Chapter 13

Line Arrangements

During the course of this lecture we encountered several situations where it was convenient to assume that a point set is "in general position". In the plane, general position usually amounts to no three points being collinear and/or no four of them being cocircular. This raises an algorithmic question: How can we test for n given points whether or not three of them are collinear? Obviously, we can test all triples in $O(n^3)$ time. Can we do better? In order to answer this question, we will take a detour through the dual plane.

Observe that points and hyperplanes in $\mathbb{R}^d$ are very similar objects, given that both can be described using d coordinates/parameters. It is thus tempting to match these parameters to each other and so create a mapping between points and hyperplanes. In $\mathbb{R}^2$ hyperplanes are lines and the standard *projective duality* transform maps a point $p = (p_x, p_y)$ to the line $p^* : y = p_x x - p_y$ and a non-vertical line $g : y = mx + b$ to the point $g^* = (m, -b)$.

Proposition 13.1 *The standard projective duality transform is*

- *incidence preserving:* $p \in g \iff g^* \in p^*$ *and*
- *order preserving:* p *is above* $g \iff g^*$ *is above* p^*.

Exercise 13.2 *Prove Proposition 13.1.*

Exercise 13.3 *Describe the image of the following point sets under this mapping*

a) a halfplane

b) $k \geqslant 3$ *collinear points*

c) a line segment

d) the boundary points of the upper convex hull of a finite point set.

Another way to think of duality is in terms of the parabola $\mathcal{P} : y = \frac{1}{2}x^2$. For a point p on $\mathcal{P}$, the dual line p^* is the tangent to $\mathcal{P}$ at p. For a point p not on $\mathcal{P}$, consider the vertical projection p' of p onto $\mathcal{P}$: the slopes of p^* and p'^* are the same, just p^* is shifted by the difference in y-coordinates.

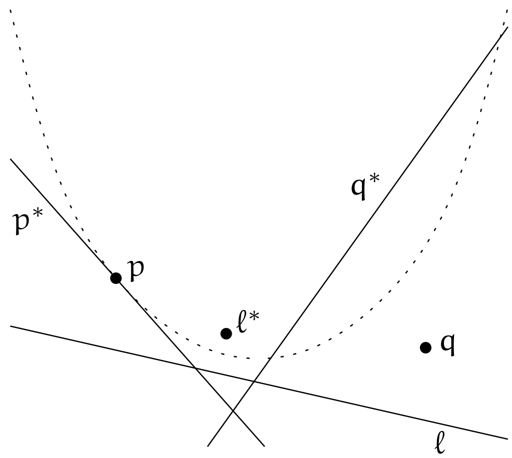


Figure 13.1: *Point ↔ line duality with respect to the parabola* $y = \frac{1}{2}x^2$.

The question of whether or not three points in the primal plane are collinear transforms to whether or not three lines in the dual plane meet in a point. This question in turn we will answer with the help of *line arrangements*, as defined below.

13.1 Arrangements

The subdivision of the plane induced by a finite set L of lines is called the **arrangement** $\mathcal{A}(L)$. Observe that all cells of the subdivision are intersections of halfplanes and thus convex. A line arrangement is **simple** if no two lines are parallel and no three lines meet in a point. Although lines are unbounded, we can regard a line arrangement a bounded object by (conceptually) putting a sufficiently large box around that contains all vertices. Such a box can be constructed in $O(n \log n)$ time for n lines.

Exercise 13.4 *How?*

Moreover, we can view a line arrangement as a planar graph by adding an additional vertex at "infinity", that is incident to all rays which leave this bounding box. For algorithmic purposes, we will mostly think of an arrangement as being represented by a doubly connected edge list (DCEL), cf. Section 5.2.

Theorem 13.5 *A simple arrangement* $\mathcal{A}(L)$ *of* n *lines in* $\mathbb{R}^2$ *has* $\binom{n}{2}$ *vertices,* n^2 *edges, and* $\binom{n}{2} + n + 1$ *faces/cells.*

Proof. Since all lines intersect and all intersection points are pairwise distinct, there are $\binom{n}{2}$ vertices.

The number of edges we count using induction on n. For $n = 1$ we have $1^2 = 1$ edge. By adding one line to an arrangement of $n-1$ lines we split $n-1$ existing edges into two and introduce n new edges along the newly inserted line. Thus, there are in total $(n-1)^2 + 2n - 1 = n^2 - 2n + 1 + 2n - 1 = n^2$ edges.

The number f of faces can now be obtained from Euler's formula $v - e + f = 2$, where v and e denote the number of vertices and edges, respectively. However, in order to apply Euler's formula we need to consider $\mathcal{A}(L)$ as a planar graph and take the symbolic "infinite" vertex into account. Therefore,

$$f = 2 - \left(\binom{n}{2} + 1\right) + n^2 = 1 + \frac{1}{2}(2n^2 - n(n-1)) = 1 + \frac{1}{2}(n^2 + n) = 1 + \binom{n}{2} + n\,.$$

□

The **complexity** of an arrangement is simply the total number of vertices, edges, and faces (in general, cells of any dimension).

Exercise 13.6 *Consider a set of lines in the plane with no three intersecting in a common point. Form a graph G whose vertices are the intersection points of the lines and such that two vertices are adjacent if and only if they appear consecutively along one of the lines. Prove that $\chi(G) \leqslant 3$, where $\chi(G)$ denotes the chromatic number of the graph G. In other words, show how to color the vertices of G using at most three colors such that no two adjacent vertices have the same color.*

13.2 Construction

As the complexity of a line arrangement is quadratic, there is no need to look for a subquadratic algorithm to construct it. We will simply construct it incrementally, inserting the lines one by one. Let $\ell_1, \ldots, \ell_n$ be the order of insertion.

At Step i of the construction, locate ℓ_i in the leftmost cell of $\mathcal{A}(\{\ell_1, \ldots, \ell_{i-1}\})$ it intersects. (The halfedges leaving the infinite vertex are ordered by slope.) This takes $O(i)$ time. Then traverse the boundary of the face F found until the halfedge h is found where ℓ_i leaves F (see Figure 13.2 for illustration). Insert a new vertex at this point, splitting F and h and continue in the same way with the face on the other side of h.

The insertion of a new vertex involves splitting two halfedges and thus is a constant time operation. But what is the time needed for the traversal? The complexity of $\mathcal{A}(\{\ell_1, \ldots, \ell_{i-1}\})$ is $\Theta(i^2)$, but we will see that the region traversed by a single line has linear complexity only.

13.3 Zone Theorem

For a line ℓ and an arrangement $\mathcal{A}(L)$, the **zone** $Z_{\mathcal{A}(L)}(\ell)$ of ℓ in $\mathcal{A}(L)$ is the set of cells from $\mathcal{A}(L)$ whose closure intersects ℓ.

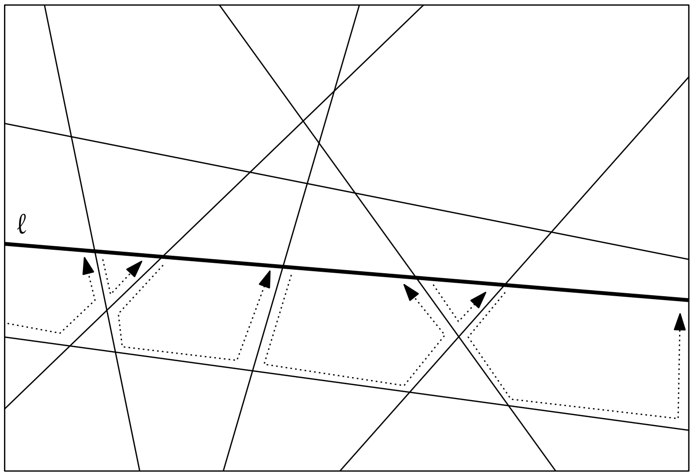
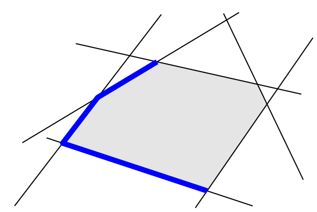


Figure 13.2: *Incremental construction: Insertion of a line ℓ. (Only part of the arrangement is shown in order to increase readability.)*

Theorem 13.7 *Given an arrangement $\mathcal{A}(L)$ of n lines in $\mathbb{R}^2$ and a line ℓ (not necessarily from L), the total number of edges in all cells of the zone $Z_{\mathcal{A}(L)}(\ell)$ is at most $6n$.*

Proof. Without loss of generality suppose that ℓ is horizontal and that none of the lines from L is horizontal. (The first condition can be addressed by rotating the plane and the second by deciding that the left vertex of a horizontal edge is higher than the right vertex.)



For each cell of $Z_{\mathcal{A}(L)}(\ell)$ split its boundary at its topmost vertex and at its bottommost vertex and orient all edges from bottom to top. Those edges that have the cell to their right are called *left-bounding* for the cell and those edges that have the cell to their left are called right-bounding. For instance, for the cell depicted to the right all left-bounding edges are shown blue and bold.

We will show that there are at most $3n$ left-bounding edges in $Z_{\mathcal{A}(L)}(\ell)$ by induction on n. By symmetry, the same bound holds also for the number of right-bounding edges in $Z_{\mathcal{A}(L)}(\ell)$.

For $n = 1$, there is at most one (exactly one, unless ℓ is parallel to and lies below the only line in L) left-bounding edge in $Z_{\mathcal{A}(L)}(\ell)$ and $1 \leqslant 3n = 3$. Assume the statement is true for $n - 1$.

If no line from L intersects ℓ, then all lines in $L \cup \{\ell\}$ are parallel and there are at most $2 < 3n$ left-bounding edges in $Z_{\mathcal{A}(L)}(\ell)$. Else consider the rightmost line r from L intersecting ℓ and the arrangement $\mathcal{A}(L \setminus \{r\})$. By the induction hypothesis there are at most $3n - 3$ left-bounding edges in $Z_{\mathcal{A}(L\setminus\{r\})}(\ell)$. Adding r back adds at most three new left-bounding edges: At most two edges (call them ℓ_0 and ℓ_1) of the rightmost cell of $Z_{\mathcal{A}(L\setminus\{r\})}(\ell)$ are intersected by r and thereby split in two. Both of these two edges may be

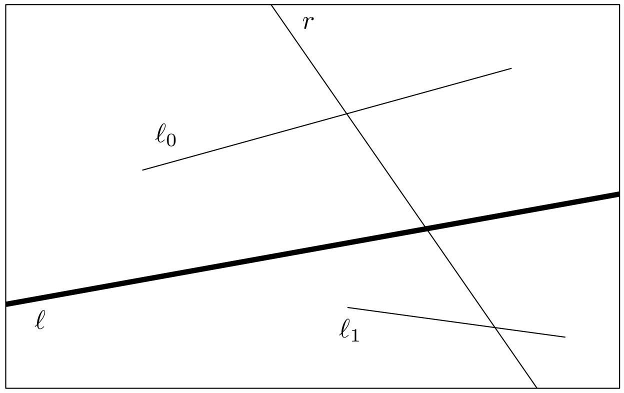


Figure 13.3: *At most three new left-bounding edges are created by adding* r *to* $\mathcal{A}(L \setminus \{r\})$.

left-bounding and thereby increase the number of left-bounding edges by at most two. In any case, r itself contributes exactly one more left-bounding edge to that cell. The line r cannot contribute a left-bounding edge to any cell other than the rightmost: to the left of r, the edges induced by r form right-bounding edges only and to the right of r all other cells touched by r (if any) are shielded away from ℓ by one of ℓ_0 or ℓ_1. Therefore, the total number of edges in $Z_{\mathcal{A}(L)}(\ell)$ is bounded from above by $3 + 3n - 3 = 3n$. □

Corollary 13.8 *The arrangement of* n *lines in* $\mathbb{R}^2$ *can be constructed in optimal* $O(n^2)$ *time and space.*

Proof. Use the incremental construction described above. In Step i, for $1 \leqslant i \leqslant n$, we do a linear search among $i - 1$ elements to find the starting face and then traverse (part of) the zone of the line ℓ_i in the arrangement $\mathcal{A}(\{\ell_1, \ldots, \ell_{i-1}\})$. By Theorem 13.7 the complexity of this zone and hence the time complexity of Step i altogether is $O(i)$. Overall we obtain $\sum_{i=1}^{n} ci = O(n^2)$ time (and space), for some constant $c > 0$, which is optimal by Theorem 13.5. □

The corresponding bounds for hyperplane arrangements in $\mathbb{R}^d$ are $\Theta(n^d)$ for the complexity of a simple arrangement and $O(n^{d-1})$ for the complexity of a zone of a hyperplane.

Exercise 13.9 *For an arrangement* $\mathcal{A}$ *of a set of* n *lines in* $\mathbb{R}^2$, *let* $\mathcal{F} := \bigcup_{C \text{ is cell of } \mathcal{A}} \overline{C}$ *denote the union of the closure of all bounded cells. Show that the complexity (number of vertices and edges of the arrangement lying on the boundary) of* $\mathcal{F}$ *is* $O(n)$.

13.4 The Power of Duality

The real beauty and power of line arrangements becomes apparent in context of projective point $\leftrightarrow$ line duality. The following problems all can be solved in $O(n^2)$ time and space by constructing the dual arrangement.

General position test. Given n points in $\mathbb{R}^2$, are any three of them collinear? (Dual: do three lines meet in a point?)

Minimum area triangle. Given n points in $\mathbb{R}^2$, what is the minimum area triangle spanned by any three of them? For any vertex ℓ^* of the dual arrangement (primal: line ℓ through two points p and q) find the closest point vertically above/below ℓ through which an input line passes (primal: closest line below/above and parallel to ℓ that passes through an input point). In this way one can find $O(n^2)$ candidate triangles by constructing the arrangement of the n dual lines[1] The smallest among those candidates can be determined by a straightforward minimum selection (comparing the area of the corresponding triangles). Observe that vertical distance is not what determines the area of the corresponding triangle but orthogonal distance. However, the points that minimize these measures for any fixed line are the same...

Exercise 13.10 *A set P of n points in the plane is said to be in ε-*general position *for $\varepsilon > 0$ if no three points of the form*

$$p + (x_1, y_1), q + (x_2, y_2), r + (x_3, y_3)$$

are collinear, where $p, q, r \in P$ and $|x_i|, |y_i| < \varepsilon$, for $i \in \{1, 2, 3\}$. In words: P remains in general position under changing point coordinates by less than ε each.

Give an algorithm with runtime $O(n^2)$ for checking whether a given point set P is in ε-general position.

13.5 Sorting all Angular Sequences.

Theorem 13.11 *Consider a set P of n points in the plane. For a point $q \in P$ let $c_P(q)$ denote the circular sequence of points from $S \setminus \{q\}$ ordered counterclockwise around q (in order as they would be encountered by a ray sweeping around q). All $c_P(q)$, $q \in P$, collectively can be obtained in $O(n^2)$ time.*

Proof. Assume without loss of generality that no two points in P have the same x-coordinate (else rotate the plane infinitesimally). Consider the projective dual P^* of P. An angular sweep around a point $q \in P$ in the primal plane corresponds to a traversal of the line q^* from left to right in the dual plane. (A collection of lines through a single point q corresponds to a collection of points on a single line q^* and slope corresponds to x-coordinate.) Clearly, the sequence of intersection points along all lines in P^* can be obtained by constructing the arrangement in $O(n^2)$ time. In the primal plane, any such sequence corresponds to an order of the remaining points according to the slope of

[1]For instance, maintain over the incremental construction for each vertex a vertically closest line. The number of vertices to be updated during insertion of a line ℓ corresponds to the complexity of the zone of ℓ in the arrangement constructed so far. Therefore maintaining this information comes at no extra cost asymptotically.

the connecting line; to construct the circular sequence of points as they are encountered around q, we have to split the sequence obtained from the dual into those points that are to the left of q and those that are to the right of q; concatenating both yields the desired sequence. □

Exercise 13.12 (Eppstein [1]) *Describe an $O(n^2)$ time algorithm that given a set P of n points in the plane finds a subset of five points that form a strictly convex empty pentagon (or reports that there is none if that is the case). Empty means that the convex pentagon may not contain any other points of P.*
Hint: Start with a point $p \in P$ that is extremal in one direction and try to find out whether there is a solution P' containing p. For this, consider the star-shaped polygon that visits all points in radial order, as seen from p.
Remark: It was shown by Harborth [5] that every set of ten or more points in general position contains a subset of five points that form a strictly convex empty pentagon.

13.6 Segment Endpoint Visibility Graphs

A fundamental problem in motion planning is to find a short(est) path between two given positions in some domain, subject to certain constraints. As an example, suppose we are given two points $p, q \in \mathbb{R}^2$ and a set $S \subset \mathbb{R}^2$ of obstacles. What is the shortest path between p and q that avoids S?

Observation 13.13 *The shortest path (if it exists) between two points that does not cross a finite set of finite polygonal obstacles is a polygonal path whose interior vertices are obstacle vertices.*

One of the simplest type of obstacle conceivable is a line segment. In general the plane may be disconnected with respect to the obstacles, for instance, if they form a closed curve. However, if we restrict the obstacles to pairwise disjoint line segments then there is always a free path between any two given points. Apart from start and goal position, by the above observation we may restrict our attention concerning shortest paths to straight line edges connecting obstacle vertices, in this case, segment endpoints.

Definition 13.14 *Consider a set S of n disjoint line segments in $\mathbb{R}^2$. The* **segment endpoint visibility graph** *$\mathcal{V}(S)$ is a plane straight line graph defined on the segment endpoints. Two segment endpoints p and q are connected in $\mathcal{V}(S)$ if and only if*

- *the line segment $\overline{pq}$ is in S or*
- *$\overline{pq} \cap s \subseteq \{p, q\}$ for every segment $s \in S$.*

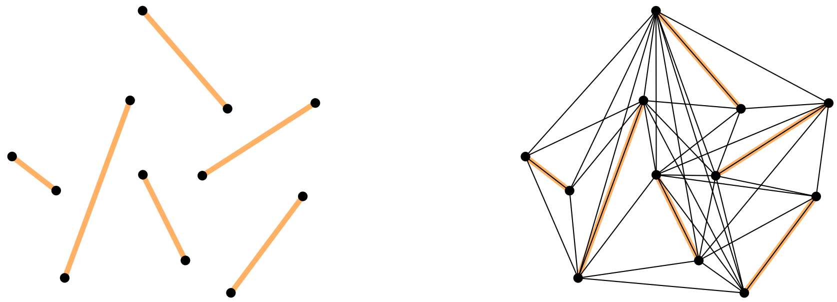


Figure 13.4: *A set of disjoint line segments and their endpoint visibility graph.*

If all segments are on the convex hull, the visibility graph is complete. If they form parallel chords of a convex polygon, the visibility graph consists of copies of K_4, glued together along opposite edges and the total number of edges is linear only.

These graphs also appear in the context of the following question: Given a set of disjoint line segments, is it possible to connect them to form (the boundary of) a simple polygon? Is it easy to see that this is not possible in general: Just take three parallel chords of a convex polygon (Figure 13.5a). However, if we do not insist that the segments appear on the boundary, but allow them to be diagonals or epigonals, then it is always possible [7, 6]. In other words, the segment endpoint visibility graph of disjoint line segments is Hamiltonian, unless all segments are collinear. It is actually essential to allow epigonals and not only diagonals [9, 4] (Figure 13.5b).

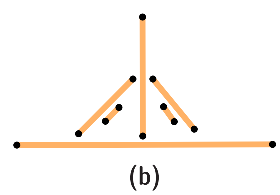


Figure 13.5: *Sets of disjoint line segments that do not allow certain polygons.*

Constructing $\mathcal{V}(S)$ for a given set S of disjoint segments in a brute force way takes $O(n^3)$ time. (Take all pairs of endpoints and check all other segments for obstruction.)

Theorem 13.15 (Welzl [10]) *The segment endpoint visibility graph of n disjoint line segments can be constructed in worst case optimal $O(n^2)$ time.*

Proof. For simplicity we assume general position, that is, no three endpoints are collinear and no two have the same x-coordinate. It is no problem to handle such degeneracies explicitly.

We have seen above how all sorted angular sequences can be obtained from the dual line arrangement in $O(n^2)$ time. Topologically sweep the arrangement from left to right (corresponds to changing the slope of the primal rays from $-\infty$ to $+\infty$) while maintaining for each segment endpoint p the segment $s(p)$ it currently "sees" (if any). Initialize by brute force in $O(n^2)$ time (direction vertically downwards). Each intersection of two lines corresponds to two segment endpoints "seeing" each other along the primal line whose dual is the point of intersection. In order to process an intersection, we only need that all preceding (located to the left) intersections of the two lines involved have already been processed. This order corresponds to a topological sort of the arrangement graph where all edges are directed from left to right. (Clearly, this graph is acyclic.) A topological sort can be obtained, for instance, via (reversed) post order DFS in linear time.

When processing an intersection, there are four cases. Let p and q be the two points involved such that p is to the left of q.

1. The two points belong to the same input segment $\rightarrow$ output the edge pq, no change otherwise.
2. q is obscured from p by $s(p) \rightarrow$ no change.
3. q is endpoint of $s(p) \rightarrow$ output pq and update $s(p)$ to $s(q)$.
4. Otherwise q is endpoint of a segment t that now obscures $s(p) \rightarrow$ output pq and update $s(p)$ to t.

Thus any intersection can be processed in constant time and the overall runtime of this algorithm is quadratic. □

13.7 Ham Sandwich Theorem

Suppose two thieves have stolen a necklace that contains rubies and diamonds. Now it is time to distribute the prey. Both, of course, should get the same number of rubies and the same number of diamonds. On the other hand, it would be a pity to completely disintegrate the beautiful necklace. Hence they want to use as few cuts as possible to achieve a fair gem distribution.

To phrase the problem in a geometric (and somewhat more general) setting: Given two finite sets R and D of points, construct a line that bisects both sets, that is, in either halfplane defined by the line there are about half of the points from R and about half of the points from D. To solve this problem, we will make use of the concept of levels in arrangements.

Definition 13.16 *Consider an arrangement $\mathcal{A}(L)$ induced by a set L of n non-vertical lines in the plane. We say that a point p is on the k-***level** *in $\mathcal{A}(L)$ if and only p lies on some line from L and there are at most $k-1$ lines below and at most $n-k$ lines above p. The 1-level and the n-level are also referred to as* **lower** *and* **upper envelope***, respectively.*

Another way to look at the k-level is to consider the lines to be real functions; then the lower envelope is the pointwise minimum of those functions, and the k-level is defined by taking pointwise the k^{th}-smallest function value.

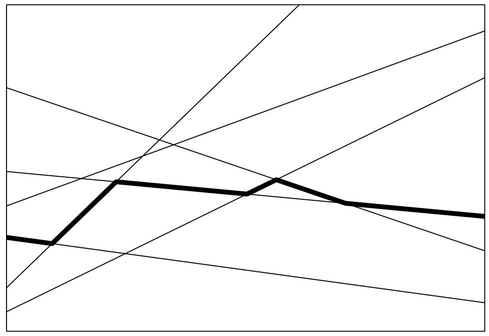


Figure 13.6: *The 3-level of an arrangement.*

Theorem 13.17 *Let* $R, D \subset \mathbb{R}^2$ *be finite sets of points. Then there exists a line that bisects both* R *and* D*. That is, in either open halfplane defined by* ℓ *there are no more than* $|R|/2$ *points from* R *and no more than* $|D|/2$ *points from* D*.*

Proof. Without loss of generality suppose that both $|R|$ and $|D|$ are odd. (If, say, $|R|$ is even, simply remove an arbitrary point from R. Any bisector for the resulting set is also a bisector for R.) We may also suppose that no two points from $R \cup D$ have the same x-coordinate. (Otherwise, rotate the plane infinitesimally.)

Let R^* and D^* denote the set of lines dual to the points from R and D, respectively. Consider the arrangement $\mathcal{A}(R^*)$. The median level of $\mathcal{A}(R^*)$ defines the bisecting lines for R. As $|R| = |R^*|$ is odd, both the leftmost and the rightmost segment of this level are defined by the same line ℓ_r from R^*, the one with median slope. Similarly there is a corresponding line ℓ_d in $\mathcal{A}(D^*)$.

Since no two points from $R \cup D$ have the same x-coordinate, no two lines from $R^* \cup D^*$ have the same slope, and thus ℓ_r and ℓ_d intersect. Consequently, being piecewise linear continuous functions, the median level of $\mathcal{A}(R^*)$ and the median level of $\mathcal{A}(D^*)$ intersect (see Figure 13.7 for an example). Any point that lies on both median levels corresponds to a primal line that bisects both point sets simultaneously. □

How can the thieves use Theorem 13.17? If they are smart, they drape the necklace along some convex curve, say, a circle. Then by Theorem 13.17 there exists a line that simultaneously bisects the set of diamonds and the set of rubies. As any line intersects the circle at most twice, the necklace is cut at most twice. It is easy to turn the proof given above into an $O(n^2)$ algorithm to construct a line that simultaneously bisects both sets.

You can also think of the two point sets as a discrete distribution of a ham sandwich that is to be cut fairly, that is, in such a way that both parts have the same amount of ham and the same amount of bread. That is where the name “ham sandwich cut” comes from. The theorem also holds in $\mathbb{R}^d$, saying that any d finite point sets (or finite Borel measures, if you want) can simultaneously be bisected by a hyperplane. This implies that the thieves can fairly distribute a necklace consisting of d types of gems using at most d cuts.

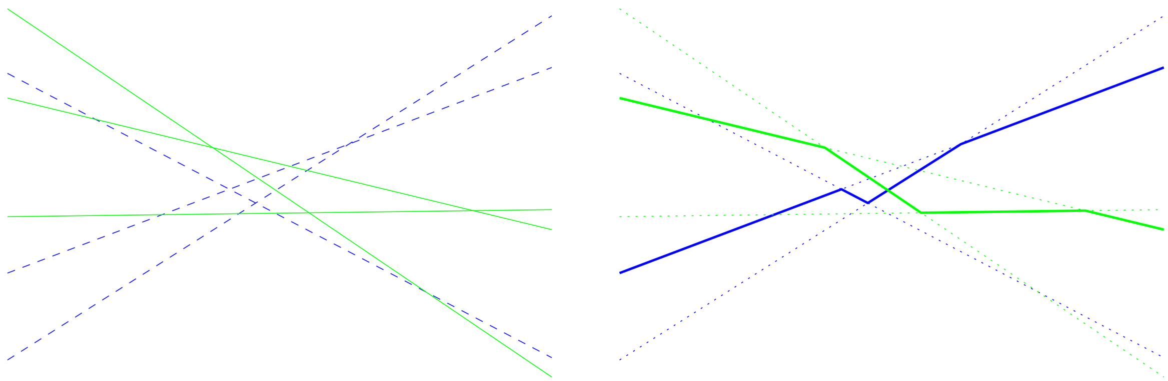


Figure 13.7: *An arrangement of 3 green lines (solid) and 3 blue lines (dashed) and their median levels (marked bold on the right hand side).*

Algorithmically the problem gets harder in higher dimension. But in the plane, a ham sandwich cut can be found in linear time using a sophisticated prune and search algorithm by Lo, Matoušek and Steiger [8].

Exercise 13.18 *The goal of this exercise is to develop a data structure for halfspace range counting.*

a) Given a set $P \subset \mathbb{R}^2$ of n points in general position, show that it is possible to partition this set by two lines such that each region contains at most $\lceil \frac{n}{4} \rceil$ points.

b) Design a data structure of size $O(n)$, which can be constructed in time $O(n \log n)$ and allows you, for any halfspace h, to output the number of points $|P \cap h|$ of P contained in this halfspace h in time $O(n^\alpha)$, for some $0 < \alpha < 1$.

Exercise 13.19 *Prove or disprove the following statement: Given three finite sets A, B, C of points in the plane, there is always a circle or a line that bisects A, B and C simultaneously (that is, no more than half of the points of each set are inside or outside the circle or on either side of the line, respectively).*

13.8 3-Sum

The 3-Sum problem is the following: Given a set S of n integers, does there exist a three-tuple[2] of elements from S that sum up to zero? By testing all three-tuples this can obviously be solved in $O(n^3)$ time. If the tuples to be tested are picked a bit more cleverly, we obtain an $O(n^2)$ algorithm.

Let $(s_1, \ldots, s_n)$ be the sequence of elements from S in increasing order. Then we test the tuples as follows.

[2] That is, an element of S may be chosen twice or even three times, although the latter makes sense for the number 0 only. :-)

```
For i = 1, ..., n {
    j = i, k = n.
    While k ⩾ j {
        If s_i + s_j + s_k = 0 then exit with triple s_i, s_j, s_k.
        If s_i + s_j + s_k > 0 then k = k − 1 else j = j + 1.
    }
}
```

The runtime is clearly quadratic (initial sorting can be done in $O(n \log n)$ time). Regarding the correctness observe that the following is an invariant that holds at the start of every iteration of the inner loop: $s_i + s_x + s_k < 0$, for all $i \leqslant x < j$, and $s_i + s_j + s_x > 0$, for all $k < x \leqslant n$.

Interestingly, this is the essentially the best algorithm known for 3-Sum. It is widely believed that the problem cannot be solved in sub-quadratic time, but so far this has been proved in some very restricted models of computation only, such as the linear decision tree model [2].

13.9 3-Sum hardness

There is a whole class of problems that are equivalent to 3-Sum up to sub-quadratic time reductions [3]; such problems are referred to as **3-Sum-hard**.

Definition 13.20 *A problem P is 3-Sum-hard if and only if every instance of 3-Sum of size n can be solved using a constant number of instances of P—each of $O(n)$ size—and $o(n^2)$ additional time.*

For instance, it is not hard to show that the following variation of 3-Sum—let us denote it by 3-Sum°—is 3-Sum hard: Given a set S of n integers, does there exist a three-element subset of S whose elements sum up to zero?

As another example, consider the Problem **GeomBase**: Given n points on the three horizontal lines $y = 0$, $y = 1$, and $y = 2$, is there a non-horizontal line that contains at least three of them?

3-Sum can be reduced to GeomBase as follows. For an instance $S = \{s_1, \dots, s_n\}$ of 3-Sum, create an instance P of GeomBase in which for each s_i there are three points in P: $(s_i, 0)$, $(-s_i/2, 1)$, and $(s_i, 2)$. If there are any three collinear points in P, there must be one from each of the lines $y = 0$, $y = 1$, and $y = 2$. So suppose that $p = (s_i, 0)$, $q = (-s_j/2, 1)$, and $r = (s_k, 2)$ are collinear. The inverse slope of the line through p and q is $\frac{-s_j/2-s_i}{1-0} = -s_j/2 - s_i$ and the inverse slope of the line through q and r is $\frac{s_k+s_j/2}{2-1} = s_k + s_j/2$. The three points are collinear if and only if the two slopes are equal, that is, $-s_j/2 - s_i = s_k + s_j/2 \iff s_i + s_j + s_k = 0$.

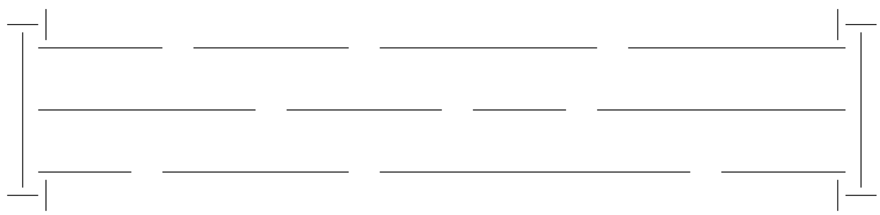
A very similar problem is **General Position**, in which one is given n arbitrary points and has to decide whether any three are collinear. For an instance S of 3-Sum°, create an instance P of General Position by projecting the numbers s_i onto the curve $y = x^3$, that is, $P = \{(a, a^3) \mid a \in S\}$.

Suppose three of the points, say, (a, a^3), (b, b^3), and (c, c^3) are collinear. This is the case if and only if the slopes of the lines through each pair of them are equal. (Observe that a, b, and c are pairwise distinct.)

$$\begin{aligned}
(b^3 - a^3)/(b - a) &= (c^3 - b^3)/(c - b) \iff \\
b^2 + a^2 + ab &= c^2 + b^2 + bc \iff \\
b &= (c^2 - a^2)/(a - c) \iff \\
b &= -(a + c) \iff \\
a + b + c &= 0 .
\end{aligned}$$

Minimum Area Triangle is a strict generalization of General Position and, therefore, also 3-Sum-hard.

In **Segment Splitting/Separation**, we are given a set of n line segments and have to decide whether there exists a line that does not intersect any of the segments but splits them into two non-empty subsets. To show that this problem is 3-Sum-hard, we can use essentially the same reduction as for GeomBase, where we interpret the points along the three lines $y = 0$, $y = 1$, and $y = 2$ as sufficiently small "holes". The parts of the lines that remain after punching these holes form the input segments for the Splitting problem. Horizontal splits can be prevented by putting constant size gadgets somewhere beyond the last holes, see the figure below. The set of input segments for the segment



splitting problem requires sorting the points along each of the three horizontal lines, which can be done in $O(n \log n) = o(n^2)$ time. It remains to specify what "sufficiently small" means for the size of those holes. As all input numbers are integers, it is not hard to see that punching a hole of $(x - 1/4, x + 1/4)$ around each input point x is small enough.

In **Segment Visibility**, we are given a set S of n horizontal line segments and two segments $s_1, s_2 \in S$. The question is: Are there two points, $p_1 \in s_1$ and $p_2 \in s_2$ which can see each other, that is, the open line segment $\overline{p_1 p_2}$ does not intersect any segment from S? The reduction from 3-Sum is the same as for Segment Splitting, just put s_1 above and s_2 below the segments along the three lines.

In **Motion Planning**, we are given a robot (line segment), some environment (modeled as a set of disjoint line segments), and a source and a target position. The question is: Can the robot move (by translation and rotation) from the source to the target position, without ever intersecting the "walls" of the environment?

To show that Motion Planning is 3-Sum-hard, employ the reduction for Segment Splitting from above. The three "punched" lines form the doorway between two rooms, each modeled by a constant number of segments that cannot be split, similar to the boundary gadgets above. The source position is in one room, the target position in the other, and to get from source to target the robot has to pass through a sequence of three collinear holes in the door (suppose the doorway is sufficiently small compared to the length of the robot).

Exercise 13.21 *The 3-Sum' problem is defined as follows: given three sets S_1, S_2, S_3 of n integers each, are there $a_1 \in S_1$, $a_2 \in S_2$, $a_3 \in S_3$ such that $a_1 + a_2 + a_3 = 0$? Prove that the 3-Sum' problem and the 3-Sum problem as defined in the lecture ($S_1 = S_2 = S_3$) are equivalent, more precisely, that they are reducible to each other in subquadratic time.*

Questions

56. *How can one construct an arrangement of lines in $\mathbb{R}^2$?* Describe the incremental algorithm and prove that its time complexity is quadratic in the number of lines (incl. statement and proof of the Zone Theorem).

57. *How can one test whether there are three collinear points in a set of n given points in $\mathbb{R}^2$?* Describe an $O(n^2)$ time algorithm.

58. *How can one compute the minimum area triangle spanned by three out of n given points in $\mathbb{R}^2$?* Describe an $O(n^2)$ time algorithm.

59. *What is a ham-sandwich cut? Does it always exist? How to compute it?* State and prove the theorem about the existence of a ham-sandwich cut in $\mathbb{R}^2$ and describe an $O(n^2)$ algorithm to compute it.

60. *What is the endpoint visibility graph for a set of disjoint line segments in the plane and how can it be constructed?* Give the definition and explain the relation to shortest paths. Describe the $O(n^2)$ algorithm by Welzl, including full proofs of Theorem 13.11 and Theorem 13.15.

61. *Is there a subquadratic algorithm for General Position?* Explain the term 3-Sum hard and its implications and give the reduction from 3-Sum to General Position.

62. *Which problems are known to be 3-Sum-hard?* List at least three problems (other than 3-Sum) and briefly sketch the corresponding reductions.

References

[1] David Eppstein, Happy endings for flip graphs. *J. Comput. Geom.*, **1**, 1, (2010), 3–28, URL http://jocg.org/index.php/jocg/article/view/21.

[2] Jeff Erickson, Lower bounds for linear satisfiability problems. *Chicago J. Theoret. Comput. Sci.*, **1999**, 8, URL http://cjtcs.cs.uchicago.edu/articles/1999/8/contents.html.

[3] Anka Gajentaan and Mark H. Overmars, On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom. Theory Appl.*, **5**, (1995), 165–185, URL http://dx.doi.org/10.1016/0925-7721(95)00022-2.

[4] Branko Grünbaum, Hamiltonian polygons and polyhedra. *Geombinatorics*, **3**, 3, (1994), 83–89.

[5] Heiko Harborth, Konvexe Fünfecke in ebenen Punktmengen. *Elem. Math.*, **33**, (1979), 116–118, URL http://dx.doi.org/10.5169/seals-32945.

[6] Michael Hoffmann, *On the existence of paths and cycles*. Ph.D. thesis, ETH Zürich, 2005, URL http://dx.doi.org/10.3929/ethz-a-004945082.

[7] Michael Hoffmann and Csaba D. Tóth, Segment endpoint visibility graphs are Hamiltonian. *Comput. Geom. Theory Appl.*, **26**, 1, (2003), 47–68, URL http://dx.doi.org/10.1016/S0925-7721(02)00172-4.

[8] Chi-Yuan Lo, Jiří Matoušek, and William L. Steiger, Algorithms for ham-sandwich cuts. *Discrete Comput. Geom.*, **11**, (1994), 433–452, URL http://dx.doi.org/10.1007/BF02574017.

[9] Masatsugu Urabe and Mamoru Watanabe, On a counterexample to a conjecture of Mirzaian. *Comput. Geom. Theory Appl.*, **2**, 1, (1992), 51–53, URL http://dx.doi.org/10.1016/0925-7721(92)90020-S.

[10] Emo Welzl, Constructing the visibility graph for n line segments in $O(n^2)$ time. *Inform. Process. Lett.*, **20**, (1985), 167–171, URL http://dx.doi.org/10.1016/0020-0190(85)90044-4.