omega(n\alpha(n)^s)$. Later Agarwal et al. [10] proved that the upper bounds stated in Theorem 4.5 are almost optimal. In particular, using a rather involved doubly-inductive scheme, they constructed a $DS(n, 4)$-sequence of length $\Omega(n2^{\alpha(n)})$. Then, by recursing on s, they generalized their construction of $DS(n, 4)$-sequences to higher-order sequences. The following theorem summarizes their result.

Theorem 5.2 ([10]) (i) $\lambda_4(n) = \Omega(n \cdot 2^{\alpha(n)})$.

(ii) *For $s > 1$, there exists a polynomial $Q_s(q)$ of degree at most $s-1$, such that*

$$\lambda_{2s+2}(n) \geq n \cdot 2^{\frac{\alpha^s(n)}{s!} + Q_s(\alpha(n))}.$$

Open Problem 1 *Obtain tights bounds on $\lambda_s(n)$ for $s > 4$, especially for odd values of s.*

Wiernik and Sharir [149] proved that the $DS(n, 3)$-sequence $S_k(m)$ constructed above can be realized as the lower envelope sequence of a set of n segments, which leads to the following fairly surprising result:

Theorem 5.3 ([149]) *The lower envelope of n segments can have $\Omega(n\alpha(n))$ breakpoints in the worst case.*

Shor [142] gave a simpler example of n segments whose lower envelope also has $\Omega(n\alpha(n))$ breakpoints. These results also yield an $\Omega(n\alpha(n))$ lower bound on many other unrelated problems, including searching in totally monotone matrices [91] and counting the number of distint edges in the convex hull of a planar point set as the points are being updated dynamically [148]. Shor has also shown that there exists a set of n degree-4 polynomials whose lower envelope has $\Omega(n\alpha(n))$ breakpoints [143] (which is somewhat weak, because the upper bound for this quantity is $\lambda_4(n) = O(n \cdot 2^{\alpha(n)})$). We conclude this section by mentioning another open problem, which we believe is one of the most challenging and interesting problems related to Davenport–Schinzel sequences.

Open Problem 2 *Is there a natural geometric realization of higher order sequences? For example, can the lower envelope of n conic sections have $\Omega(n2^{\alpha(n)})$ breakpoints?*

6 Davenport–Schinzel Sequences and Arrangements

In this section we consider certain geometric and topological structrues induced by a family of arcs in the plane, where Davenport–Schinzel sequences play a major role in their analysis. Specifically, let $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ be a collection of n Jordan arcs in the plane, each pair of which intersect in at most s points, for some fixed constant s.[1]

Definition 6.1 The *arrangement* $\mathcal{A}(\Gamma)$ of Γ is the planar subdivision induced by the arcs of Γ; that is, $\mathcal{A}(\Gamma)$ is a planar map whose *vertices* are the endpoints of the arcs of Γ and their pairwise intersection points, whose *edges* are maximal (relatively open) connected portions of the γ_i's that do not contain a vertex, and whose *faces* are the connected components of $\mathbb{R}^2 - \bigcup \Gamma$. The *combinatorial complexity* of a face is the number of vertices (or edges) on its boundary, and the *combinatorial complexity* of $\mathcal{A}(\Gamma)$ is the total complexity of all of its faces.

The maximum combinatorial complexity of $\mathcal{A}(\Gamma)$ is clearly $\Theta(sn^2) = \Theta(n^2)$, and $\mathcal{A}(\Gamma)$ can be computed in time $O(n^2 \log n)$, under an appropriate model of computation, using the sweep-line algorithm of Bentley and Ottmann [28]. A slightly faster algorithm, with running time $O(n\lambda_{s+2}(n))$, is mentioned in Section 6.3. Many applications, however, need to compute only a small portion of the arrangement, such as a single face, a few faces, or some other substructures that we will consider shortly. Using *DS*-sequences, one can show that the combinatorial complexity of these substructures is substantially smaller than that of the entire arrangement. This fact is then exploited in the design of efficient algorithms, whose running time is close to the bound on the complexity of the substructures that these algorithms aim to construct. In this section we review combinatorial and algorithmic results related to these substructures, in which *DS*-sequences play a crucial role.

6.1 Complexity of a single face

It is well known that the complexity of a single face in an arrangement of n lines is at most n [121], and a linear bound on the complexity of a face in an arrangement of rays is also known (see Alevizos et al. [13, 14]). The result of Wiernik and Sharir [149] on the lower envelopes of segments implies that the unbounded face in an arrangement of n line segments has $\Omega(n\alpha(n))$ vertices in the worst case. A matching upper bound was proved by Pollack et al. [120], which was later extended by Guibas et al. [72] to general Jordan arcs. The case of closed or unbounded Jordan curves was treated in [132].

[1] A *Jordan arc* is an image of the closed unit interval under a continuous bijective mapping. Similarly, a *closed Jordan curve* is an image of the unit circle under a similar mapping, and an *unbounded Jordan curve* is an image of the open unit interval (or of the entire real line) that separates the plane.

Theorem 6.2 ([72, 132]) *Let Γ be a set of n Jordan arcs in the plane, each pair of which intersect in at most s points, for some fixed constant s. Then the combinatorial complexity of any single face in $\mathcal{A}(\Gamma)$ is $O(\lambda_{s+2}(n))$. If each arc in Γ is a Jordan curve (closed or unbounded), then the complexity of a single face is at most $\lambda_s(n)$.*

Proof (Sketch): We only consider the first part of the theorem; the proof of the second part is simpler, and can be found in [132, 139]. Let f be a given face in $\mathcal{A}(\Gamma)$, and let C be a connected component of its boundary. We can assume that C is the only connected component of ∂f. Otherwise, we repeat the following analysis for each connected component and sum their complexities. Since each arc appears in at most one connected component, the bound follows. For each arc γ_i, let u_i and v_i be its endpoints, and let γ_i^+ (respectively, γ_i^-) be the directed arc γ_i oriented from u_i to v_i (respectively, from v_i to u_i).

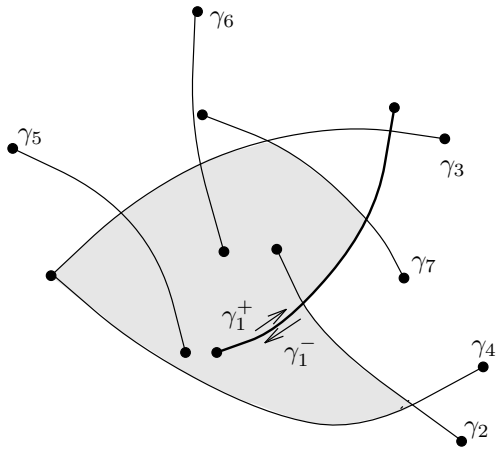

FIGURE 6. A single face and its associated boundary sequence; all arcs are positively oriented from left to right.

Without loss of generality, assume that C is the exterior boundary component of f. Traverse C in counterclockwise direction (so that f lies to our left) and let $S = \langle s_1, s_2, \ldots, s_t \rangle$ be the circular sequence of oriented arcs in Γ in the order in which they appear along C (if C is unbounded, S is a linear, rather than circular, sequence). More precisely, if during our traversal of C we encounter an arc γ_i and follow it in the direction from u_i to v_i (respectively, from v_i to u_i) then we add γ_i^+ (respectively, γ_i^-) to S. See Figure 6 for an illustration. Note that in this example *both* sides of an arc γ_i belong to the outer connected component.

Let $\xi_1, \ldots, \xi_{2n}$ denote the oriented arcs of Γ. For each ξ_i we denote by $|\xi_i|$ the nonoriented arc γ_j coinciding with ξ_i. For the purpose of the proof, we transform each arc γ_i into a very thin closed Jordan curve $\gamma_i^\star$ by taking two nonintersecting copies of γ_i lying very close to one another, and by joining them at their endpoints. This will perturb the face f slightly

but can always be done in such a way that the combinatorial complexity of C does not decrease. Note that this transformation allows a natural identification of one of the two sides of $\gamma_i^\star$ with γ_i^+ and the other side with γ_i^-.

It can be shown (see [72, 139]) that the portions of each arc ξ_i appear in S in a circular order that is consistent with their order along the oriented ξ_i. In particular, there exists a starting point in S (which depends on ξ_i) so that if we read S in circular order starting from that point, we encounter these portions of ξ_i in their order along ξ_i. For each directed arc ξ_i, consider the linear sequence V_i of all appearances of ξ_i in S, arranged in the order they appear along ξ_i. Let μ_i and ν_i denote respectively the index in S of the first and of the last element of V_i. Consider $S = \langle s_1, \ldots, s_t \rangle$ as a linear, rather than a circular, sequence (this change is not needed if C is unbounded). For each arc ξ_i, if $\mu_i > \nu_i$ we split the symbol ξ_i into two distinct symbols ξ_{i1}, ξ_{i2}, and replace all appearances of ξ_i in S between the places μ_i and t (respectively, between 1 and ν_i) by ξ_{i1} (respectively, by ξ_{i2}). Note that the above claim implies that we can actually split the arc ξ_i into two connected subarcs, so that all appearances of ξ_{i1} in the resulting sequence represent portions of the first subarc, whereas all appearances of ξ_{i2} represent portions of the second subarc. This splitting produces a sequence $S^\star$, of the same length as S, composed of at most $4n$ symbols.

With all these modifications, one can then prove that S^* is a $DS(4n, s+2)$-sequence. This is done by showing that each quadruple of the form $\langle a \cdots b \cdots a \cdots b \rangle$ in S^* corresponds, in a unique manner, to an intersection point between the two arcs of Γ that a and b represent. See [72, 139] for more details. This completes the proof of the first part of the theorem. □

Theorem 6.2 has the following interesting consequence. Let $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ be a set of n closed Jordan curves, each pair of which intersects in at most s points. Let $K = conv(\Gamma)$ be the convex hull of the curves in Γ. Divide the boundary of K into a minimum number of subarcs, $\alpha_1, \alpha_2, \ldots, \alpha_m$, such that the relative interior of each α_i has a nonempty intersection with exactly one of the curves γ_j. Then the number m of such arcs is at most $\lambda_s(n)$; see [132] for a proof.

Recently, Arkin et al. [19] showed that the complexity of a single face in an arrangement of line segments with h distinct endpoints is only $O(h \log h)$ (even though the number of segments can be $\Theta(h^2)$). A matching lower bound is proved is by Matoušek and Valtr [103]. The upper bound by Arkin et al. does not extend to general arcs. Har-Peled [75] has also obtained improved bounds on the complexity of a single face in many special cases.

6.2 Computing a single face

Let Γ be a collection of n Jordan arcs, as above, and let x be a point that does not lie on any arc of Γ. We wish to compute the face of $\mathcal{A}(\Gamma)$ that contains x. We assume that each arc in Γ has at most a constant number of points of vertical tangency, so that we can break it into $O(1)$ x-monotone Jordan arcs.

We assume a model of computation allowing infinite-precision real arithmetic, in which certain primitive operations involving one or two arcs (e.g., computing the intersection points of a pair of arcs, the points of vertical tangency of an arc, the intersections of an arc with a vertical line, etc.) are assumed to take constant time.

If Γ is a set of n lines, or a set of n rays, then a single face can be computed in time $O(n \log n)$. In the case of lines, this is done by dualizing the lines to points and using any optimal convex hull algorithm [121]; the case of rays is somewhat more involved, and is described in [13, 14]. However, these techniques do not extend to arrangements of more general Jordan arcs. Pollack et al. [120] presented an $O(n\alpha(n)\log^2 n)$-time algorithm for computing the unbounded face in certain arrangements of line segments, but the first algorithm that works for general arcs was given by Guibas et al. [72]. Later, several other efficient algorithms—both randomized and deterministic—have been proposed. We first present randomized (Las Vegas) algorithms[2] for computing a single face, and then review the deterministic solution of [72], and mention some other related results. Randomized algorithms have recently been designed for many geometric problems; see, e.g., [44, 112, 133]. They are often much simpler than their deterministic counterparts, and are sometimes more efficient, as the present case will demonstrate. The efficiency of a Las Vegas randomized algorithm will be measured by its expected running time in the worst case, where the expectation is taken with respect to the internal randomizations performed by the algorithm.

Randomized algorithms. The randomized algorithms that we will describe actually compute the so-called *vertical decomposition* of f. This decomposition, which we denote by $f^{||}$, is obtained by drawing a vertical segment from each vertex and from each point of vertical tangency of the boundary of f in both directions, and extend it until it meets another edge of f, or else all the way to $\pm\infty$. The vertical decomposition partitions f into 'pseudo-trapezoidal' cells, each bounded by at most two arcs of Γ and at most two vertical segments. To simplify the presentation, we will refer to these cells simply as trapezoids; see Figure 7 for an illustration.

We first present a rather simple randomized divide-and-conquer algorithm due to Clarkson [41] (see also [139]). The basic idea of the algorithm is as follows: Randomly choose a subset $\Gamma_1 \subseteq \Gamma$ of $\lfloor n/2 \rfloor$ arcs. Recursively compute the vertical decompositions $f_1^{||}, f_2^{||}$ of the faces f_1, f_2 containing x in $\mathcal{A}(\Gamma_1)$ and in $\mathcal{A}(\Gamma \setminus \Gamma_1)$, respectively. Then merge $f_1^{||}$ and $f_2^{||}$ to compute the vertical decomposition of the face f of $\mathcal{A}(\Gamma)$ that contains x. The merge step essentially performs a simultaneous depth-first search over the trapezoids of $f_1^{||}$ and of $f_2^{||}$, in which it computes the *intersection cells* $\triangle_1 \cap \triangle_2$, for $\triangle_1 \in f_1^{||}$, $\triangle_2 \in f_2^{||}$, that lie in $f^{||}$. After having computed all such intersection cells, $f^{||}$ can be computed in additional $O(|f^{||}|)$ time; see [139] for details. Although the merge step is quite naive, and can take quadratic time in the worst case, one can nevertheless show that the randomization makes

[2] A *Las Vegas* algorithm always terminates with the correct output, but its running time is a random variable (over the internal randomizations it performs).

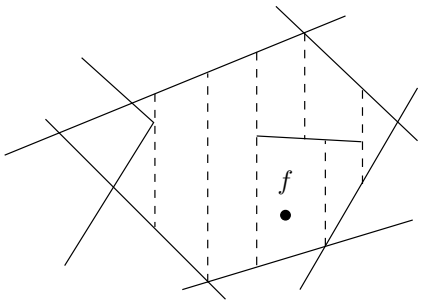

FIGURE 7. Vertical decomposition of a face in an arrangement of line segments; here each cell is indeed a trapezoid or a triangle.

this step fast—its expected time is only $O(\lambda_{s+2}(n))$. Hence, the expected running time of the algorithm is $O(\lambda_{s+2}(n)\log n)$.

The second randomized algorithm, due to Chazelle et al. [34], constructs the vertical decomposition $f^{||}$ of the face containing x incrementally, by adding the arcs of Γ one by one in a random order (the choice of the insertion order is the only randomized step in the algorithm), where each permutation of Γ is chosen with equal probability. While the worst-case running time of this algorithm is also quadratic, the expected running time is only $O(\lambda_{s+2}(n)\log n)$, as for the preceding algorithm.

The basic idea of this algorithm is as follows. Let $\langle\gamma_1, \gamma_2, \ldots, \gamma_n\rangle$ denote the insertion sequence, let $\Gamma_i = \{\gamma_1, \ldots, \gamma_i\}$, and let $f_i^{||}$ be the vertical decomposition of the face containing x in $\mathcal{A}(\Gamma_i)$, for $i = 1, \ldots, n$. When γ_{i+1} is inserted, it may chop off a part of f_i by separating it from the point x, so some of the trapezoids of $f_i^{||}$ may not appear in $f_{i+1}^{||}$, and some of them, which are crossed by γ_{i+1}, will have to be replaced by new trapezoids that have γ_{i+1} on their boundary. Thus, adding γ_{i+1} requires the following steps: Compute the set of trapezoids in $f_i^{||}$ that γ_{i+1} intersects; determine the set of new cells that appear in $f_{i+1}^{||}$, and find the portion of f_i that is chopped off by γ_{i+1}, if any; finally, discard the trapezoids of $f_i^{||}$ that do not appear in $f_{i+1}^{||}$.

To facilitate the execution of these steps, the algorithm stores $f_i^{||}$ as a vertical adjacency graph, whose edges connect pairs of trapezoids sharing a vertical edge (more precisely, having overlapping vertical edges); for each trapezoid in $f_i^{||}$, we store the list of trapezoids that are its neighbors in the vertical adjacency graph. The algorithm also maintains a directed acyclic graph (dag) G, referred to as the *history dag*. The nodes of G, after the ith insertion stage, correspond to the trapezoids that appeared in at least one $f_j^{||}$, for $j \le i$. The root of the dag corresponds to the entire plane. There is a directed edge from a node v to a node w if the corresponding trapezoids τ_v and τ_w intersect and if τ_v (resp. τ_w) appeared in $f_j^{||}$ (resp. $f_k^{||}$) for some $j < k$. If τ_v is a trapezoid of $f_i^{||}$, then v is an *active* leaf (in the

version of G after the ith insertion), and if τ_v was a trapezoid of $f_{i+1}^{||}$ but is not in $f_i^{||}$, and γ_i does not cross τ_v, then v is an *inactive* leaf, in the sense that no successor of τ_v will ever be created. All other nodes of G are *inner* nodes, and represent trapezoids that existed in some $f_j^{||}$, but were crossed by some arc γ_k, for $j < k \leq i$. The purpose of the dag G is to facilitate, through a top-down traversal of it, a simple and efficient technique for finding all active trapezoids that the newly inserted arc intersects.

How exactly the above steps are executed and how the data structures are updated is somewhat involved, and is described in detail in [34, 139]. As mentioned above, the expected running time of the algorithm is $O(\lambda_{s+2}(n) \log n)$. Moreover, the expected size and depth of G are $O(\lambda_{s+2}(n))$ and $O(\log n)$, respectively, so we also obtain a point-location data structure that can determine, in $O(\log n)$ expected time, whether a query point lies in f. A somewhat simpler variant of the randomized incremental algorithm is given in [49].

Theorem 6.3 ([34, 41, 49]) *Given a collection Γ of n Jordan arcs, each pair of which intersect in at most s points, and a point x not lying on any arc, the face of $\mathcal{A}(\Gamma)$ containing x can be computed by a randomized algorithm in $O(\lambda_{s+2}(n) \log n)$ expected running time, in an appropriate model of computation.*

Deterministic algorithms. We now sketch a deterministic, divide-and-conquer algorithm, due to Guibas et al. [72], for computing f. The high-level description of the algorithm is quite simple, and is similar to the first randomized algorithm described above. We partition Γ into two subsets Γ_1, Γ_2, of roughly $n/2$ arcs each, recursively compute the faces, f_1, f_2, of $\mathcal{A}(\Gamma_1)$, $\mathcal{A}(\Gamma_2)$, respectively, that contain x, and then 'merge' these two faces to obtain the desired face f. Note that f is the connected component of $f_1 \cap f_2$ containing x. However, as already noted, it is generally too expensive to compute this intersection in its entirety, and then select the component containing x, because the boundaries of f_1 and f_2 might have $\Omega(n^2)$ points of intersection. We therefore need a more careful way of performing the merge.

The setup for the merge step is as follows. We are given two connected (but not necessarily simply connected) regions in the plane, which we denote, respectively, as the red region R and the blue region B. Both regions contain the point x in their interior, and our task is to calculate the connected component f of $R \cap B$ that contains x. The boundaries of R and B are composed of (maximal connected) portions of the given curves in Γ, each of which will be denoted in what follows as an *arc segment* (or 'subarc').

For technical reasons, we extend this task as follows. Let P be the set of points containing x and all endpoints of the arcs of Γ that lie on the boundary of either R or B. Clearly, $|P| \leq 2n + 1$. For each $w \in P$, let f_w denote the connected component of $R \cap B$ that contains w (these components are not necessarily distinct, and some may be empty). Our task is now to calculate all these components (but produce each distinct component just once, even if it contains several points of P). We refer to this task as the *red–blue merge*.

We call the resulting components f_w *purple regions*, as each of them is covered by both the red and the blue regions. An illustration of this merge is shown below in Figure 8.

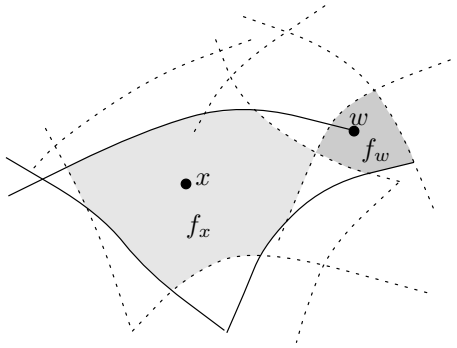

FIGURE 8. The red–blue merge; the solid arcs are the blue arcs, and the dashed arcs are red.

The algorithm relies heavily on the following technical result, called the *combination lemma*, which is interesting in its own right. We first introduce a few notation. Let $R_1, \ldots, R_m$ be a collection of distinct faces in an arrangement of a set Γ_r of 'red' Jordan arcs, and let $B_1, \ldots, B_n$ be a similar collection of faces in an arrangement of a set Γ_b of 'blue' Jordan arcs (where, again, each pair of arcs from $\Gamma_r \cup \Gamma_b$ are assumed to intersect in at most s points). Let $P = \{p_1, \ldots, p_k\}$ be a collection of points, so that each $p_i \in P$ belongs to one red face R_{m_i} and to one blue face B_{n_i}. Let E_i be the connected component of $R_{m_i} \cap B_{n_i}$ containing p_i (i.e., E_i is the 'purple' face of the combined arrangement $\mathcal{A}(\Gamma_r \cup \Gamma_b)$ containing p_i). Then we have the following result.

Lemma 6.4 (Combination Lemma, [72]) *The total combinatorial complexity of all the regions E_i is at most $O(r + b + k)$, where r and b are the total number of arc segments composing the boundaries of the red faces and of the blue faces, respectively.*

Remark 6.5 A stronger combination lemma was obtained by Edelsbrunner et al. [56] for the case of line segments. They proved that the total complexity of the purple regions E_i's is bounded by $r+b+O(u+v+k)$, where u (resp. v) is the total number of connected components of the boundaries of the red (resp. blue) faces. Recently, Har-Peled [75] generalized the combination lemma to the overlay of more than two arrangements.

The combination lemma implies that the complexity of all the 'purple' regions in the overlay of the faces f_1 and f_2 is $O(r+b) = O(\lambda_{s+2}(n))$. Exploiting this bound, Guibas et al. [72] describe a somewhat involved sweep-line algorithm that sweeps over f_1 and f_2, and computes the purple regions in time $O(\lambda_{s+2}(n) \log n)$. The main idea behind this sweep is that it is performed separately, but simultaneously, over the red, blue, and purple arrangements, in a manner that processes only a small number of red-blue intersections. See [72, 139] for more details. Hence, the overall running time of the algorithm is $O(\lambda_{s+2}(n) \log^2 n)$.

Recently, Amato et al. [17] have succeeded in derandomizing the algorithm by Chazelle et al. [34], described above, for a set of segments. The worst-case running time of their algorithm is $O(n\alpha^2(n)\log n)$. Hence, we can conclude the following.

Theorem 6.6 ([17, 72]) *Given a collection Γ of n Jordan arcs, each pair of which intersect in at most s points, and a point x not lying on any arc, the face of $\mathcal{A}(\Gamma)$ containing x can be computed by a deterministic algorithm in time $O(\lambda_{s+2}(n)\log^2 n)$, in an appropriate model of computation. The running time improves to $O(\lambda_s(n)\log^2 n)$ for collections of Jordan curves (closed or unbounded), and to $O(n\alpha^2(n)\log n)$ for collections of line segments.*

We conclude this subsection by mentioning two open problems.

Open Problem 3 *(i) Given a set Γ of n segments and a point p, can the face in $\mathcal{A}(\Gamma)$ containing p be computed in time $O(n\log h)$, where h is the number of edges in the face?*

(ii) Given a set Γ of n Jordan arcs, each pair of which intersects in at most s points, and a point p, can the face in $\mathcal{A}(\Gamma)$ containing p be computed in time $O(\lambda_{s+2}\log n)$?

6.3 Zones

The *zone* of a curve γ_0 in the arrangement $\mathcal{A}(\Gamma)$ of a collection Γ of n Jordan arcs is the set of all faces of $\mathcal{A}(\Gamma)$ that γ_0 intersects. The complexity of the zone is the sum of the complexities of all the faces in the zone.

Zones were initially studied for arrangements of lines and hyperplanes [52, 58, 59], but they are also easy to analyze in the context of general arcs. The following theorem demonstrates a close relationship between zones and faces in an arrangement.

Theorem 6.7 ([54]) *The complexity of the zone of a curve γ_0 in an arrangement $\mathcal{A}(\Gamma)$ of n Jordan arcs, each pair of which intersect in at most s points, is $O(\lambda_{s+2}(n))$, assuming that γ_0 intersects every arc of Γ in at most some constant number of points.*

Proof: Split every arc $\gamma \in \Gamma$ into two subarcs at each intersection point of γ and γ_0, and leave sufficiently small gaps between these pieces. In this manner all faces in the zone of γ_0 are merged into one face, at the cost of increasing the number of arcs from n to $O(n)$. Now we can apply Theorem 6.2 to conclude the proof. □

If Γ is a set of n lines and γ_0 is also a line, then after splitting each line of Γ at their intersection points with γ_0 we obtain a collection of $2n$ rays, and therefore the complexity of the unbounded face is $O(n)$. In fact, in this case one can show that the edges of the zone form a $DS(4n, 2)$ sequence, thereby obtaining an upper bound of $8n - 1$ on the complexity of the zone. Applying a more careful analysis, Bern et al. [30] proved the following theorem.

Theorem 6.8 ([30]) *The complexity of the zone of a line in an arrangement of n lines is at most $5.5n$, and this bound is tight within an additive constant term, in the worst case.*

See [13, 30, 36, 54] for other results and applications of zones of arcs.

An immediate consequence of Theorem 6.7 is an efficient algorithm for computing the arrangement $\mathcal{A}(\Gamma)$. Suppose we add the arcs of Γ one by one and maintain the arrangement of the arcs added so far. Let Γ_i be the set of arcs added in the first i stages, and let γ_{i+1} be the next arc to be added. Then in the $(i+1)$st stage one has to update only those faces of $\mathcal{A}(\Gamma_i)$ which lie in the zone of γ_{i+1}, and this can easily be done in time proportional to the complexity of the zone; see Edelsbrunner et al. [54] for details. By Theorem 6.7, the total running time of the algorithm is $O(n\lambda_{s+2}(n))$, and, by Theorem 6.8, the arrangement of a set of n lines can be computed in $O(n^2)$ time. If the arcs of Γ are added in a *random* order, then the expected running time of the above algorithm is $O(n \log n + k)$, where k is the number of vertices in $\mathcal{A}(\Gamma)$ [34, 45, 112], which is at most quadratic in n. The latter time bound is worst-case optimal.

Theorem 6.7 can also be used to obtain an upper bound on the complexity of any m faces of $\mathcal{A}(\Gamma)$. Specifically, let $\{f_1, \ldots, f_m\}$ be a subset of m distinct faces in $\mathcal{A}(\Gamma)$, and let n_f denote the number of vertices in a face f of $\mathcal{A}(\Gamma)$. Then, using the Cauchy-Schwarz inequality,

$$\begin{aligned}
\sum_{i=1}^{m} n_{f_i} &\leq m^{1/2}\Big(\sum_i n_{f_i}^2\Big)^{1/2} \leq m^{1/2}\Big(\sum_{f\in\mathcal{A}(\Gamma)} n_f^2\Big)^{1/2} \\
&= O\left[m^{1/2}\Big(\sum_{f\in\mathcal{A}(\Gamma)} n_f \lambda_{s+2}(k_f)\Big)^{1/2}\right] \\
&= O\left[m^{1/2}\left(\frac{\lambda_{s+2}(n)}{n}\right)^{1/2}\Big(\sum_{f\in\mathcal{A}(\Gamma)} n_f k_f\Big)^{1/2}\right],
\end{aligned}$$

where k_f is the number of arcs in Γ that appear along the boundary of f. It is easily verified that

$$\sum_{f\in\mathcal{A}(\Gamma)} n_f k_f \leq \sum_{\gamma\in\Gamma} \sum_{f\in zone(\gamma,\Gamma\setminus\{\gamma\})} n_f = O(n\lambda_{s+2}(n))\,.$$

Hence, we obtain the following result.

Theorem 6.9 ([54, 75]) *Let Γ be a set of n arcs satisfying the conditions stated earlier. The maximum number of edges bounding any m distinct faces of $\mathcal{A}(\Gamma)$ is $O(m^{1/2}\lambda_{s+2}(n))$.*

It should be noted that Theorem 6.9 is weaker than the best bounds known for the complexity of m distinct faces in arrangements of several special types of arcs, such as lines, segments, and circles (see [20, 32, 42]), but it applies to arrangements of more general arcs.

6.4 Levels in arrangements

Let Γ be a set of n x-monotone, unbounded Jordan curves, each pair of which intersects in at most s points. The *level* of a point $p \in \mathbb{R}^2$ in $\mathcal{A}(\Gamma)$ is the number of curves of Γ lying strictly below p, and the level of an edge $e \in \mathcal{A}(\Gamma)$ is the common level of all the points lying in the relative interior of e. For a nonnegative integer $k < n$, the *k-level* (respectively, *($\leq k$)-level*) of $\mathcal{A}(\Gamma)$ is (the closure of) the union of all edges in $\mathcal{A}(\Gamma)$ whose level is k (respectively, at most k). Note that the graph of the lower envelope E_Γ is the 0-level, so the complexity of the 0-level is at most $\lambda_s(n)$. Very little is known about the complexity of an arbitrary level, even for arrangements of lines, and there is a big gap between the known upper and lower bounds. (The so-called *k-set* problem, of obtaining sharp bounds on the complexity of an arbitrary level, is one of the most challenging open problems in combinatorial geometry; see [50, 64, 101, 115, 147].) However, tight bounds are known for the complexity of ($\leq k$)-levels in arrangements of curves:

Theorem 6.10 ([45, 137]) *Let Γ be a set of n x-monotone curves, each pair intersecting in at most s points, and let $0 < k < n$ be an integer. The number of edges in $\mathcal{A}(\Gamma)$ of level at most k is $O(k^2\lambda_s(\lfloor n/k \rfloor))$, and this bound is tight in the worst case.*

The proof of the theorem is based on an elegant probabilistic analysis technique, due to Clarkson and Shor [45], which has been applied to a variety of other problems as well. An immediate corollary of the above theorem is the following claim.

Corollary 6.11 ([15]) *The number of edges in the ($\leq k$)-level of an arrangement of n lines in the plane is $\Theta(nk)$.*

Corollary 6.11 can be extended to arrangements of hyperplanes in higher dimensions as well, where the number of vertices in ($\leq k$)-level is $\Theta(n^{\lfloor d/2 \rfloor}k^{\lceil d/2 \rceil})$ [45]. Efficient algorithms for computing ($\leq k$)-levels in arrangements are given in [5, 65, 111].

Theorem 6.10 can be extended to a more general setting. Let $\mathcal{K} = \{K_1, \ldots, K_n\}$ be a collection of n regions in $\mathbb{R}^2$ so that the boundaries of any two them intersect in at most s points. Let $f(r)$ denote the expected number of vertices on the boundary of the union of a random subset of r regions of $\mathcal{K}$. For example, if the boundary of each K_i is an x-monotone curve, then $f(r) = O(\lambda_s(r))$ (see Corollary 2.3). Using the same probabilistic technique of [45], the following theorem can be proved.

Theorem 6.12 ([137]) *Let $\mathcal{K} = \{K_1, \ldots, K_n\}$ be a collection of n regions in $\mathbb{R}^2$ so that the boundaries of any two them intersect in at most s points. For any integer $1 \leq k \leq n-2$, the number of intersection points of the boundaries of regions in $\mathcal{K}$ that lie in the interior of at most k regions of $\mathcal{K}$ is $O(k^2 f(\lfloor n/k \rfloor))$. If each ∂K_i is an x-monotone curve, then the number of such vertices is $O(k^2\lambda_s(\lfloor n/k \rfloor))$.*

Open Problem 4 *Obtain a tight bound on the complexity of a single level in an arrangements of lines in the plane.*

7 Miscellaneous Applications

In the previous section we presented applications of Davenport–Schinzel sequences to planar arrangements, but the scope of geometric applications of these sequences is much wider. It is beyond the scope of this survey chapter to present all these applications in detail, as they are quite diverse and require rather sophisticated and problem-specific geometric machinery. Instead, we briefly review here as many applications as space allows us, and provide more details for a few of them. More details, and additional applications, can be found in [139].

7.1 Applications of $DS(n, 2)$-sequences

Without having made it explicit, we have already encountered some combinatorial applications of $DS(n, 2)$-sequences in the previous sections (e.g., the analysis of the complexity of the zone of a line in an arrangement of lines). Here we present a few additional applications. See also Edelsbrunner and Guibas [55] and Ramos [122] for additional applications of this kind.

Voronoi diagrams. Let $S = \{p_1, \ldots, p_n\}$ be a set of n points in the plane. The *Voronoi diagram* of S, denoted as $Vor(S)$, under the Euclidean metric ρ, is a subdivision of the plane into cells $V(p_i)$, for $p_i \in S$, where $V(p_i) = \{x \in \mathbb{R}^2 \mid \rho(x, p_i) \leq \rho(x, p_j), \text{ for } 1 \leq j \leq n\}$. See [23, 99] for comprehensive surveys on Voronoi diagrams. Fortune [66] showed that $Vor(S)$ can be computed efficiently by sweeping the xy-plane from bottom to top with a horizontal line $\ell(t) : y = t$ (i.e., by varying t from $-\infty$ to $+\infty$). The basic idea of his algorithm is as follows.

For a point $p_i = (x_i, y_i) \in S$, let

$$C_i = \{(x, y, z) \mid (x - x_i)^2 + (y - y_i)^2 = z^2; z \geq 0\};$$

this is a cone in 3-space. Let $\mathcal{C} = \{C_i \mid 1 \leq i \leq n\}$. It is easily checked that, by definition, $Vor(S)$ is the minimization diagram of $\mathcal{C}$ (regarding $\mathcal{C}$ as a set of graphs of bivariate functions). The algorithm actually sweeps a slanted plane $h(t) : y + z = t$ across 3-space, varying t from $-\infty$ to $+\infty$, and maintains the xy-projection $M(t)$ of the cross section $h(t) \cap E_{\mathcal{C}}$, where $E_{\mathcal{C}}$ is the lower envelope of $\mathcal{C}$. The projection $\pi_i(t)$ of the intersection of $h(t)$ with the cone C_i is nonempty if and only if $t \geq y_i$, and is then a parabola with directrix $\ell(t)$ and focus p_i. Then $M(t)$ is easily seen to be the lower envelope of the parabolas π_i in the xy-plane. Since any two such parabolas intersect in at most 2

points, Theorem 3.1 implies that $M(t)$ has at most $2n-1$ breakpoints. Fortune proved that, as the value of t varies, the combinatorial structure of $M(t)$ changes at $O(n)$ *critical* values of t, and that $M(t)$ can be updated in $O(\log n)$ time at each critical value of t. Putting all these observations together, he obtained an optimal $O(n \log n)$-time algorithm for computing $Vor(S)$. See [66, 139] for further details.

Triangulation of convex polygons. A $DS(n,2)$-sequence is called *canonical* if its length is $2n-1$ and if its symbols are numbered so that the leftmost appearance of i precedes the leftmost appearance of j whenever $i < j$. Roselle [127] has shown that there exists a close relationship between triangulations of convex polygons and canonical $DS(n,2)$-sequences. A triangulation T of a convex $(n+1)$-gon P, whose vertices are labeled $1, 2, \ldots, n+1$ in counterclockwise order, can be represented by the set T^* consisting of all the diagonals (i,j), with $i < j$, that form T, and of the sides $(i, i+1)$, for $1 \le i \le n$, and $(1, n+1)$, of P. For each vertex i of P, let $\xi(i)$ be the sequence of all vertices $\langle j_1, \ldots, j_{k_i} \rangle$, arranged in *decreasing* (i.e., clockwise) order, such that $j_l < i$ and $(j_l, i) \in T^*$, for each $1 \le l \le k_i$. Let $\varphi(T)$ denote the sequence

$$\varphi(T) \;=\; \xi(2) \,\|\, \xi(3) \,\|\, \cdots \,\|\, \xi(n+1).$$

Theorem 7.1 ([127]) *Let T be a triangulation of a convex $(n+1)$-gon. Then the sequence $\varphi(T)$ is a canonical $DS(n,2)$-sequence. Conversely, every canonical $DS(n,2)$-sequence is the image $\varphi(T)$ of some triangulation T of a convex $(n+1)$-gon.*

It is easily verified that a convex $(n+1)$-gon can be triangulated in $\binom{2n-2}{n-1}/n$ different ways, so Theorem 7.1 implies that this is also the number of distinct canonical $DS(n,2)$-sequences; see also [109]. There are several other combinatorial structures that are equivalent to canonical $DS(n,2)$-sequences, including certain rooted plane trees and bracketing a formal product [95]. See [69, 95] for other results on enumeration of $DS(n,2)$-sequences.

7.2 Motion planning

In this subsection we describe several applications of Davenport–Schinzel sequences to algorithmic motion planning in robotics. A typical motion-planning problem can be defined as follows: we are given a robot system B with k degrees of freedom and an environment filled with obstacles. The *configuration space* of B is a k-dimensional parametric space, each point of which represents a possible placement of B by a k-tuple of real numbers that gives the values of the parameters controlling the k degrees of freedom of B. As an example of such a configuration space, consider the case where B is a rigid polygon moving (by translations and rotations) in the plane. Here B has three degrees of freedom, and any placement of B can be represented by the triple (x, y, θ), where (x, y) are the coordinates of some fixed reference point attached to B, and θ is the orientation of B. If we allow B only to translate,

it has only two degrees of freedom, and its placements can be represented by the pair (x, y) of the coordinates of the reference point.

The presence of obstacles in the robot's environment causes portions of the configuration space of B to be 'forbidden' or non-free. A placement of B is called *free* if B does not intersect any obstacle at that placement; otherwise it is called *non-free*. A non-free placement π is called *semi-free* if B does not intersect the interior of any obstacle at π. Our goal is to compute the *free configuration space* of B, which we denote by FP, consisting of all free placements of B. The boundary of FP consists of semi-free placements of B. The motion-planning problem that we consider is to determine, for any given pair of free placements, Z_1, Z_2, of B, whether there exists a continuous obstacle-avoiding motion of B from Z_1 to Z_2, and, if so, to plan such a motion.

This problem, under reasonable assumptions concerning the geometry of B and of the obstacles, can be re-stated as the problem of computing the connected components of FP, and of representing them in an appropriate discrete combinatorial fashion. This follows from the observation that a collision-free motion of B is a connected arc in FP, and such an arc connects Z_1 and Z_2 if and only if they lie in the same (arcwise-) connected component of FP.

The space FP can be defined in terms of an arrangement of surfaces within the configuration space, as follows. For each obstacle feature w (e.g., an obstacle corner, boundary edge, face, etc.) and each robot feature s, let $\sigma_{w,s}$ denote the locus of all placements of B at which s makes contact with w. Under reasonable assumptions concerning the shape of the robot and of the obstacles, the possible types of degrees of freedom of B, and an appropriate choice of the features w, s, we can assume that each locus $\sigma_{w,s}$ is a (portion of some) $(k-1)$-dimensional algebraic surface of bounded degree. Let Σ denote the resulting collection of surface patches $\sigma_{w,s}$. We refer to these surfaces as *contact surfaces*, and let n denote their number. Let $Z \in FP$ be some initial free placement of B. As we move B from Z, it will remain free as long as the corresponding path traced in the configuration space does not reach any contact surface. The free configuration space of B is therefore a collection of some of the cells of the arrangement $\mathcal{A}(\Sigma)$ of the contact surfaces. Moreover, if we only want to compute the portion of FP that consists of all free placements reachable from a fixed initial free placement Z of B, then this portion is the cell of $\mathcal{A}(\Sigma)$ that contains Z.

Hence, the problem has been reduced to the problem of computing a single cell in an arrangement of a collection Σ of n algebraic surface patches, of low bounded degree, in $\mathbb{R}^k$. In a companion chapter in this volume, we will return to this problem when we discuss higher-dimensional arrangements. For the time being, let us consider only the case $k = 2$, where we regard the configuration space of B as a planar region, and the problem becomes that of computing a single face in a planar arrangement of n low-degree algebraic arcs. By the results of Section 6 we immediately conclude:

Theorem 7.2 ([72]) *With the above notation, the combinatorial complexity of the space*

C of all free placements of a general robot system B with two degrees of freedom, which are reachable from Z by a collision-free motion, is $O(\lambda_{s+2}(n))$, where n is the number of contact arcs and s is the maximum number of intersections between any pair of these arcs. Moreover, C can be constructed, in an appropriate model of computation, in $O(\lambda_{s+2}(n)\log^2 n)$ deterministic time, or in $O(\lambda_{s+2}(n)\log n)$ randomized expected time.

Note that, once C is available, a path that connects Z to some desired target placement (that also lies in C) can easily be computed in time $O(\lambda_{s+2}(n))$.

Let us also consider a special case of this result, in which B is an arbitrary rigid polygonal object with p edges, translating (but not rotating) in a polygonal environment Q whose boundary consists of q edges. As noted above, any placement of B can be represented by the position (x, y) of some fixed reference point P attached to B. For each obstacle edge w and each vertex v of B, let $\gamma_{w,v}$ denote the locus of all placements of B at which v touches w; clearly, this is a line segment obtained by an appropriate translation of w. Similarly, for each obstacle corner c and each side e of B, let $\gamma_{c,e}$ denote the locus of all placements of B at which e touches c; this is also an appropriately translated copy of e. Hence, in this simple instance, the contact loci are $O(pq)$ straight segments in the plane, and the set of all free placements reachable from a given initial free placement Z of B is the face containing Z in the arrangement formed by these $O(pq)$ contact segments. The analysis of Section 6 thus immediately implies that the combinatorial complexity of the space C of all free placements of B that are reachable from Z by a collision-free translational motion is $O(pq\alpha(pq))$. This bound was recently improved by Har-Peled et al. [76] to $O(pq\alpha(p))$. Hence, we obtain the following result.

Theorem 7.3 ([76]) *With the above notations, the combinatorial complexity of the space C of all free placements of the translating convex polygon B that are reachable from Z by a collision-free translational motion is $O(pq\alpha(p))$. Moreover, C can be constructed in $O(pq\alpha^2(p)\log pq)$ deterministic time, or in $O(pq\alpha(p)\log pq)$ randomized expected time.*

Note that the above analysis holds for any arbitrary polygonal region B. In fact, B does not even have to be connected, and may consist of several disjoint pieces, all translating rigidly together. However, if B is a single *convex* polygon, better results can be obtained, which we mention here, for the sake of completeness. Suppose that in this case the obstacles consist of m convex polygons with pairwise-disjoint interiors (non-convex obstacles are assumed to be cut into convex pieces). For each convex obstacle O, let γ_O denote the locus of all placements of B at which it touches O (but their interiors remain disjoint). As is well known (see, e.g., [87]), γ_O is (the boundary of) a closed convex polygon, which is the *Minkowski sum* $O \oplus (-B_0) = \{x - y \mid x \in O, y \in B_0\}$, where B_0 is a standard placement of B at which the reference point lies at the origin. As is also well known, the number of edges of γ_O is at most $k + n_O$, where n_O is the number of edges bounding O. It has been shown in [87] that, for any pair of distinct obstacles O, O' (with pairwise-disjoint interiors), the

polygons γ_O, $\gamma_{O'}$ intersect in at most two points, assuming general position of B and of the obstacles. Hence, applying Theorem 3.1 we conclude that the space C of free placements of B reachable from Z has complexity at most $\lambda_2(m) = 2m - 1$. Note, however, that here we measure complexity only in terms of the number of intersections of the loci γ_O that appear along ∂C; to this we have to add the total number of vertices of the individual polygons γ_O, which is at most $\sum_O (k + n_O) = km + n$. Hence, we conclude that the boundary of the desired free component C contains at most $km + n$ *reflex* corners and at most $2m - 1$ non-reflex (convex) corners. Moreover, as shown in [87], the entire free configuration space of B (which is simply the complement of the union of the polygons γ_O) has at most $6m - 12$ non-reflex corners (for $m > 2$) and at most $km + n$ reflex corners (see also Section 7.8).

For systems with more than two degrees of freedom, the situation is more involved (and will be mostly delegated to the companion paper). However, there are certain special motion-planning problems which admit a more direct analysis of their combinatorial and algorithmic properties, in which Davenport–Schinzel sequences are explicitly used.

One such problem is that of planning the motion of a *convex* polygonal robot B translating and rotating in a planar polygonal environment Q, as above. Since B has three degrees of freedom, any vertex of FP is a 'semi-free' placement of B at which it makes three simultaneous contacts with obstacles, while otherwise remaining free. We refer to such placements as *critical placements* of B. Leven and Sharir [100] proved that the number of critical placements of B is $O(pq\lambda_6(pq))$. This is done by reducing the problem to the interaction of $O(pq)$ upper and lower envelopes of collections of univariate partially defined functions of the orientation θ, so that each collection consists of $O(pq)$ functions, and each pair of functions intersect in at most four points; see [88, 100] for more details. This implies that the combinatorial complexity of the *entire* free configuration space of B is also $O(pq\lambda_6(pq))$. (To appreciate this bound, we note that if B is non-convex, its entire free configuration space can be shown to have $\Omega((pq)^3)$ complexity [139].) This combinatorial bound has led to an efficient algorithm for constructing FP [88, 89], whose running time is $O(pq\lambda_6(pq)\log pq)$. Chew and Kedem [39] gave an $O(p^4q\lambda_3(q)\log q)$-time algorithm for finding a largest similar copy of B that can be placed inside Q without intersecting the interior of any obstacle, and also to plan a high-clearance motion of B inside Q. Later Sharir and Toledo [141] gave an $O(p^2q\lambda_6(pq)\log^3 pq \log\log pq)$-time algorithm for the largest placement problem, using the result of [100] and the so called parametric searching technique.

Some other results on motion-planning and related problems, which exploit the theory of Davenport–Schinzel sequences, can be found in [3, 6, 9, 83].

7.3 Shortest paths

Computing a collision-free shortest path between two points amidst a collection $\mathcal{O}$ of polyhedral obstacles in $\mathbb{R}^3$ is a fundamental problem in robotics (it is a special case of optimal motion planning). Canny and Reif [33] showed that the problem is NP-Hard, which has motivated the study of efficient construction of approximate shortest paths and of developing

polynomial-time algorithms for special cases of the problem.

Clarkson [43] presented a polynomial-time algorithm for constructing approximate shortest paths in 3-space (see [40, 116] for other such algorithms). For a given $\varepsilon > 0$, his algorithm constructs a graph G_ε, whose nodes are points in $\mathbb{R}^3$, and whose edges connect some pairs of these points by straight segments. The size of G_ε is $O\left(n^2 \log(n\rho) + n^2\lambda_s(n)/\varepsilon^4\right)$, where s is a fixed constant and ρ is the ratio of the length of the longest edge in $\mathcal{O}$ to the (straight) distance between p and q. He then reduces the problem to that of constructing a shortest path in G_ε (where the weight of an edge is its Euclidean length), and shows that the ratio between the length of the path obtained in this manner and the actual collision-free shortest path between p and q is at most $1 + \varepsilon$. The running time of his algorithm is $O\left((n^2\lambda_s(n)/\varepsilon^4)\log(n/\varepsilon) + n^2\log(n\rho)\log(n\log\rho)\right)$.

A special case of shortest paths in 3-space, which has been widely studied, is when $\mathcal{O}$ consists of a single convex polytope and p, q lie on its surface (see, for example, [37, 106, 140]). A shortest path on the surface of a convex polytope can be represented by the sequence of edges that it crosses, and we refer to such a sequence of edges as a *shortest-path edge sequence*. It is known that there are $\Theta(n^4)$ shortest path edge-sequences [108, 130]. Agarwal et al. [4] have shown that the exact set of all shortest-path edge sequences can be computed in time $O(n^5\lambda_s(n)\log n)$, for some constant $s > 0$, improving a previous algorithm by Schevon and O'Rourke [131].

Baltsan and Sharir [27] considered the special case where $\mathcal{O}$ consists of two disjoint convex polytopes (and p and q lie anywhere in the free space). Using Davenport–Schinzel sequences to bound the number of candidate paths that one has to consider, they presented an algorithm with running time $O(n^2\lambda_{10}(n)\log n)$ to find an exact collision-free shortest path between p and q.

If the moving object is not a point and the object is allowed to rotate, the problem of computing a shortest path becomes significantly more difficult, even in the planar case. (In fact, even the notion of shortest path becomes much vaguer now.) Suppose we want to compute an optimal path for moving a line segment $\gamma = pq$ (allowing both translations and rotations) amid polygonal obstacles with a total of n edges. Assume that the cost of a path is defined as the total distance traveled by one of its endpoints, say, p, and restrict the problem further by requiring that p moves along polygonal paths that can bend only at obstacle vertices. This rather restricted version of the problem was studied by Papadimitriou and Silverberg [117], who gave an $O(n^4 \log n)$-time algorithm for computing a shortest path in the above setting. Sharir [136] improved the running time to $O(n^3\alpha(n)\log^2 n)$, using Davenport–Schinzel sequences and planar arrangements.

7.4 Transversals of planar objects

Let $\mathcal{S} = \{ S_1, S_2, \ldots, S_n \}$ be a collection of n compact convex sets in the plane. A line that intersects all sets of $\mathcal{S}$ is called a *transversal* (or a *stabber*) of $\mathcal{S}$. Note that a line intersects

a set if and only if it intersects its convex hull, so convexity is usually not a real restriction. For each set $S_i \in \mathcal{S}$, let S_i^* denote the set of points dual to the lines that intersect S_i, using a standard duality transform [52]. S_i^* is bounded from above by a convex x-monotone curve A_i and from below by a concave x-monotone curve B_i; see Figure 9. The *stabbing region* of $\mathcal{S}$ (or the *space of all transversals*) is the intersection $\mathcal{S}^* = \bigcap_{i=1}^n S_i^*$. By definition, $\mathcal{S}^*$ is the set of points dual to all nonvertical transversals of $\mathcal{S}$ [22, 57].

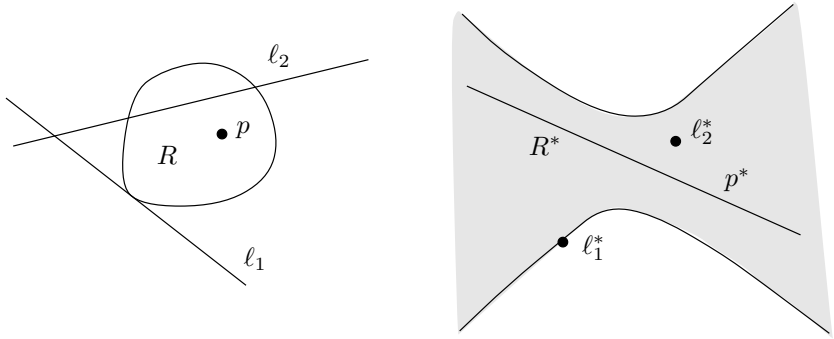

FIGURE 9. A convex set R and its stabbing region R^*.

The complexity of $\mathcal{S}^*$ can be measured by the number of its vertices, where a vertex is an intersection point between the boundaries of two regions S_i^* and S_j^* that lies on $\partial\mathcal{S}^*$. Since $\mathcal{S}^*$ is the region lying between the lower envelope of the set $\mathcal{A} = \{A_i \mid 1 \le i \le n\}$ and the upper envelope of the set $\mathcal{B} = \{B_i \mid 1 \le i \le n\}$, Corollary 2.3 implies that the complexity of $\mathcal{S}^*$ is $O(\lambda_s(n))$, and that it can be computed in time $O(\lambda_s(n)\log n)$, where s is the maximum number of common upper tangents or of common lower tangents between any two objects of $\mathcal{S}$ [22]. If $\mathcal{S}$ is a set of n convex polygons with a total of m vertices, then A and B are sets of n piecewise-linear curves, with a total of m segments, so Corollary 3.4 implies that the complexity of $\mathcal{S}^*$ is $O(m\alpha(n))$, and one can also show that $\mathcal{S}^*$ can be computed in time $O(m \log n)$.

For any $0 \le k < n$, a line ℓ is called a *k-transversal* if it intersects at least $n-k$ objects of $\mathcal{S}$. If we define K_{2i-1} to be the region lying above A_i and K_{2i} to be the region lying below B_i, then the point dual to ℓ lies in the interior of at most k K_j's. Using Theorem 6.12, Sharir showed that the complexity of the space of all k-transversals of $\mathcal{S}$ is $O((k+1)^2\lambda_s(\lfloor n/(k+1)\rfloor))$ [137, 139].

Next, suppose that $\mathcal{S}$ is a set of pairwise disjoint convex objects. We say that a directed transversal $\vec{\ell}$ induces a *geometric permutation* $\langle i_1, i_2, \ldots, i_n\rangle$ of $\mathcal{S}$ if $\vec{\ell}$ intersects the objects of $\mathcal{S}$ in the order $\langle S_{i_1}, \ldots, S_{i_n}\rangle$. A directed transversal $\vec{\ell}$ is called *extremal* if it is tangent to two objects S_{i_1}, S_{i_2} and these objects are contained in the closed halfplane that lies to the left of $\vec{\ell}$. Edelsbrunner and Sharir [60] proved that, there is an extremal transversal inducing each geometric permutation. Let $U = \langle\theta_1, \ldots, \theta_m\rangle$ be the (circular) sequence of all orientations, sorted in clockwise order, of the extremal directed transversals of $\mathcal{S}$. It is

shown in [60] that U can be mapped to a $DS(n, 2)$-cycle, which implies that the number of geometric permutations of $\mathcal{S}$ is at most $2n-2$. This bound is known to be tight in the worst case; see [25, 84]. Other results on geometric permutations can be found in [24, 26, 63], and see [70] for a comprehensive survey on geometric transversal theory.

7.5 Dynamic geometry

In this section we consider various problems related to a set of points in the plane, each moving along some predefined trajectory. As mentioned in the introduction, Atallah's paper [21] on dynamic geometry problems was the first paper in computational geometry to apply Davenport–Schinzel sequences.

We assume that we are given a collection $S = \{p_1, \ldots, p_n\}$ of n points in the plane such that the coordinates of each p_i are functions of time. Let $p_i(t) = (x_i(t), y_i(t))$ denote the position of the point p_i at time t, and let $S(t)$ denote the configuration of S at time t. We assume that $x_i(t), y_i(t)$, for $1 \leq i \leq n$, are polynomials of maximum degree s, for some constant s, or similarly well-behaved functions. We want to study how various geometric structures defined by S change with time.

We first bound the number of times the nearest neighbor of some point p_i of S changes. For every $j \neq i$, let

$$g_{ij}(t) = d^2(p_i(t), p_j(t)) = (x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2,$$

and let $G_i(t) = \min_{j \neq i} g_{ij}(t)$. By definition, the nearest neighbor of $p_i(t)$ changes only at breakpoints of G_i. Since each g_{ij} is a polynomial of degree at most $2s$, G_i has at most $\lambda_{2s}(n)$ breakpoints. This bound is almost tight, because the nearest neighbor of a point can change $\Omega(n)$ times, in the worst case. The same technique can also be used to bound the number of times at which the closest pair of points in $S(t)$ changes. For this, consider the function $G(t) = \min_{i<j} g_{ij}(t)$. Again, by definition, the closest pair in $S(t)$ changes only at breakpoints of $G(t)$, and $G(t)$ has at most $\lambda_{2s}(\binom{n}{2}) \leq n\lambda_{2s}(n)$ breakpoints. This bound can also be shown to be close to optimal in the worst case. Using a more involved argument, Atallah [21] showed that the number of combinatorial changes in the convex hull of S is $O(n\lambda_{2s+2}(n))$. An $\Omega(n^2)$ lower bound on the number of combinatorial changes is proved in [7].

A more interesting and harder problem in this area is to bound the number of changes in the *Delaunay triangulation* of S (the dual of the Voronoi diagram of S; see [52]), which was left as an open problem in [21]. The best known upper bound is $O(n^2\lambda_s(n))$ [67, 71, 83], and the best known lower bound is $\Omega(n^2)$. If each point of S is moving with unit speed along a line, then the upper bound on the number of changes in the Delaunay triangulation can be improved to $O(n^3)$; see, e.g., Guibas et al. [71]. It is conjectured that the actual bound is $O(n\lambda_r(n))$, for some r depending on s, even when the points are moving at different speeds. Chew [38] showed that if the underlying metric for the Delaunay triangulation is L_1 or

L_∞, instead of the Euclidean metric, the number of combinatorial changes in the Delaunay triangulation of a set of n points moving in the plane, each with a constant velocity (along some line), is only $O(n^2\alpha(n))$, thus establishing the conjecture in this special case. Aonuma et al. [18] have shown that, given k sets of points in the plane, each consisting of n points and moving rigidly according to some continuous function of time, the number of combinatorial changes in the Voronoi diagram of these kn points is $O(k^4 n\lambda_s(n))$, for an appropriate constant s. The bound was improved by Huttenlocher et al. [80] to $O(n^2k^2\lambda_s(k))$. See [12, 124, 125, 126] for bounds on dynamic Voronoi diagrams in higher dimensions, and [73, 82, 85, 104, 107, 113, 147] for other results in dynamic geometry.

Open Problem 5 *Obtain a tight bound on the number of changes in the Delaunay triangulation of a set of moving points, each of which is moving with a fixed velocity.*

7.6 Hausdorff distance and Voronoi surfaces

Let S be a set of points in $\mathbb{R}^2$. The *Voronoi surface* of S is defined as the bivariate function $\pi(x) = \min_{q\in S}\ \rho(q, x)$, for $x \in \mathbb{R}^2$, where $\rho(\cdot, \cdot)$ is the distance function in $\mathbb{R}^2$, which we will assume to be some fixed L_p metric. By definition, the orthogonal projection of the graph of π onto the xy-plane is the Voronoi diagram of S under the metric ρ.

Let $S_1, \ldots, S_m$ be a family of pairwise-disjoint point sets in the plane, with $|S_i| = n_i$ and $\sum_{i=1}^m n_i = n$, and let $\pi_i(x)$ denote the Voronoi surface of S_i. The upper envelope of these surfaces is the (graph of the) function $\Pi(x) = \max_{1\le i\le m} \pi_i(x)$. Thus $\Pi(x)$ gives the largest distance from x to its m nearest neighbors, one from each set S_i. For a point $q \in S$, let $V(q) \subseteq \mathbb{R}^2$ be the set of points at which the function $\rho(q, x)$ attains Π. If $q \in S_i$ then, by definition, $V(q)$ is contained in the Voronoi cell $V_i(q)$ of q in $Vor(S_i)$. Moreover, if we denote by $V_{ij}(q)$ the Voronoi cell of q in $Vor(S_i \cup S_j)$, for any $j \neq i$, then $V(q) = V_i(q) \setminus \bigcup_{j\neq i} V_{ij}(q)$. Since each Voronoi cell $V_{ij}(q)$ is star-shaped with respect to q [98], we can interpret the boundary of $\bigcup_{j\neq i} V_{ij}(q)$ as the upper envelope of the boundaries of the cells $V_{ij}(q)$, each represented as a univariate function $r = f_{ij}(\theta)$, where (r, θ) are polar coordinates about q. Using these observations, Huttenlocher et al. [81] showed that the total number of vertices of the regions $V(q)$, summed over all points $q \in S$, is $O(mn\alpha(mn))$. For the L_1 and L_∞-metrics, the number of vertices is only $O(mn)$. Moreover, Π can be computed in time $O(mn \log mn)$ under any L_p metric.

The study of the upper envelope of Voronoi surfaces is motivated by the following application. Let A and B be two sets of points in the plane. The *Hausdorff distance* between A and B is defined as

$$H(A, B) = \max\{h(A, B), h(B, A)\},$$

where $h(A, B) = \max_{a\in A}\ \min_{b\in B}\ \rho(a, b)$, and $\rho(\cdot, \cdot)$ is, say, some L_p-metric. Suppose we fix the set A and allow B to translate; then $D(A, B)$, the *minimum Hausdorff distance under*

translation between A and B is defined as

$$D(A,B) = \min_x H(A, B \oplus x),$$

where $B \oplus x = \{b + x \mid x \in B\}$. The function $D(A,B)$ is used as a measure of the degree to which the two sets resemble each other. Huttenlocher et al. [81] showed that $D(A,B)$ can be computed by constructing the upper envelope of a family of Voronoi surfaces of $2n$ point sets, each consisting of n points. Hence, $D(A,B)$ can be computed in time $O(n^3 \log n)$. See also [79, 80] for some related results.

The algorithm of [81] can be extended to compute the minimum Hausdorff distance $D(A,B)$ for sets A, B of nonintersecting segments under the L_1 or L_∞-metric. Alt et al. [16] presented an $O(n^7 \log n)$-time algorithm for computing $D(A,B)$ for sets of nonintersecting segments under the L_2 metric, which has been improved by Agarwal et al. [11]. If we allow both translations and rotations, the problem of computing a placement that minimizes the Hausdorff distance becomes considerably more difficult. See [11, 61] for efficient approximate solutions.

7.7 Visibility problems

Let Σ be a *polyhedral terrain* (i.e., the graph of a continuous piecewise-linear, totally defined function $z = \Sigma(x,y)$) having n edges, and let ξ be a fixed point lying above Σ. Without loss of generality, we can assume that ξ lies on the z-axis. We wish to preprocess Σ into a data structure for answering *ray-shooting* queries from ξ, i.e., for a query ray ρ emanating from ξ, we wish to find efficiently the first intersection point of ρ with Σ. Cole and Sharir [46] presented a data structure of size $O(n\alpha(n)\log n)$ that can answer a query in time $O(\log n)$. We give a brief description of their data structure.

For any object w in 3-space, let $w^\star$ denote its projection onto the xy-plane. For technical reasons, we consider only those rays emerging from ξ into the halfspace $y > 0$, and so we consider only the portion of Σ lying in that half-space. The edges of (this portion of) Σ can be partially ordered so that $u < v$ for two edges u, v if there exists a horizontal ray from $\xi^\star$ intersecting both $u^\star, v^\star$ such that its intersection with $u^\star$ is nearer to $\xi^\star$. This order is extended to a total order $\mathcal{E} = \langle e_1 < e_2 < ... < e_n \rangle$. Next, for each edge e_i, we define a partial function $h_i : \mathbb{S} \to [0, 2\pi)$, where $\mathbb{S}$ is the unit circle of orientations, as follows. For each $\theta \in \mathbb{S}$, if the ray $r^\star_\theta$ in the xy-plane emanating from $\xi^\star$ in direction θ does not intersect $e^\star_i$, then $h_i(\theta)$ is undefined. Otherwise, let $p^\star$ be the intersection point of $r^\star_\theta$ and $e^\star_i$. We define $h_i(\theta)$ to be the polar angle (i.e., the angle with the z-axis) of the ray emanating from ξ and passing through p, where $p \in e_i$ is the point whose projection is $p^\star$. For each subset $E \subseteq \mathcal{E}$, we define

$$h_E(\theta) = \min_{e_i \in E} h_i(\theta)$$

to be the lower envelope of the h_i's. We call the graph of h_E the *upper rim* of E. Intuitively, the upper rim of E corresponds to the 'skyline' of E seen from ξ. Using $DS(n,3)$-sequences,

it is easily seen that h_E has $O(|E|\alpha(|E|))$ breakpoints.

We construct a balanced binary tree T that stores a collection of upper rims of subsets of edges of Σ. Let $E_1 = \{e_1, \ldots, e_{n/2}\}$ and let $E_2 = \{e_{n/2+1}, \ldots, e_n\}$. The root of T stores the upper rim h_{E_1} and the two subtrees of the root represent recursively the search structures for the "nearer" half E_1 and for the "farther" half E_2 of the edges of Σ. It is easily seen that the total storage (and preprocessing time) that T requires is $O(n\alpha(n)\log n)$. We can now process each ray-shooting query as follows. Let ρ be a given ray emerging from ξ in direction (θ, φ), where θ is the horizontal orientation of the ray and φ is its azimuth. We perform a binary search through T by first comparing ρ with the topmost upper rim h_{E_1}, stored at the root u of T. We compute, in $O(\log n)$ time, the value of $h_{E_1}(\theta)$. If $\varphi < h_{E_1}(\theta)$ then ρ lies above all the edges of E_1, and we continue the search through T at the right child of u; otherwise ρ must hit some face of Σ bounded on its far side by an edge of E_1, and we continue the search through T at the left child of u. When this search is completed, we will have found two edges $e_i < e_j$, necessarily bounding the same face f of Σ, such that ρ passes above e_i and below e_j, from which the first intersection point of ρ and Σ (which lies on f) can be calculated in constant time. The whole search takes $O(\log^2 n)$ time, which can be improved to $O(\log n)$ using fractional cascading [35].

Theorem 7.4 ([46]) *Given Σ and ξ as above, we can preprocess them in time $O(n\alpha(n)\log n)$ into a data structure of size $O(n\alpha(n)\log n)$ that supports $O(\log n)$-time ray-shooting queries from ξ.*

Cole and Sharir also showed, using $DS(n, 4)$-sequences, that the total number of combinatorially different views of Σ, as the view point ξ moves along the z-axis, is $O(n\lambda_4(n))$ (and can be $\Omega(n\lambda_3(n))$ in the worst case); see also Mulmuley [110]. Bern et al. [29] proposed a data structure of size $O(n\lambda_4(n))$ that can answer a ray-shooting query, for rays emanating from any point on the z-axis, in $O(\log n)$ time per query. The problem can also be extended to situations where the view point is allowed to vary in more general regions, but then the solutions require techniques that are related to higher-dimensional arrangements. This will be picked up in the companion chapter in this volume.

7.8 Union of Jordan regions

Let $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$ be a set of n closed or unbounded Jordan curves in the plane, and let K_i denote any one of the two regions bounded by γ_i. Let $K = \bigcup_{i=1}^n K_i$. We want to bound the combinatorial complexity of K, that is, the number of intersection points of arcs of Γ that appear on the boundary ∂K of K. As already mentioned in Section 7.2, Kedem et al. [87] have proved that if any two curves in Γ intersect in at most two points, then ∂K contains at most $6n - 12$ intersection points (provided $n \geq 3$), and that this bound is tight in the worst case. On the other hand, if pairs of curves in Γ can intersect in four or more points, then ∂K may contain $\Omega(n^2)$ intersection points in the worst case. This raises the

question of what happens if any two curves in Γ intersect in at most three points. Using $DS(n,3)$-sequences, Edelsbrunner et al. [53] have shown that the maximum combinatorial complexity of the union K is $\Theta(n\alpha(n))$ (here we need to assume that the curves in Γ are unbounded, because two closed Jordan curves, in general position, cannot intersect in three points). The lower bound follows from Theorem 5.3, and the upper bound requires a rather sophisticated analysis of the topological structure of K.

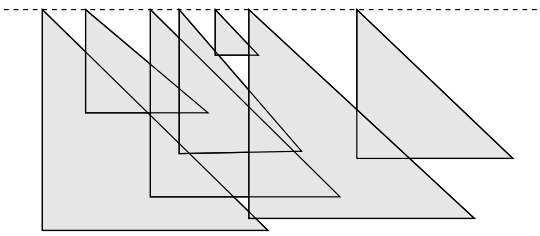

FIGURE 10. "Almost homothetic" right-angle triangles.

Consider next the case where Γ is the set of boundaries of n 'almost homothetic' right-angle triangles that satisfy the following conditions (see Figure 10): Each of the corresponding regions K_i is a triangle whose orthogonal sides are parallel to the x- and y-axes, whose right-angle vertex is the lowest-leftmost point of the triangle, whose top vertex lies on the x-axis, and whose hypotenuse has orientation in the range $[\frac{3\pi}{4} - \varepsilon, \frac{3\pi}{4} + \varepsilon]$, for some small constant $\varepsilon > 0$, Matoušek at al. [102] proved, using $DS(n,4)$-sequences, that the complexity of the union K is $O(\lambda_4(n))$. This bound for the above fairly restricted problem implies the main result of [102], that the complexity of the union of n 'fat' triangles, namely, triangles satisfying the property that each of their angles is at least some fixed constant angle, is $O(n \log\log n)$. Note that the union of n arbitrary triangles can have $\Omega(n^2)$ vertices in the worst case. See also [8, 86] for other results involving fat triangles that exploit Davenport–Schinzel sequences.

7.9 Extremal $\{0,1\}$-matrices.

Let $M = \{M_{ij}\}$ be an $m \times n$ matrix such that $M_{ij} \in \{0,1\}$ for all $1 \le i \le m$ and $1 \le j \le n$; we call M a $\{0,1\}$-*matrix*. A *configuration*

$$C = \{C_{ij} \mid 1 \le i \le u,\ 1 \le j \le v\}$$

is a $u \times v$ matrix with 1's and blanks as its entries. We say that M *does not contain* C if there is no submatrix A of M that contains the 1-entries of C, that is, there are no u rows $i_1 < i_2 < \cdots < i_u$ and v columns $j_1 < j_2 < \cdots < j_v$ such that $M_{i_s j_t} = 1$ for all (s,t) for which C_{st} is 1. An upper bound on the number of 1's in M for various C is useful in solving certain combinatorial problems in geometry, and also in bounding the time and storage

complexity of certain geometric algorithms; see, for example, Bienstock and Győri [31], Efrat and Sharir [62], and Pach and Sharir [114].

Let C_0 be the 2×4 configuration

$$\begin{pmatrix} 1 & & 1 & \\ & 1 & & 1 \end{pmatrix}.$$

By definition, a matrix M does not contain C_0 if there are no two rows $i_1 < i_2$ and four distinct columns $j_1 < j_2 < j_3 < j_4$ such that

$$M_{i_1 j_1} = M_{i_2 j_2} = M_{i_1 j_3} = M_{i_2 j_4} = 1. \tag{7.1}$$

Füredi and Hajnal [68] have shown that the number of 1's in a $\{0,1\}$-matrix M that does not contain C_0 is $\Theta(m\alpha(m)+n)$. They prove the upper bound by transforming M to a $DS(n,3)$-sequence, and they construct an $m \times n$ matrix, using a recursive scheme similar to the one used in Section 5, to prove the lower bound.

8 Concluding Remarks

In this chapter we surveyed the basic results on the Davenport–Schinzel sequences and some of their immediate applications to a variety of geometric problems. It is impossible to include all the results and all the applications in a survey paper. We refer the interested readers to [139] for details and for many other applications of Davenport–Schnizel sequences.

Finally, A natural question to ask is whether the theory of Davenport–Schinzel sequences can be extended to higher dimensions. Lower envelopes and minimization diagrams have a natural extension to higher dimensions, and they are discussed in a companion chapter in this volume. However, no purely combinatorial generalization of Davenport–Schinzel sequences to higher dimensions has been proposed so far.

References

[1] W. Ackermann, Zum Hilbertschen Aufbau der reellen Zahlen, *Mathematical Annals*, 99 (1928), 118–133.

[2] R. Adamec, M. Klazar, and P. Valtr, Generalized Davenport-Schinzel sequences with linear upper bound, *Discrete Math.*, 108 (1992), 219–229.

[3] P. K. Agarwal, N. Amenta, B. Aronov, and M. Sharir, Largest placements and motion planning of a convex polygon, *Algorithms for Robotic Motion and Manipulation* (J.-P. Laumond and M. Overmars, eds.), A.K. Peters, 1997, pp. 143–154.

[4] P. K. Agarwal, B. Aronov, J. O'Rourke, and C. A. Schevon, Star unfolding of a polytope with applications, *SIAM J. Comput.*, 26 (1997), 1679–1713.

[5] P. K. Agarwal, M. de Berg, J. Matoušek, and O. Schwarzkopf, Constructing levels in arrangements and higher order Voronoi diagrams, *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, 1994, pp. 67–75.

[6] P. K. Agarwal, A. Efrat, M. Sharir, and S. Toledo, Computing a segment center for a planar point set, *J. Algorithms*, 15 (1993), 314–323.

[7] P. K. Agarwal, L. J. Guibas, J. Hershberger, and E. Veach, Maintaining the extent of a moving point set, *Proc. 5th Workshop Algorithms Data Struct.*, *Lecture Notes Comput. Sci.*, Vol. 1272, Springer-Verlag, 1997, pp. 31–44.

[8] P. K. Agarwal, M. J. Katz, and M. Sharir, Computing depth orders and related problems, *Comput. Geom. Theory Appl.*, 5 (1995), 187–206.

[9] P. K. Agarwal and M. Sharir, Red-blue intersection detection algorithms, with applications to motion planning and collision detection, *SIAM J. Comput.*, 19 (1990), 297–321.

[10] P. K. Agarwal, M. Sharir, and P. Shor, Sharp upper and lower bounds on the length of general Davenport-Schinzel sequences, *J. Combin. Theory Ser. A*, 52 (1989), 228–274.

[11] P. K. Agarwal, M. Sharir, and S. Toledo, Applications of parametric searching in geometric optimization, *J. Algorithms*, 17 (1994), 292–318.

[12] G. Albers and T. Roos, Voronoi diagrams of moving points in higher dimensional spaces, *Proc. 3rd Scand. Workshop Algorithm Theory*, *Lecture Notes Comput. Sci.*, Vol. 621, Springer-Verlag, 1992, pp. 399–409.

[13] P. Alevizos, J. D. Boissonnat, and F. P. Preparata, An optimal algorithm for the boundary of a cell in a union of rays, *Algorithmica*, 5 (1990), 573–590.

[14] P. Alevizos, J. D. Boissonnat, and F. P. Preparata, An optimal algorithm for the boundary of a cell in a union of rays: Corrigendum, *Algorithmica*, 6 (1991), 292–293.

[15] N. Alon and E. Győri, The number of small semispaces of a finite set of points in the plane, *J. Combin. Theory Ser. A*, 41 (1986), 154–157.

[16] H. Alt, B. Behrends, and J. Blömer, Approximate matching of polygonal shapes, *Ann. Math. Artif. Intell.*, 13 (1995), 251–266.

[17] N. M. Amato, M. T. Goodrich, and E. A. Ramos, Computing faces in segment and simplex arrangements, *Proc. 27th Annu. ACM Sympos. Theory Comput.*, 1995, pp. 672–682.

[18] H. Aonuma, H. Imai, K. Imai, and T. Tokuyama, Maximin location of convex objects in a polygon and related dynamic Voronoi diagrams, *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, 1990, pp. 225–234.

[19] E. M. Arkin, D. Halperin, K. Kedem, J. S. B. Mitchell, and N. Naor, Arrangements of segments that share endpoints: Single face results, *Discrete Comput. Geom.*, 13 (1995), 257–270.

[20] B. Aronov, H. Edelsbrunner, L. J. Guibas, and M. Sharir, The number of edges of many faces in a line segment arrangement, *Combinatorica*, 12 (1992), 261–274.

[21] M. J. Atallah, Some dynamic computational geometry problems, *Comput. Math. Appl.*, 11 (1985), 1171–1181.

[22] M. J. Atallah and C. Bajaj, Efficient algorithms for common transversals, *Inform. Process. Lett.*, 25 (1987), 87–91.

[23] F. Aurenhammer, Voronoi diagrams: A survey of a fundamental geometric data structure, *ACM Comput. Surv.*, 23 (1991), 345–405.

[24] D. Avis, J.-M. Robert, and R. Wenger, Lower bounds for line stabbing, *Inform. Process. Lett.*, 33 (1989), 59–62.

[25] D. Avis and R. Wenger, Algorithms for line transversals in space, *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, 1987, pp. 300–307.

[26] D. Avis and R. Wenger, Polyhedral line transversals in space, *Discrete Comput. Geom.*, 3 (1988), 257–265.

[27] A. Baltsan and M. Sharir, On the shortest paths between two convex polyhedra, *J. ACM*, 35 (1988), 267–287.

[28] J. L. Bentley and T. A. Ottmann, Algorithms for reporting and counting geometric intersections, *IEEE Trans. Comput.*, C-28 (1979), 643–647.

[29] M. Bern, D. Dobkin, D. Eppstein, and R. Grossman, Visibility with a moving point of view, *Algorithmica*, 11 (1994), 360–378.

[30] M. Bern, D. Eppstein, P. Plassman, and F. Yao, Horizon theorems for lines and polygons, in: *Discrete and Computational Geometry: Papers from the DIMACS Special Year* (J. Goodman, R. Pollack, and W. Steiger, eds.), Amer. Math. Soc., Providence, RI, 1991, pp. 45–66.

[31] D. Bienstock and E. Győri, An extremal problem on sparse 0-1 matrices, *SIAM J. Discrete Math.*, 4 (1991), 17–27.

[32] R. Canham, A theorem on arrangements of lines in the plane, *Israel J. Math.*, 7 (1969), 393–397.

[33] J. Canny and J. H. Reif, New lower bound techniques for robot motion planning problems, *Proc. 28th Annu. IEEE Sympos. Found. Comput. Sci.*, 1987, pp. 49–60.

[34] B. Chazelle, H. Edelsbrunner, L. Guibas, M. Sharir, and J. Snoeyink, Computing a face in an arrangement of line segments and related problems, *SIAM J. Comput.*, 22 (1993), 1286–1302.

[35] B. Chazelle and L. J. Guibas, Fractional cascading: I. A data structuring technique, *Algorithmica*, 1 (1986), 133–162.

[36] B. Chazelle and D. T. Lee, On a circle placement problem, *Computing*, 36 (1986), 1–16.

[37] J. Chen and Y. Han, Shortest paths on a polyhedron, *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, 1990, pp. 360–369.

[38] L. P. Chew, Near-quadratic bounds for the L_1 Voronoi diagram of moving points, *Proc. 5th Canad. Conf. Comput. Geom.*, 1993, pp. 364–369.

[39] L. P. Chew and K. Kedem, A convex polygon among polygonal obstacles: Placement and high-clearance motion, *Comput. Geom. Theory Appl.*, 3 (1993), 59–89.

[40] J. Choi, J. Sellen, and C. K. Yap, Approximate Euclidean shortest path in 3-space, *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, 1994, pp. 41–48.

[41] K. Clarkson, Computing a single face in arrangements of segments, Unpublished manuscript, 1990.

[42] K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir, and E. Welzl, Combinatorial complexity bounds for arrangements of curves and spheres, *Discrete Comput. Geom.*, 5 (1990), 99–160.

[43] K. L. Clarkson, Approximation algorithms for shortest path motion planning, *Proc. 19th Annu. ACM Sympos. Theory Comput.*, 1987, pp. 56–65.

[44] K. L. Clarkson, Randomized geometric algorithms, in: *Computing in Euclidean Geometry* (D.-Z. Du and F. K. Hwang, eds.), World Scientific, Singapore, 1992, pp. 117–162.

[45] K. L. Clarkson and P. W. Shor, Applications of random sampling in computational geometry, II, *Discrete Comput. Geom.*, 4 (1989), 387–421.

[46] R. Cole and M. Sharir, Visibility problems for polyhedral terrains, *J. Symbolic Comput.*, 7 (1989), 11–30.

[47] H. Davenport, A combinatorial problem connected with differential equations II, *Acta Arithmetica*, 17 (1971), 363–372.

[48] H. Davenport and A. Schinzel, A combinatorial problem connected with differential equations, *Amer. J. Math.*, 87 (1965), 684–689.

[49] M. de Berg, K. Dobrindt, and O. Schwarzkopf, On lazy randomized incremental construction, *Discrete Comput. Geom.*, 14 (1995), 261–286.

[50] T. Dey, Improved bounds for k-sets and k-th levels, *Proc. 38th Annu. IEEE Sympos. Found. Comput. Sci.*, 1997, pp. 156–161.

[51] A. Dobson and S. Macdonald, Lower bounds for lengths of Davenport-Schinzel sequences, *Utilitas Math.*, 6 (1974), 251–257.

[52] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer-Verlag, Heidelberg, 1987.

[53] H. Edelsbrunner, L. Guibas, J. Hershberger, J. Pach, R. Pollack, R. Seidel, M. Sharir, and J. Snoeyink, Arrangements of Jordan arcs with three intersections per pair, *Discrete Comput. Geom.*, 4 (1989), 523–539.

[54] H. Edelsbrunner, L. Guibas, J. Pach, R. Pollack, R. Seidel, and M. Sharir, Arrangements of curves in the plane: Topology, combinatorics, and algorithms, *Theoret. Comput. Sci.*, 92 (1992), 319–336.

[55] H. Edelsbrunner and L. J. Guibas, Topologically sweeping an arrangement, *J. Comput. Syst. Sci.*, 38 (1989), 165–194. Corrigendum in 42 (1991), 249–251.

[56] H. Edelsbrunner, L. J. Guibas, and M. Sharir, The complexity and construction of many faces in arrangements of lines and of segments, *Discrete Comput. Geom.*, 5 (1990), 161–196.

[57] H. Edelsbrunner, H. A. Maurer, F. P. Preparata, A. L. Rosenberg, E. Welzl, and D. Wood, Stabbing line segments, *BIT*, 22 (1982), 274–281.

[58] H. Edelsbrunner, J. O'Rourke, and R. Seidel, Constructing arrangements of lines and hyperplanes with applications, *SIAM J. Comput.*, 15 (1986), 341–363.

[59] H. Edelsbrunner, R. Seidel, and M. Sharir, On the zone theorem for hyperplane arrangements, *SIAM J. Comput.*, 22 (1993), 418–429.

[60] H. Edelsbrunner and M. Sharir, The maximum number of ways to stab n convex nonintersecting sets in the plane is $2n-2$, *Discrete Comput. Geom.*, 5 (1990), 35–42.

[61] A. Efrat, Finding approximate matching of points and segments under translation, Unpublished manuscript, 1995.

[62] A. Efrat and M. Sharir, A near-linear algorithm for the planar segment center problem, *Discrete Comput. Geom.*, 16 (1996), 239–257.

[63] P. Egyed and R. Wenger, Ordered stabbing of pairwise disjoint convex sets in linear time, *Abstracts 1st Canad. Conf. Comput. Geom.*, 1989, p. 23.

[64] P. Erdős, L. Lovász, A. Simmons, and E. Straus, Dissection graphs of planar point sets, in: *A Survey of Combinatorial Theory* (J. N. Srivastava, ed.), North-Holland, Amsterdam, Netherlands, 1973, pp. 139–154.

[65] H. Everett, J.-M. Robert, and M. van Kreveld, An optimal algorithm for the ($\leq k$)-levels, with applications to separation and transversal problems, *Internat. J. Comput. Geom. Appl.*, 6 (1996), 247–261.

[66] S. J. Fortune, A sweepline algorithm for Voronoi diagrams, *Algorithmica*, 2 (1987), 153–174.

[67] J.-J. Fu and R. C. T. Lee, Voronoi diagrams of moving points in the plane, *Internat. J. Comput. Geom. Appl.*, 1 (1991), 23–32.

[68] Z. Füredi and P. Hajnal, Davenport-Schinzel theory of matrices, *Discrete Math.*, 103 (1992), 233–251.

[69] D. Gardy and D. Gouyou-Beauchamps, Enumeration of some Davenport-Schinzel sequences, Report 564, Université de Paris-Sud, Paris, 1990.

[70] J. E. Goodman, R. Pollack, and R. Wenger, Geometric transversal theory, in: *New Trends in Discrete and Computational Geometry* (J. Pach, ed.), Springer-Verlag, Heidelberg–New York–Berlin, 1993, pp. 163–198.

[71] L. Guibas, J. S. B. Mitchell, and T. Roos, Voronoi diagrams of moving points in the plane, *Proc. 17th Internat. Workshop Graph-Theoret. Concepts Comput. Sci.*, *Lecture Notes Comput. Sci.*, Vol. 570, Springer-Verlag, 1991, pp. 113–125.

[72] L. J. Guibas, M. Sharir, and S. Sifrony, On the general motion planning problem with two degrees of freedom, *Discrete Comput. Geom.*, 4 (1989), 491–521.

[73] P. Gupta, R. Janardan, and M. Smid, Fast algorithms for collision and proximity problems involving moving geometric objects, *Proc. 2nd Annu. European Sympos. Algorithms*, *Lecture Notes Comput. Sci.*, Vol. 855, Springer-Verlag, 1994, pp. 278–289.

[74] D. Halperin and M. Sharir, New bounds for lower envelopes in three dimensions, with applications to visibility in terrains, *Discrete Comput. Geom.*, 12 (1994), 313–326.

[75] S. Har-Peled, *The Complexity of Many Cells in the Overlay of Many Arrangements*, M.S. Thesis, Dept. Computer Science, Tel Aviv University, Tel Aviv, Israel, 1995.

[76] S. Har-Peled, T. M. Chan, B. Aronov, D. Halperin, and J. Snoeyink, The complexity of a single face of a Minkowski sum, *Proc. 7th Canad. Conf. Comput. Geom.*, 1995, pp. 91–96.

[77] S. Hart and M. Sharir, Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes, *Combinatorica*, 6 (1986), 151–177.

[78] J. Hershberger, Finding the upper envelope of n line segments in $O(n \log n)$ time, *Inform. Process. Lett.*, 33 (1989), 169–174.

[79] D. P. Huttenlocher and K. Kedem, Computing the minimum Hausdorff distance for point sets under translation, *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, 1990, pp. 340–349.

[80] D. P. Huttenlocher, K. Kedem, and J. M. Kleinberg, On dynamic Voronoi diagrams and the minimum Hausdorff distance for point sets under Euclidean motion in the plane, *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, 1992, pp. 110–120.

[81] D. P. Huttenlocher, K. Kedem, and M. Sharir, The upper envelope of Voronoi surfaces and its applications, *Discrete Comput. Geom.*, 9 (1993), 267–291.

[82] H. Hwang, R. Chang, and H. Tu, The separability problem in dynamic computational geometry, Unpublished manuscript, 1990.

[83] K. Imai, S. Sumino, and H. Imai, Minimax geometric fitting of two corresponding sets of points, *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, 1989, pp. 266–275.

[84] M. Katchalski, T. Lewis, and A. Liu, The different ways of stabbing disjoint convex sets, *Discrete Comput. Geom.*, 7 (1992), 197–206.

[85] N. Katoh, T. Tokuyama, and K. Iwano, On minimum and maximum spanning trees of linearly moving points, *Discrete Comput. Geom.*, 13 (1995), 161–176.

[86] M. J. Katz, 3-D vertical ray shooting and 2-D point enclosure, range searching, and arc shooting amidst convex fat objects, Research Report 2583, INRIA, BP93, 06902 Sophia-Antipolis, France, 1995.

[87] K. Kedem, R. Livne, J. Pach, and M. Sharir, On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles, *Discrete Comput. Geom.*, 1 (1986), 59–71.

[88] K. Kedem and M. Sharir, An efficient motion planning algorithm for a convex rigid polygonal object in 2-dimensional polygonal space, *Discrete Comput. Geom.*, 5 (1990), 43–75.

[89] K. Kedem, M. Sharir, and S. Toledo, On critical orientations in the Kedem-Sharir motion planning algorithm for a convex polygon in the plane, *Discrete Comput. Geom.*, 17 (1997), 227–240.

[90] J. Ketonen and R. Solovay, Rapidly growing Ramsey functions, *Annals of Math.*, 113 (1981), 267–314.

[91] M. M. Klawe, Superlinear bounds on matrix searching, *Proc. 1st ACM-SIAM Sympos. Discrete Algorithms*, 1990, pp. 485–493.

[92] M. Klazar, A general upper bound in extremal theory of sequences, *Comment. Math. Univ. Carol.*, 33 (1992), 737–746.

[93] M. Klazar, Two results on a partial ordering of finite sequences, *Comment. Math. Univ. Carol.*, 34 (1993), 667–675.

[94] M. Klazar, A linear upper bound in extremal theory of sequences, *J. Combin. Theory Ser. A*, 68 (1994), 454–464.

[95] M. Klazar, *Combinatorial Aspects of Davenport–Schinzel Sequences*, Ph.D. Thesis, Department of Applied Mathematics, Charles University, Prague, Czech Republic, 1995.

[96] M. Klazar and P. Valtr, Generalized Davenport–Schinzel sequences, *Combinatorica*, 14 (1994), 463–476.

[97] P. Komjáth, A simplified construction of nonlinear Davenport–Schinzel sequences, *J. Combin. Theory Ser. A*, 49 (1988), 262–267.

[98] D. T. Lee, Two-dimensional Voronoi diagrams in the L_p-metric, *J. ACM*, 27 (1980), 604–618.

[99] D. Leven and M. Sharir, Intersection and proximity problems and Voronoi diagrams, in: *Advances in Robotics 1: Algorithmic and Geometric Aspects of Robotics* (J. T. Schwartz and C.-K. Yap, eds.), Lawrence Erlbaum Associates, Hillsdale, NJ, 1987, pp. 187–228.

[100] D. Leven and M. Sharir, On the number of critical free contacts of a convex polygonal object moving in two-dimensional polygonal space, *Discrete Comput. Geom.*, 2 (1987), 255–270.

[101] L. Lovász, On the number of halving lines, *Annal. Univ. Scie. Budapest. de Rolando Eötvös Nominatae, Sectio Math.*, 14 (1971), 107–108.

[102] J. Matoušek, J. Pach, M. Sharir, S. Sifrony, and E. Welzl, Fat triangles determine linearly many holes, *SIAM J. Comput.*, 23 (1994), 154–169.

[103] J. Matoušek and P. Valtr, The complexity of the lower envelope of segments with h endpoints, in *Intuitive Geometry, Colloq. Math. Societ y János Bolyai*, North Holland, Amsterdam, 1997, pp. 407–411.

[104] N. Megiddo, Dynamic location problems, *Oper. Res.*, 6 (1986), 313–319.

[105] W. Mills, Some Davenport–Schinzel sequences, *Congressus Numer. IX: Proc. 3rd Manitoba Conf. on Numer. Math. and Computing* (R. Thomas and H. Williams, eds.), 1973, pp. 307–313.

[106] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou, The discrete geodesic problem, *SIAM J. Comput.*, 16 (1987), 647–668.

[107] C. Monma and S. Suri, Transitions in geometric minimum spanning trees, *Discrete Comput. Geom.*, 8 (1992), 265–293.

[108] D. M. Mount, The number of shortest paths on the surface of a polyhedron, *SIAM J. Comput.*, 19 (1990), 593–611.

[109] R. Mullin and R. Stanton, A map-theoretic approach to Davenport–Schinzel sequences, *Pacific J. Math.*, 40 (1972), 167–172.

[110] K. Mulmuley, Hidden surface removal with respect to a moving point, *Proc. 23rd Annu. ACM Sympos. Theory Comput.*, 1991, pp. 512–522.

[111] K. Mulmuley, On levels in arrangements and Voronoi diagrams, *Discrete Comput. Geom.*, 6 (1991), 307–338.

[112] K. Mulmuley, *Computational Geometry: An Introduction Through Randomized Algorithms*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[113] T. Ottmann and D. Wood, Dynamical sets of points, *Comput. Vision Graph. Image Process.*, 27 (1984), 157–166.

[114] J. Pach and M. Sharir, On vertical visibility in arrangements of segments and the queue size in the Bentley-Ottmann line sweeping algorithm, *SIAM J. Comput.*, 20 (1991), 460–470.

[115] J. Pach, W. Steiger, and E. Szemerédi, An upper bound on the number of planar k-sets, *Discrete Comput. Geom.*, 7 (1992), 109–123.

[116] C. H. Papadimitriou, An algorithm for shortest-path motion in three dimensions, *Inform. Process. Lett.*, 20 (1985), 259–263.

[117] C. H. Papadimitriou and E. B. Silverberg, Optimal piecewise linear motion of an object among obstacles, *Algorithmica*, 2 (1987), 523–539.

[118] J. Paris and L. Harrington, A mathematical incompleteness in Peano arithmetic, in: *Handbook of Mathematical Logic* (J. Barwise, ed.), North-Holland, Amsterdam, Netherlands, 1977, pp. 1133–1142.

[119] C. Peterkin, Some results on Davenport–Schinzel sequences, *Congressus Numer. IX: Proc. 3rd Manitoba Conf. on Numer. Math. and Computing* (R. Thomas and H. Williams, eds.), 1973, pp. 337–344.

[120] R. Pollack, M. Sharir, and S. Sifrony, Separating two simple polygons by a sequence of translations, *Discrete Comput. Geom.*, 3 (1988), 123–136.

[121] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.

[122] E. Ramos, Intersection of unit-balls and diameter of a point set in R^3, *Computat. Geom. Theory Appl.*, 6 (1996), in press.

[123] H. Rogers, *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, New York, 1967.

[124] T. Roos, Voronoi diagrams over dynamic scenes, *Proc. 2nd Canad. Conf. Comput. Geom.*, 1990, pp. 209–213.

[125] T. Roos, *Dynamic Voronoi diagrams*, Ph.D. Thesis, Bayerische Julius-Maximilians-Univ., Würzburg, Germany, 1991.

[126] T. Roos, Tighter bounds on Voronoi diagrams of moving points, *Proc. 5th Canad. Conf. Comput. Geom.*, 1993, pp. 358–363.

[127] D. Roselle, An algorithmic approach to Davenport–Schinzel sequences, *Utilitas Math.*, 6 (1974), 91–93.

[128] D. Roselle and R. Stanton, Results on Davenport–Schinzel sequences, *Congressus Numer. I, Proc. Louisiana Conf. on Combinatorics, Graph Theory, and Computing* (R. Mullin et al., eds.), 1970, pp. 249–267.

[129] D. Roselle and R. Stanton, Some properties of Davenport–Schinzel sequences, *Acta Arithmetica*, 17 (1971), 355–362.

[130] C. Schevon and J. O'Rourke, The number of maximal edge sequences on a convex polytope, *Proc. 26th Allerton Conf. Commun. Control Comput.*, October 1988, pp. 49–57.

[131] C. Schevon and J. O'Rourke, An algorithm for finding edge sequences on a polytope, Technical Report JHU-89/03, Dept. Comput. Sci., Johns Hopkins Univ., Baltimore, MD, February 1989.

[132] J. T. Schwartz and M. Sharir, On the two-dimensional Davenport-Schinzel problem, *J. Symbolic Comput.*, 10 (1990), 371–393.

[133] R. Seidel, Backwards analysis of randomized geometric algorithms, in: *New Trends in Discrete and Computational Geometry* (J. Pach, ed.), Springer-Verlag, Heidelberg, Germany, 1993, pp. 37–68.

[134] M. Sharir, Almost linear upper bounds for the length of general Davenport-Schinzel sequences, *Combinatorica*, 7 (1987), 131–143.

[135] M. Sharir, Improved lower bounds on the length of Davenport-Schinzel sequences, *Combinatorica*, 8 (1988), 117–124.

[136] M. Sharir, A note on the Papadimitriou-Silverberg algorithm for planning optimal piecewise linear motion of a ladder, *Inform. Process. Lett.*, 32 (1989), 187–190.

[137] M. Sharir, On k-sets in arrangements of curves and surfaces, *Discrete Comput. Geom.*, 6 (1991), 593–613.

[138] M. Sharir, Almost tight upper bounds for lower envelopes in higher dimensions, *Discrete Comput. Geom.*, 12 (1994), 327–345.

[139] M. Sharir and P. K. Agarwal, *Davenport-Schinzel Sequences and Their Geometric Applications*, Cambridge University Press, New York, 1995.

[140] M. Sharir and A. Schorr, On shortest paths in polyhedral spaces, *SIAM J. Comput.*, 15 (1986), 193–215.

[141] M. Sharir and S. Toledo, Extremal polygon containment problems, *Comput. Geom. Theory Appl.*, 4 (1994), 99–118.

[142] P. Shor, Geometric realization of superlinear Davenport–Schinzel sequences: I. Line segments, Unpublished manuscript, 1990.

[143] P. Shor, Private communication, 1990.

[144] R. Stanton and P. Dirksen, Davenport–Schinzel sequences, *Ars Combinatorica*, 1 (1976), 43–51.

[145] R. Stanton and D. Roselle, A result on Davenport–Schinzel sequences, in: *Combinatorial Theory and Its Applications, Colloq. Math. Society János Bolyai* (P. Erdős et al., eds.), Vol. 4, North-Holland, Amsterdam, 1969, pp. 1023–1027.

[146] E. Szemerédi, On a problem of Davenport and Schinzel, *Acta Arithmetica*, 25 (1974), 213–224.

[147] H. Tamaki and T. Tokuyama, How to cut pseudo-parabolas into segments, *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, 1995, pp. 230–237.

[148] A. Tamir, Improved complexity bounds for center location problems on networks by using dynamic data structures, *SIAM J. Discrete Math.*, 1 (1988), 377–396.

[149] A. Wiernik and M. Sharir, Planar realizations of nonlinear Davenport-Schinzel sequences by segments, *Discrete Comput. Geom.*, 3 (1988), 15–47.