Results In Discrete Geometry

Helen Anna Louise Bunting

PhD Thesis
University College London
1994

Abstract

This thesis deals with three different areas within the subject of discrete geometry.

Firstly I prove an upper bound and give an example that yields a lower bound, to show that the maximum number $f(d,n)$ of rich cells in an arrangement of n hyperplanes in $\Re^d$ is $f(d,n) = \Theta(n^{d-2})$. A cell of the arrangement is called *rich* if it is bounded non-trivially by every hyperplane. I define convex position for hyperplanes in $\Re^d$, and show that the Carathéodory number for lines in the plane is five. I consider extending these ideas, but show that a Helly number without redundancy does not exist for general convex sets, though for halfspaces $d+2$ is such a number.

In the second chapter I consider the 180° art gallery problem. I show that the number, $f_{180}(n)$, of guards required to survey a simple polygon with n sides is $f_{180}(n) \leq \lfloor (4n+1)/9 \rfloor$, if their angle of vision is restricted to at most 180°. This result has since been extended and the bound currently stands at $\lfloor 2n/5 \rfloor$. I prove that on the class of monotone polygons the bounds are identical with those obtained when guards may survey a full rotation: $f_{180}(n) = \min\{\lfloor n/3 \rfloor, \lfloor r/2 \rfloor + 1\}$, for a monotone polygon with n vertices, r of which are reflex. I also show that $f_\theta(n) \leq \lfloor n/2 \rfloor$ for $\theta < 180°$; that $f_{60}(n) \leq n-2$; and that $f_\theta(n) \geq n-2$ for $\theta < 60°$.

In the final chapter I describe an algorithm that answers the 3-dimensional diameter problem in a worst-case time of $O(n^{3/2} \log n)$. Previously only a sketch of this algorithm existed, in a paper by Chazelle, here I describe the algorithm in detail. For a long time this was the best known algorithm, however during the preparation of this thesis I have heard of the discovery of a new algorithm for the diameter that runs in an optimal time of $O(n \log n)$.

Acknowledgements

I would like to thank my supervisor David Larman for all of his help during the course of this research, and the Science and Engineering Research Council for their enabling financial support. Also Janos Pach who has been a prolific source of interesting open problems; especial thanks go to him and Imre Bárány for the many discussions.

Contents

List of Figures

Chapter 1

Arrangements Of Hyperplanes

1.1 Introduction

In this chapter I investigate the complex structures that can exist within an arrangement of hyperplanes. I concentrate on regions that have a piece of every hyperplane in their boundary. The problems solved here arose from the (so far unsuccesful except when $d = 2$) search for a '*Carathéodory*' number k such that a set of hyperplanes in $\Re^d$ are in convex position provided every k are in convex position. Such a result could form the start of an attempt to generalise the idea of convexity to take k-flats in $\Re^d$ as the basic objects, rather than points (see [15] for a similar attempt).

A family $\mathcal{F}$ of hyperplanes in $\Re^d$ is said to be in *convex position* if there exists a (compact) convex body just touching every member of $\mathcal{F}$. Recall that a consequence of Carathéodory's theorem [8] is that a set of points in $\Re^d$ is in convex position (forming the vertex set of their convex hull) provided all $(d+2)$-tuples of the points have this property. In section 1.4 an analogous result is obtained for a set of lines in the plane. It will be shown that a set of lines in the plane are in convex position provided every five of the lines are in convex position. Also in section 1.5 I show that such a concept cannot be used to obtain a Helly Theorem with no redundancies for general convex sets, though this is possible, of course, for halfspaces.

Given an arrangement of n hyperplanes in $\Re^d$ we call a cell of the arrangement *rich* if its boundary contains a piece of each of the hyperplanes, in other words if it has n facets, one supported by each hyperplane. The hyperplanes are in convex position if and only if there is some rich cell in

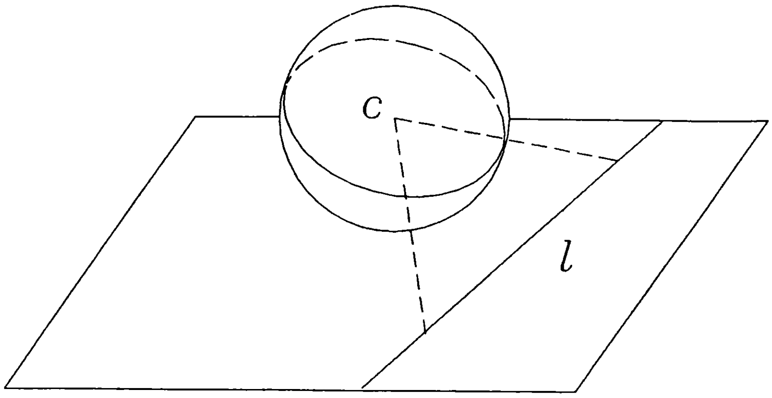


Figure 1.1: Projecting a hyperplane onto a great circle.

their arrangement. In section 1.2 it will be shown that there are at most $(n+8)^{d-2}/(d-2)!$ such rich cells, and in section 1.3 an example is constructed which gives a lower bound of the same order. The results given in this chapter will also appear in [3].

1.1.1 Map μ_d

To aid analysis of possible arrangements of lines in the plane, we introduce a geometric transform μ_d which maps halfspaces in $\Re^d$ to points on S^d. First, working in $\Re^{d+1}$ place a copy of S^d centre c onto $\Re^d$ so that they touch at one point. A hyperplane l of $\Re^d$ is projected onto a great circle by cutting S^d with the hyperplane H that contains l and c. See figure 1.1. We define μ_d to be the map which identifies the halfspace $l^+ \subset H^+$ with the point p where the outernormal to H^+ at c meets the sphere, in figure 1.2 overleaf the halfspace corresponding to p is shaded. The other halfspace l^- is mapped similarly to $\bar{p}$, the point antipodal to p.

Notice that (if we consider the points as points in $\Re^{d+1}$) when a set of hyperplanes is projected from $\Re^d$ onto great circles of S^d there is a one-to-one correspondence between the points x in $\Re^d$ and the points $[x,c] \cap S^d$ of the open lower hemisphere. The arrangement in $\Re^d$ is the same as the arrangement in the open lower hemisphere.

Lemma 1. *In $\Re^d$ some k hyperplanes through $\mathbf{0}$ bound a cone with k facets if and only if the k outer normals are in convex position in other words the dual cone* $\mathrm{Cone}\{n_1,\ldots,n_k\}$ *has k extreme rays.*

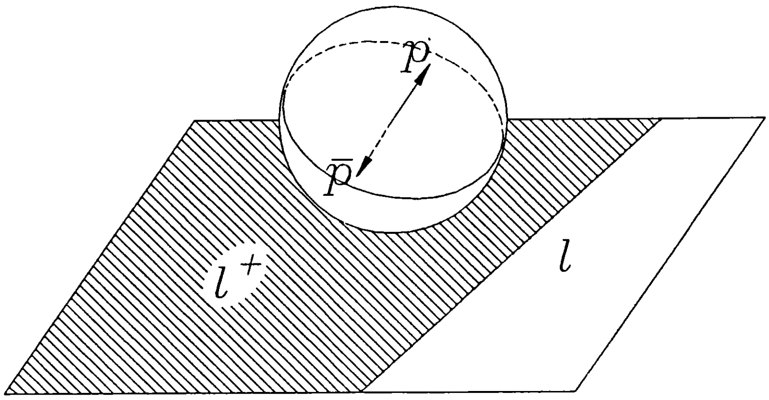


Figure 1.2: $\mu(l^-) = p$, $\mu(l^+) = \bar{p}$

Proof: Consider a cone C formed by k hyperplanes $H_1, \ldots, H_k$ in $\Re^d$, where the cone is defined by the halfspaces that have outer normals $n_1, \ldots, n_k$.

$$C = \cap H_i^- = \cap\{x : <n_i, x> \leq 0\}$$

If we assume that H_i forms a facet of C then we can find a point u_i in the interior of this facet thus $<n_i, u_i>= 0$, and $<n_j, u_i>< 0 \quad j \neq i$ for each $i \in \{1, \ldots, k\}$. Then consider the normals $n_1, \ldots, n_k$ out from $\mathbf{0}$; the hyperplane $H_i' = \{x : <u_i, x>= 0\}$ supports n_i, so the normals lie in convex position.

The other implication follows similarly. If we assume that the normals lie in convex position then it is possible to find a hyperplane H_i' with normal u_i that supports n_i. Thus $<u_i, n_i>= 0$ and $<u_i, n_j>< 0$ for $j \neq i$. But then u_i lies in the interior of the facet of C due to H_i. □

Corollary 1. *Points $p_1, \ldots, p_k$ on S^d are in (spherically) convex position forming P' if and only if in $\Re^{d+1}$ the corresponding halfspaces (with outer normals $\vec{cp_i}$[1]) intersect in a cone C apex c with k facets. This cone intersects the sphere in a spherical polyhedron P^* dual to P' which is bounded by k great circles.*

Proof: This follows directly since if the k outernormals are in convex position it follows that the normals lie in one halfspace. So if these normals intersect the sphere at the points $p_1, \ldots, p_k$ then these points lie in spherically convex position. □

[1]Where c is the centre of the sphere.

If μ_d is used to transform a set of hyperplanes i.e. pairs of halfspaces, to points on the sphere then because each region of the arrangement is a polyhedron it will identify with a collection of points P' (in convex position) on the sphere. Conversely every set of k spherically convex points of S^d have an associated dual P^* (notation is that of corollary 1) but not all such sets of points will yield a polyhedron P in $\Re^d$. In fact only the part of P^* that lies in the open lower hemisphere projects onto a region in $\Re^d$.

Given a set of hyperplanes we can use μ_d to find the equivalent normals p_i and $\bar{p_i}$ on the sphere but we wish to restrict our attention to k-gons P' that give actual regions in the space $\Re^d$ which are bounded by all k hyperplanes. We will call such a polygon P' of normals *admissible.*

To illustrate this consider the case when $d = 2$. Notice that P' is admissible provided the dual polygon P^* has no facet lying entirely in the closed upper hemisphere; as $d = 2$ P^* can have at most one vertex in this hemisphere.

(i) P is bounded if and only if P^* lies entirely in the lower hemisphere.

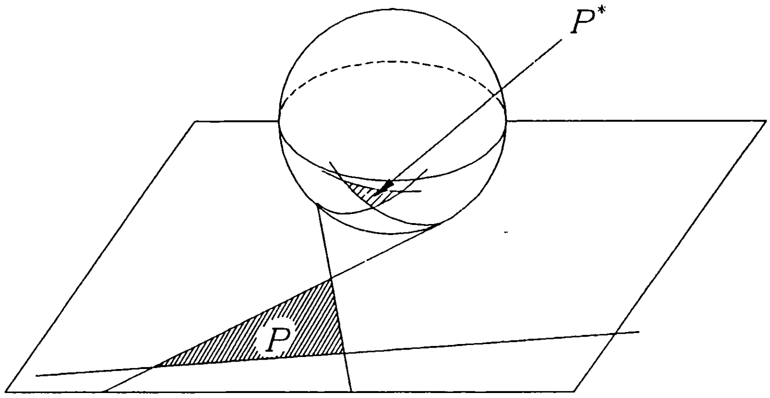


Figure 1.3: P is bounded.

(ii) P is unbounded if and only if P^* has one vertex in the closed upper hemisphere, the cell P^* meets the equator which represents infinity, this is illustrated in figure 1.4 overleaf.

(iii) If a facet of P^* is in the upper hemisphere the corresponding line does not meet the region P in $\Re^d$. Observe that the region diagonally opposite to P^* will also occur in the spherical arrangement. In the lower hemisphere the 'offending' facet/great circle bounds this cell of the arrangement which has a different admissible polygon.

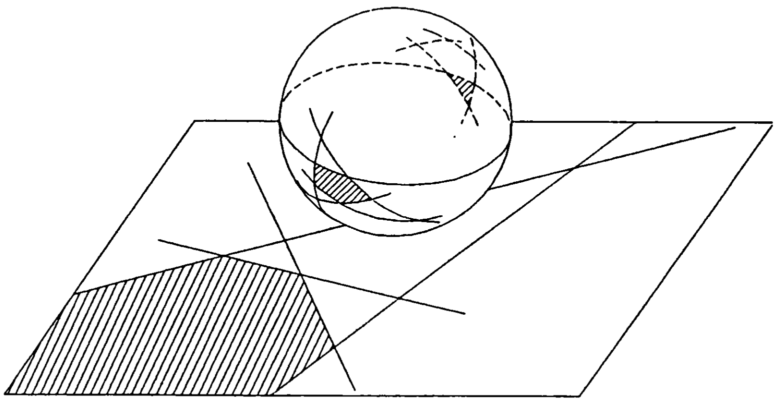


Figure 1.4: P is unbounded.

Keeping the same notation as that in lemma 1 we consider the case where a normal n does not define a facet. In the following *sph* will denote the spherical hull, which is the spherically convex polygon formed by the intersection of all hemispheres containing some set of points which lie on the sphere. The facets of this object are arcs of great circles.

Lemma 2. *$n \in$ (int)sph$\{n_1, \ldots, n_k\}$[2] if and only if $<n, y> \leq 0 (< 0)$ for all $y \in C - \{0\}$.*

Proof: Observe that $y \in C - \{\mathbf{0}\}$ if and only if the hyperplane H_y through $\mathbf{0}$ normal to y supports the n_i's; as in both cases $<y, n_i> \leq 0$ for all n_i.

Picking any $y \in C - \{\mathbf{0}\}$ and assuming that $n \in$ (int)sph$\{n_1, \ldots, n_k\}$, because H_y supports sph$\{n_1, \ldots, n_k\}$, it must be that $<y, n> \leq 0$ (< 0).

Similarly if $<p, n> \leq 0$ (< 0) for all $p \in C - \{\mathbf{0}\}$,then taking any hyperplane H_y with normal y which is a supporting hyperplane for the n_i's, we have that $y \in C - \{\mathbf{0}\}$. Thus $<n, y> \leq 0$ (< 0) for all such supporting hyperplanes which implies that $n \in$ (int)sph$\{n_1, \ldots, n_k\}$. □

1.1.2 Admissibility

In order to classify admissibility we are interested in where the equator plane (through c parallel to $\Re^d$) or its associated great circle lies in relation to P^* the dual spherical polyhedron. Consider the point k on the sphere

[2] Notice that we could equally well say $n \in$ (int)Cone$\{n_1, \ldots, n_k\} = C$.

where the normal to the equator plane at c meets the sphere, chosen so that $\vec{ck}$ is the outer normal to the halfspace containing $\Re^d$.

Let hyperplanes $H_1, \ldots, H_n$ lie in $\Re^d$. Then $H_1, \ldots, H_n$ bound a polyhedron P with n facets only if the equivalent points $p_1, \ldots, p_n$ are in convex position on the sphere with an admissible spherical hull P', that is one whose dual P^* has no facet lying entirely in the closed upper hemisphere. P' is defined to be *admissable* if one of the following three cases occur.

(i) When P^* does not meet the equator and lies entirely in the lower hemisphere P is bounded. The lemma above tells us that this case occurs if and only if $k \in \text{int sph}\{p_1, \ldots, p_n\}$.

(ii) When P^* meets the equator, yet does not enter the open upper hemisphere P is unbounded, this occurs if and only if $k \in \text{bd sph}\{p_1, \ldots, p_n\}$ by lemma 2 also.

(iii) The only other admissible case occurs when P^* does meet the upper hemisphere, yet none of its facets lie entirely in the upper hemisphere in this case $H_1, \ldots, H_n$ and the equator plane form a cone with $(n+1)$ facets. Here again P is unbounded. This occurs if and only if k is a vertex of the $(n+1)$-gon $\text{sph}\{p_1, \ldots, p_n, k\}$, by lemma 1.

In any other case P' is not admissible.

1.2 An Upper Bound On The Number Of Rich Cells

1.2.1 A Recurrence Relation

Let $H = \{H_1, H_2, \ldots, H_n\}$ be n hyperplanes in $\Re^d$ and consider the arrangement $\mathcal{A}(H)$ of these hyperplanes. Define $f_{\mathcal{A}}(d, n)$ to be the number of rich cells in $\mathcal{A}(H)$, we want to determine $f(d, n)$ the maximum number of rich cells over all such arrangements.

Claim 3. *$f(d, n) \leq f(d, n-1) + f(d-1, n-1)$, for $d, n \geq 2$.*

Proof: Firstly consider the contribution of H_n. A rich cell of $\mathcal{A}(H)$ can only occur when H_n cuts a rich cell of $\mathcal{A}(H - H_n)$. H_n can cut such a cell into at most two rich cells of $\mathcal{A}(H)$.

If some hyperplane H_k is parallel to H_n then no region on H_n can act as a facet of two rich cells as H_k lies uniquely in H_n^+ say, and cannot bound a facet in H_n^-. Hence:

$$f_{\mathcal{A}}(d,n) \leq f(d,n-1) \leq f(d,n-1) + f(d-1,n-1).$$

Otherwise, say H_n divides a rich cell C of $\mathcal{A}(H - H_n)$ into two rich cells C_A and C_B of $\mathcal{A}(H)$. Then some region R of H_n is a shared facet of C_A and C_B, R lies in the $(d-1)$-flat H_n, and R must have $(n-1)$ facets, as follows:

If the flats $H_i \cap H_n$, $i = 1, \ldots, (n-1)$ are distinct then, if R has less than $(n-1)$ facets then some $H_k \cap H_n$, $(1 \leq k < n)$ does not support R. H_n does not cut the facet of cell C due to H_k. This facet must lie uniquely in one halfspace H_n^+ say, but then H_k can support a facet of only one of the cells C_A or C_B.

If two of the flats coincide in H_n then the corresponding facets of C must be separated by H_n (as H_n divides C and C is convex). Again the facet due to some H_k lies uniquely in H_n^+ and the above argument holds.

Thus R has $(n-1)$ facets and so is a rich cell in the arrangement of the $(d-2)$-flats $H_k \cap H_n$, $k = 1, \ldots, n-1$, lying in H_n. There can be at most $f(d-1, n-1)$ such regions R in H_n.

All other rich cells in $\mathcal{A}(H - H_n)$ can yield at most one rich cell in $\mathcal{A}(H)$.

$$f_{\mathcal{A}}(d,n) \leq f(d,n-1) + f(d-1,n-1)$$

□

1.2.2 Boundary Conditions

The results when $d = 1$ are obvious: $f(1,1) = 2$, $f(1,2) = 1$, and $f(1,k) = 0$ whenever $k \geq 3$. Also it is clear that $f(d,1) = 2$ for all d. We could use this as the starting position for the recurrence, but a better bound can be obtained if a few more cases are investigated.

Claim 4. *Whenever* $1 \leq k \leq d$, $f(d,k) = 2^k$.

Proof: The proof is by induction. We have observed that $f(1,1) = 2$ and $f(d,1) = 2$ for all d so the result holds for $d = 1$, and when $k = 1$. Assuming that the result holds in up to $(d-1)$ dimensions for all $k \leq d-1$, and in d dimensions for some $k < d$ planes, the recurrence relation yields:

$$\begin{aligned} f(d,k+1) &\leq f(d,k) + f(d-1,k) \\ &= 2^k + 2^k \\ &= 2^{k+1}. \end{aligned}$$

So $f(d,k) \leq 2^k$, and this bound is realised by an arrangement of $k \leq d$ mutually orthogonal hyperplanes in $\Re^d$, which has 2^k rich cells. □

Claim 5. *When* $d \geq 2$

$$\begin{aligned} f(d,d+1) &= \binom{d+1}{d+1} + \binom{d+1}{d} + \cdots + \binom{d+1}{2} \\ &= 2^{d+1} - d - 2. \end{aligned}$$

Proof: Consider the arrangement $\mathcal{A}$ of $(d+1)$ hyperplanes in $\Re^d$.
(i) If the hyperplanes lie in general position, that is no two planes are parallel and no $(d+1)$ hyperplanes have a point in common then the planes will bound a simplex in $\Re^d$:

As in figure 1.1, place a copy of the sphere S^d on top of $\Re^d$ (so that they meet at a single point), and project the $(d+1)$ hyperplanes on to great circles of S^d in the obvious way. The arrangement of the hyperplanes in $\Re^d$ corresponds to the subdivision which lies on the lower (open) hemisphere of S^d.[3] Let the boundary of this hemisphere be called the ***equator***. The surface of S^d is divided into 2^{d+1} spherical simplices each determined by a sequence of $(d+1)$ plus and minus signs (signifying positive or negative hemispheres/halfspaces) [17, p.304]. We can assume that no vertex of the subdivision lies on the equator (as this implies that two planes are parallel). Picking any one of the 2^{d+1} simplices consider its identifying sequence $\sigma_1, \ldots, \sigma_{d+1} \in \{+,-\}^{d+1}$, associate with each σ_i the vertex v_i of the simplex

[3]Each point $\mathbf{x} \in \Re^d$ maps to $[\mathbf{x}, \mathbf{c}] \cap S^d$, where $\mathbf{x}$ is written as a point in $\Re^{d+1}$, and $\mathbf{c}$ is the centre of S^d in $\Re^{d+1}$.

so that v_i does not lie on the ith great circle. If v_i is in the lower hemisphere let $\bar{v}_i = v_i$ and $\bar{\sigma}_i = \sigma_i$. If v_i is in the upper hemisphere take $\bar{v}_i$ to be the point which is antipodal to v_i and take $\bar{\sigma}_i = -\sigma_i$. Then $\bar{\sigma}_1, \ldots, \bar{\sigma}_{d+1}$ describe one of the spherical simplices all of whose vertices $\bar{v}_1, \ldots, \bar{v}_{d+1}$ lie in the open lower hemisphere. Hence in $\Re^d$ we can find a unique[4] simplex which is bounded by the $(d+1)$ hyperplanes.

The number of rich cells in such an arrangement is as follows:- The simplex itself is a rich cell. The space outside the simplex is divided into unbounded cells that 'hang from' each of the faces. Each $(d-k)$-face $1 \leq k < d$ of the simplex is formed by the intersection of k of the hyperplanes and is itself a $(d-k)$ simplex in other words a further $(d-k+1)$ of the hyperplanes support its boundary. Hence the region outwards from each such face is bounded by all $(d+1)$ hyperplanes and there are $\begin{pmatrix} d+1 \\ d-k+1 \end{pmatrix}$ such faces [24, p.53]. The region off each vertex (when $k = d$) is supported by only d hyperplanes and so is not rich. Thus

$$f(d, d+1) = \begin{pmatrix} d+1 \\ d+1 \end{pmatrix} + \begin{pmatrix} d+1 \\ d \end{pmatrix} + \cdots + \begin{pmatrix} d+1 \\ 2 \end{pmatrix}.$$

(ii) If any two of the hyperplanes in $\mathcal{A}$ are parallel then each of the parallel planes cannot divide a cell rich in the remaining $(d-1)$ planes into two cells rich in d planes. So in this case there can be no more than $f(d, d-1)$ rich cells and $f(d, d-1) = 2^{d-1} \leq 2^{d+1} - d - 2$ whenever $d \geq 2$.
(iii) Lastly consider the case when the hyperplanes of $\mathcal{A}$ have a point in common.

In $\Re^2$ if three lines have a point in common then there are no rich cells.

In $\Re^3$, if any three of the planes have a line in common then their arrangement is equivalent to three lines with a point in common in two dimensions, and has no rich cells. Otherwise three of the planes have exactly one point in common, this arrangement has (at most) 2^3 rich cells by claim 4, the addition of the fourth plane cannot divide any of these rich cells into two new rich cells as this can occur at most zero times (this is the number of rich cells when three lines in the plane have a point in common), and $f_{\mathcal{A}}(3,4) \leq 2^3 \leq 2^4 - 3 - 2$. The case for $\Re^4$ follows similarly as $f_{\mathcal{A}}(4,5) \leq f(4,4) + f_{\mathcal{A}}(3,4) \leq 2^4 + 2^3 \leq 2^5 - 4 - 2$.

[4] Any change in the identifying sequence will flip at least one vertex into the upper hemisphere.

The general case is not actually required here as only the result $f(3,4)$ is used; however for completeness a sketch of the general case follows. Let $d \geq 5$ and let $\mathcal{A}$ be an arrangement of $(d+1)$ hyperplanes in $\Re^d$ that have a point in common. There are two cases.
(a) If the point of intersection is unique, then some set $\mathcal{A}'$ of d of the hyperplanes meets in a unique point. Consider the addition of the $(d+1)$st plane to this arrangement. Observe that the $(d+1)$st plane must miss at least two of the rich cells of $\mathcal{A}'$; if this were not true it would imply that the hyperplane was fully dimensional. This means that these two cells cannot yield rich cells in the final arrangement: at most $(2^d - 2)$ of the rich cells of $\mathcal{A}'$ can yield rich cells of $\mathcal{A}$. Inductively assume the claim holds for the arrangement in the $(d+1)$st hyperplane, then

$$\begin{aligned} f_{\mathcal{A}}(d, d+1) &\leq (2^d - 2) + 2^{(d-1)+1} - (d-1) - 2 \\ &= 2^{d+1} - d - 3 \end{aligned}$$

so the result holds.
(b) Otherwise the $(d+1)$ hyperplanes have a k flat in common for some k with $1 \leq k \leq d-2$. All of the cells of this arrangement are cones out from the k-flat; all other faces of $\mathcal{A}$ have dimension strictly larger than k. Pick a $(d-k)$ flat perpendicular to the k flat, this meets all faces of $\mathcal{A}$ and it meets the k flat in a unique point; the $d+1$ hyperplanes meet it in $(d-k-1)$ dimensional 'hyperplanes'. Observe that any rich cell in $\mathcal{A}$ must be rich in this $(d-k)$ flat. If we count the number of rich cells here then we have a bound on the number of rich cells in $\mathcal{A}$.

In a $(d-k)$ flat we have $d+1$ 'hyperplanes' through a unique point. Inductively assume that the arrangement of any $(d-k+1)$ of these has at most $2^{d-k+1} - d - 2 + k$ rich cells. Adding each of the remaining k 'hyperplanes' can at most double the number of rich cells.

$$\begin{aligned} f_{\mathcal{A}}(d, d+1) &\leq 2^k(2^{d-k+1} - (d+2-k)) \\ &= 2^{d+1} - d - 2 - (2^k - 1)(d+2-k) + k \\ &\leq^5 2^{d+1} - d - 2 - 4 \cdot (2^k - 1) + k \\ &= 2^{d+1} - d - 2 - (2^k - k) - (3 \cdot 2^k - 4) \\ &\leq 2^{d+1} - d - 2 \end{aligned}$$

Throughout this case it is important to note that when we drop a dimension there may be less than d distinct 'hyperplanes', however in each case the bound is safe. The result holds. □

Corollary 2. $f(3,4) = 11$ *and* $f(4,5) = 26$.

Results for 2 dimensions

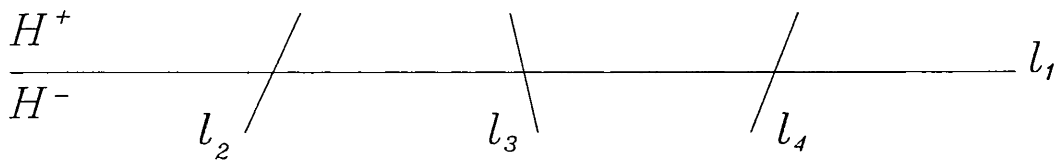
It has already been shown that $f(2,1) = 2$ and $f(2,2) = 4$ by claim 4, and that $f(2,3) = 4$ by claim 5. Case analysis gives $f(2,4) = 2$.

Lemma 6. $f(2,5) = 1$.

Proof: Using the recurrence relation $f(2,5) \leq f(2,4) + f(1,4) = 2$, and certainly $f(2,5) \geq 1$, so we assume two rich cells and obtain a contradiction.

Consider lines $l_1, \ldots l_5$ in $\Re^2$, and let the arrangement of these lines have two rich cells. At least one of the lines l_1 say, supports a bounded facet on each of the rich cells. Note that here no two rich cells can share a common facet as this occurs at most $f(1,4) = 0$ times, using the argument in claim 3.

(1) If the rich cells have facets which share a common point on l_1 then let the facets be bounded by l_2, l_3 and l_4 as shown.



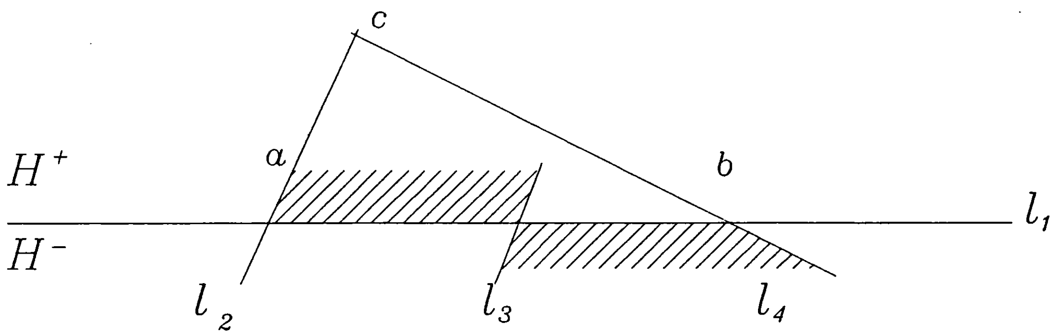
The cells must lie on opposite sides H^+ and H^- of l_1 or else they would share a common facet along l_3.

The fifth line l_5 cannot be parallel to l_1 or it would not be possible to have rich cells in both H^+ and H^-, so l_5 must cross l_1 outside the line segment $[a, b]$ on l_1 determined by the two facets.

(i) Let l_2 and l_4 meet at c in H^+ say, Then one of the rich cells i.e. that in H^+ must be bounded (it lies inside $\triangle\{a, b, c\}$), call this cell C_1.

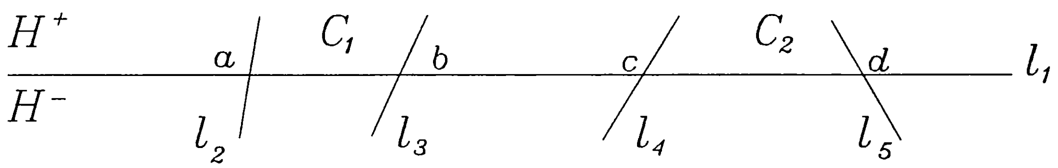
[5] $1 \leq k \leq d - 2$ implies that $d + 2 - k \geq 4$.

(i) Let l_2 and l_4 meet at c in H^+ say, Then one of the rich cells i.e. that in H^+ must be bounded (it lies inside $\triangle\{a,b,c\}$), call this cell C_1. Because



l_5 forms a facet of C_1 it must enter the triangle, yet it avoids $[a,b]$; this means that l_5 crosses both l_2 and l_4 in H^+. Now l_5 cannot again enter the cone apex c through $[a,b]$, as this would contradict the intersection of convex sets being convex. Therefore l_5 meets the cone only in H^+ say. The cone contains both C_1 and C_2, yet l_5 cannot form a facet in H^-, which is a contradiction.
(ii) Similarly when l_4 is parallel to l_2: Line l_5 cannot also be parallel to l_2, so l_5 cuts both l_2 and l_4, but cannot cross $[a,b]$, l_5 must intersect both of these lines in one halfspace H^+ say. The cell in H^- cannot be rich.

(2) If the facets of the two rich cells are separated along l_1 simple case analysis[6] shows that no pair of lines can be parallel. Let the lines meet l_1 as shown, with $l_1 \cap l_2 = a$, $l_1 \cap l_3 = b$, $l_1 \cap l_4 = c$, $l_1 \cap l_5 = d$, and let $l_2 \cap l_3 = p$ and $l_4 \cap l_5 = q$.



Then cell $C_1 \subseteq K_1$, where K_1 is the cone with apex p through $[a,b]$.

In order to form a facet of C_1 l_4 and l_5 must each intersect K_1 in a closed line segment or a half-line. If $q \notin \text{int}K_1$ then it can't be that both l_4 and l_5 are facets of C_1. So $q \in \text{int}K_1$, and symmetrically $p \in \text{int}K_2$ where

[6]Note that both rich cells must lie between any pair of parallel lines.

K_2 is the cone apex q through $[c, d]$. This situation cannot exist.[7] □

Corollary 3. *$f(2, k) = 1$ whenever $k \geq 5$.*

1.2.3 An Upper Bound

Theorem 7. *For $n \geq d \geq 3$*

$$f(d, n) \leq \frac{(n+8)^{d-2}}{(d-2)!}.$$

Proof: The result holds for $d = 3$ by using the recurrence relation on the results for $f(3, 4)$ and $d = 2$. It can be shown (using an inductive argument) that the result holds for $f(d, d)$ i.e. that for $d \geq 3$

$$f(d, d) = 2^d \leq \frac{(d+8)^{d-2}}{(d-2)!}.$$

Inductively assume then that the result holds true in $(d-1)$ dimensions, and that in d dimensions the result holds for up to $(n-1)$ hyperplanes then

$$\begin{aligned} f(d, n) &\leq f(d, n-1) + f(d-1, n-1) \\ &\leq \frac{(n+7)^{d-2}}{(d-2)!} + \frac{(n+7)^{d-3}}{(d-3)!} \\ &\leq \frac{((n+7)+1)^{d-2}}{(d-2)!} \\ &= \frac{(n+8)^{d-2}}{(d-2)!}. \end{aligned}$$

□

Obtaining a strict value for $f(3, 5)$, and thus an improvement for $d = 3$, might enable the constant (8) in the above expression to be brought down to a best possible value of 3. This is the best possible such result that would be consistent with $f(d, d)$, $d \geq 4$, and with $f(4, 5)$.

[7]The half-line $l_3 \cap K_1$ lies between intK_1 and the points c and d. Neither l_4 nor l_5 passes through p (the apex of K_1) as such a line cannot be a facet of C_1, therefore they both pass through $l_3 \cap K_1$, so $l_3 \cap K_2$ is a closed line segment (which does not contain p). This implies that C_2 is bounded in which case l_2 and l_3 must form adjacent facets of C_2, so $p \in K_2$, which is a contradiction.

1.3 A Lower Bound

The following example was discovered by Janos Pach and Imre Bárány, and gives a lower bound on $f(d,n)$. The method involves constructing an arrangement in space one dimension higher than that required and intersecting this with a hyperplane to obtain an arrangement in $\Re^d$. First consider an arrangement of $n+1$ hyperplanes through the origin in $\Re^{d+1}$. Let $H_i = \{x : <a_i, x> = 0, a_i \in \Re^d\}$. These planes dissect the space into cones $C = \{x \in \Re^d : <\varepsilon_i a_i, x> \leq 0, i = 1, \ldots, n+1\}$, where $\varepsilon_i = \pm 1$ is a sign sequence. Such a cone is rich if it has $n+1$ facets. By lemma 1 The cone determined by $\varepsilon_1, \ldots, \varepsilon_{n+1}$ is rich if and only if Cone$\{\varepsilon_1 a_1, \ldots, \varepsilon_{n+1} a_{n+1}\}$ has $n+1$ extreme rays.

Lemma 8. $f(d,n) \geq \binom{n}{d-2} + \binom{n}{d-3} + \cdots + \binom{n}{0}$, *where* $n \geq d+2$.

Proof: Let $n \geq d+2$ and take $n+1$ points on the moment curve $M(t) = (t, t^2, \ldots, t^{d+1})$ in $\Re^{d+1}$. Let the points $M(t_i)$ have parameters $0 < t_1 < \cdots < t_{n+1}$. Consider the arrangement $\mathcal{A}$ of the hyperplanes $H_i = \{x : <x, M(t_i)> = 0\}$. A hyperplane H will be found that intersects this arrangement to give an arrangement in d dimensions with the required number of rich cells.

The arrangement $\mathcal{A}$ consists of regions each of which is defined by a choice H_i^- or H_i^+ of halfspace for each i. This corresponds to a choice of outward normal of $M(t_i)$ or $-M(t_i)$ respectively. The regions so formed are all cones apex the origin in $\Re^{d+1}$. For certain choices of normal

$$\{M(t_i), -M(t_j) : i \in A, j \in B, A \cup B = \{1, \ldots, n+1\}, A \cap B = \emptyset\}$$

the half rays through the points $M(t_i)$ and $-M(t_j)$ are in convex position.

Consider Π a general hyperplane through the origin in $\Re^{d+1}$, If the normal to Π is $(a_0, \ldots, a_d)$, then Π meets the moment curve at points where $t(a_0 + a_1 t + \cdots + a_d t^d) = 0$ in other words at $t = 0$ and at the roots of the polynomial $P(t) = \sum_0^d a_r t^r$. The points $M(t_i)$ lie in Π^+,Π^- or Π according to whether $P(t_i)$ is positive, negative or zero. Any such polynomial $P(t)$ has at most d roots and partitions the real line into at most $(d+1)$ regions where Sign$(P(t))$ is alternately positive and negative.

Consider a partition of the real line into at most $(d-1)$ open intervals as follows. Let $I_1, \ldots I_{d-1}$ denote intervals $(-\infty, \alpha_1), (\alpha_1, \alpha_2), \ldots, (\alpha_{d-2}, \infty)$

respectively, where $t_i \neq \alpha_j$ for any i, j, and $t_1 \in I_1$. If $t_i \in I_m$ and m is odd then we pick normal $M(t_i)$ and let $i \in A$. We pick $-M(t_j)$ and $j \in B$ otherwise. The normal $M(t_1)$ is picked.

For each k, if $t_k \in I_m$, $m = 1, \ldots, d-2$ then let x lie between t_k and $\min\{\alpha_m, t_{k+1}\}$, if $t_k \in I_{d-1}$ then if $k = n+1$ let $x > t_{n+1}$, else let $x \in (t_k, t_{k+1})$. Then the polynomial

$$P_k(t) = (-1)^d (t-\alpha_1)\cdots(t-\alpha_{d-2})(t-t_k)(t-x) = \sum_0^d a_r^k t^r$$

of degree d has the property that $P_k(t_k) = 0$ and $P_k(t_i) > 0$, $k \neq i \in A$, and $P_k(t_j) < 0$ if $k \neq j \in B$. Thus we have normal $n_k = (a_0^k, \ldots, a_d^k)$ with

$$\begin{aligned} <n_k, M(t_i)> &> 0 \\ <n_k, -M(t_j)> &> 0 \\ <n_k, M(t_k)> &= 0. \end{aligned}$$

The half rays from the origin through $M(t_i)$, and $-M(t_j)$, $i \in A$ $j \in B$, are in convex position. (So lemma 1 implies that the arrangement in $\Re^{d+1}$ has a region bounded by all $n+1$ hyperplanes, and identified by this choice of outer normals.)

First we consider how many possible choices of normals, defined by all such polynomials, there are. Then the arrangement in $\Re^d$ will be constructed. Effectively we have to pick roots of a polynomial of degree $(d-2)$ so that they lie in the intervals $(t_1, t_2), \ldots, (t_n, t_{n+1})$. It is optimal to pick at most one root per interval, any other roots are deemed to lie in (t_{n+1}, ∞). There are n intervals and up to $d-2$ roots are picked. The number of ways of doing this is:

$$\binom{n}{d-2} + \binom{n}{d-3} + \cdots + \binom{n}{0}$$

Take a sphere radius one through the origin, the points where each of the chosen sequences of normals cut the sphere lie in spherically convex position. Now using the argument from section 1.1.2, consider H_1 to be the equator plane. If we intersect the remaining hyperplanes $H_2, \ldots, H_{n+1}$ with $H = \{x : <M(t_1), x> = -1,\ x \in \Re^{d+1}\}$ all the sequences we have picked are admissible, and in H they yield unbounded cells each with n facets, hence there are at least the required number of rich cells. □

This example provides a lower bound for $f(d,n)$, which is of the same order as the upper bound found in the previous section. Hence:

$$\sum_{k=0}^{d-2} \binom{n}{k} \leq f(d,n) \leq \frac{(n+8)^{d-2}}{(d-2)!.}$$

All the rich cells picked in this example are unbounded. It may seem that such a large number of bounded rich cells cannot exist. However this is untrue. In fact there exist arrangements with as many as

$$\binom{n-d-1}{d-2} + \binom{n-d-1}{d-3} + \cdots + \binom{n-d-1}{0}$$

bounded rich cells, the construction is as follows.

Using a similar argument construct an arrangement of n hyperplanes through the origin in $\Re^{d+1}$. Let the normals to these hyperplanes be points on the moment curve with parameters $t_1 < \cdots < t_n$. We need to find a hyperplane H that will cut this arrangement to give a large number of bounded cells. First notice that if we fix the first $d+1$ outer-normals to be $M(t_1), \ldots, M(t_{d+1})$, then the set $\mathcal{C}$ of cones that will be considered lie in $H_1^-, \ldots, H_{d+1}^-$. In other words $< M(t_i), x > \leq 0$ for all $x \in \mathcal{C}$ and $i = 1, \ldots, d+1$. So if we let $n = (M(t_1) + \cdots + M(t_{d+1}))/(d+1)$, then $<n, x>< 0$ for all $x \in \mathcal{C} - \{0\}$.[8] Thus by lemma 2 n lies in the interior of the cone of each choice of normals.

We have to pick at most $d-2$ roots of a polynomial. We pick the roots to lie among the intervals $(t_{d+1}, t_{d+2}), \ldots (t_{n-1}, t_n)$. The number of ways of doing this is:

$$\binom{n-d-1}{d-2} + \binom{n-d-1}{d-3} + \cdots + \binom{n-d-1}{0}.$$

This means that if we take $H = \{x \in \Re^{d+1} : <n, x> = -1\}$ then combining the arguments used in section 1.1.2 and the previous example each of the cones picked will yield a bounded rich cell in H.

[8]The inner products can't all be zero at one point (other than the origin) as this would imply a polynomial of degree d having $d+1$ roots.

1.4 A Carathéodory Number For Lines In The Plane

First recall that:

Corollary 3. *For $n \geq 5$, if n lines in the plane are in convex position then they bound a unique polyhedron (with n facets).*

1.4.1 Notation

The notation introduced here will be used throughout the following argument.

Given n lines $l_1, \ldots, l_n$ in the plane P_i is defined to be a polyhedron bounded by $\{l_j : j \neq i\}$ provided of course that this exists, with P_i' being the equivalent choice of normals e.g. $p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_n$ on the sphere. Notice that the points are in convex position on the sphere and that P_i' must be admissible. Also note that these objects will be unique for $n \geq 6$.

For brevity the term n-poly will be used to denote a polyhedron with n facets which could have n or $n-1$ vertices depending whether the polyhedron is bounded or not. The term n-gon will denote a convex polygon with n vertices.

1.4.2 Assumptions

Throughout the following argument we assume that the lines are in general position in the plane so no three lines have a point in common and no two lines are parallel. Though general position is assumed the result is in fact valid for a general set of n distinct lines in the plane as follows:

If three lines pass through a point they do not bound any triangle thus some five of the lines do not lie in convex position. (Any cell bounded by five lines must be contained in a cell bounded by any three of these lines.) Similarly when more than two lines are parallel in any given direction.

If some pair(s) of lines are parallel we can use a perturbation argument to move the lines into general position. The lines can be perturbed so that they meet far away from any other intersection point. Let every five lines in the original arrangement bound a 5-poly. The existence of 5-polys not involving both the parallel lines is unaffected by the perturbation. Note that any polyhedron with a pair of parallel lines in its boundary must lie between them. In the relevant quadrant formed by the two perturbed lines no edges are destroyed or created (therefore no new 5-polys are created elsewhere as a 5-poly is unique). So in the perturbed arrangement every five lines bound a 5-poly. Theorem 10 below says that there is an n-gon in the perturbed arrangement, and it must lie in the cone between the two lines, so because we haven't added any edges to regions in this cone there is an n-gon in the original arrangement.

It follows from these assumptions that no three of the normals on the sphere lie on the same great circle. If this was the case then the three great circles normal to them all meet at some point p and the point $\bar{p}$ antipodal to this. If one of the three points is k the other two great circles meet on the equator and the two equivalent lines are parallel. If none of the points is k the three lines will all meet at a point in the plane. So, in two dimensions we can ignore the second admissibility condition defined on page 12. Thus P' is admissible if either k is in the interior of the spherical n-gon formed by $\{p_1, \ldots, p_n\}$ or if k is a vertex of the $(n+1)$-gon $\mathrm{sph}\{p_1, \ldots, p_n, k\}$.

1.4.3 The Result For Two Dimensions

Certainly there is a Carathéodory number for lines in the plane and it is at most six. This can be shown inductively by exploiting the fact that for $n \geq 7$ every $n-2$ lines bound a unique 5-gon. The argument follows easily either from the geometry of the arrangement in the plane, or by considering normals on the sphere, as in the proof of theorem 10 on page 30.

Six is the best possible result for the equivalent projective problem; on the sphere S^2 every five great circles in general position bound some spherical 5-gon, but this is not true for six great circles [21]. However in the affine problem it is trivial that every four lines in general position in the plane bound a 4-poly (e.g. take one of the great circles in S^2 to be the equator), yet this is not true for five lines.

A set of lines in the plane lie in *projectively convex position* if there is a permissable projective transformation from the lines onto a set of lines in convex position. Figure 1.5 illustrates the above observation. The six lines are not in projectively convex position. The five solid lines are in projectively convex position (the shaded region is mapped to a 5-poly by any projective transformation taking the dotted line to infinity) but not in

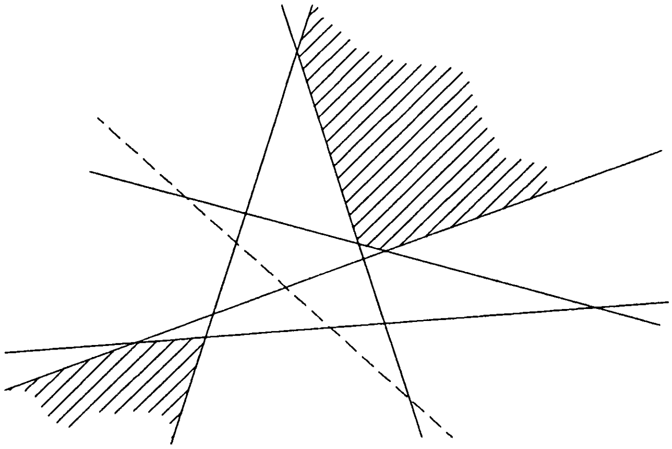


Figure 1.5: An arrangement of six lines that illustrates properties of convex and projectively convex position.

convex position.

The best possible result for the Carathéodory number for lines in the plane is therefore five. This will be proved below.

Theorem 9. *If an arrangement of six lines* $l_1, \ldots, l_6$ *in general position in* $\Re^2$ *has every five of the lines in convex position then the six lines are in convex position.*

Proof:

Remark 1: Either the six lines bound a 6-poly, or there is a 5-poly in the arrangement.

Proof of remark 1: Consider one of the 5-polys P_6 say, if l_6 misses P_6 then P_6 is a 5-poly in the arrangement of the six lines.

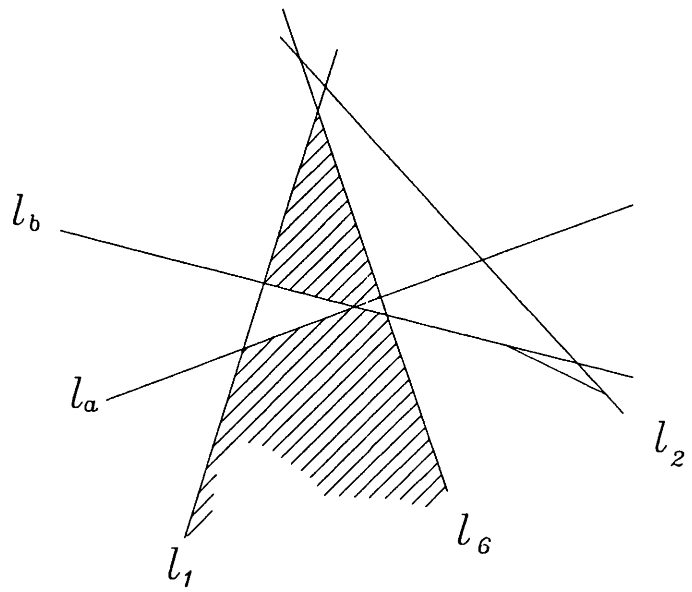
Consider the ways that l_6 can cut P_6 (any region thus formed lies in the arrangement).

If P_6 is bounded then P_6 has five vertices and l_6 must cut P_6 in a bounded line segment adding two new vertices. If l_6 splits the vertices of P_6 in the ratio 4:1 it creates a hexagon (and a triangle). If the ratio is 3:2 a pentagon is created (and a quadrilateral).

If on the other hand P_6 is unbounded then it has four vertices.
(i) If l_6 cuts P_6 in a line segment adding two vertices, it splits P_6 into one bounded and one unbounded polyhedron. If l_6 divides the vertices of P_6 in the ratio 3:1 it creates a region with five vertices this could be a bounded

pentagon or an unbounded 6-poly. If it divides the vertices in the ratio 2:2 it creates a bounded quadrilateral and an unbounded 5-poly each having four vertices.

(ii) If l_6 cuts P_6 in an unbounded line segment adding one new vertex two new unbounded polyhedra are formed. If the ratio of vertices on either side of l_6 is 3:1 one side will have four vertices and thus will be a 5-poly. The ratio 2:2 is a special case, let the unbounded edges of P_6 be l_1 and l_2, if a point moving along l_6 from the unbounded direction of P_6 meets l_1 say before l_2, then bearing in mind that four lines in general position have a unique arrangement subject to rotation we can find five lines not in convex position as illustrated below.



□

As an alternative it is also possible to make a slightly stronger remark.

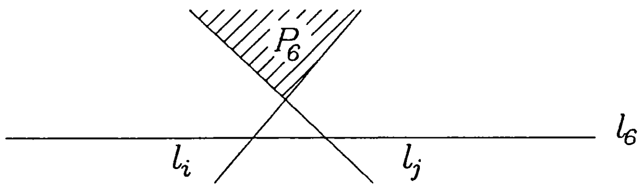
Remark 2: Either the six lines bound a 6-poly or there is a bounded 5-poly in the arrangement.

Proof of remark 2: Consider P_6. If P_6 is bounded we are done: As in remark 1 either P_6 lies in the arrangement or l_6 cuts P_6 to create a bounded pentagon or hexagon.

So assume that P_6 is unbounded. We use map μ_2 to project the arrangement of lines onto the sphere. Because P_6 is unbounded we can assume $\{k, p_1, \ldots, p_5\}$ lie in convex position on the sphere. Note that this means $\{p_1, \ldots, p_5\}$ are in convex position also. Choose p_6 so that it lies in some same hemisphere as these points. Then:
(i) If $p_1, \ldots, p_6$ are in convex position either we have an admissible 6 - gon or adapting Carathéodory's Theorem for the sphere[9] we conclude that $p_6 \in \text{sph}\{k, p_1, p_2\}$ say. Thus $p_6 \in \text{int sph}\{k, p_1, p_2, p_3, p_4\}$ so by lemma 2 the line l_6 misses the quadrilateral determined by p_1, p_2, p_3, and p_4 (which is unbounded), this means that P_5 must lie in a different quadrilateral bounded by lines $l_1, \ldots, l_4$ to that containing P_6.

Consider the arrangement of four lines in general position. This arrangement is unique subject to rotation, and it has exactly two rich cells one of which is bounded and one is not. So as P_6 lies in the unbounded quadrilateral P_5 must lie inside the bounded quadrilateral and must itself be bounded, and the previous argument holds.
(ii) If $p_1, \ldots, p_6$ are not in convex position then either $p_6 \subseteq \text{sph}\{p_i, p_j, p_k\}$ or $p_i \subseteq \text{sph}\{p_6, p_j, p_k\}$ These cases are analogous: in each case we can find some P_l which lies in a different 4-poly to P_6, so as P_6 lies in the unbounded quadrilateral P_l must lie in the bounded quadrilateral and so is bounded, and so on. □

To prove the assertion we need only show that if there is a pentagon in the arrangement of the six lines then some five lines are not in convex position. Let this pentagon be P_6 then P_6 lies in the half-space l_6^+ say. Using the notation $p_{ij} = l_i \cap l_j$ we pick p_{ij} to be the closest vertex to l_6 lying in l_6^+.



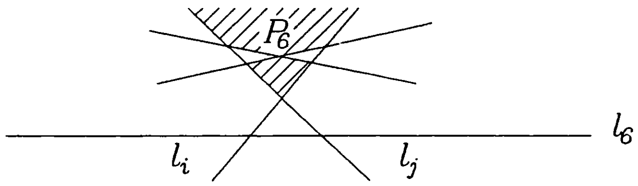
Then the triangle $\{p_{ij}, p_{i6}, p_{j6}\}$ lies in the arrangement, and P_6 must lie in

[9]Observe by projection that the same Carathéodory's theorem that applies to points in $\Re^d$ applies to points on some hemisphere of S^d.

the quadrant K bounded by lines l_i and l_j which lies away from l_6 (else some line would have to cut $[p_{ij}, p_{i6}]$ or $[p_{ij}, p_{j6}]$ contradicting the choice of p_{ij}).

Consider the order of the edges of P_6. The proof is split into cases depending on the number of edges that lie between the edge due to l_i and that due to l_j, where the edges counted are those which lie on that side of P_6 which is closer to l_6.

Case (a): If there are at least two such edges supported by l_a and l_b then both quadrilaterals due to l_i, l_j, l_a and l_b lie in K and so l_6 misses them. Hence these are five lines not in convex position.



Case (b): If there are no such edges.

bi) Then if one of the other lines l_a say, meets K in a bounded line segment. Due to the choice of i and j, l_a meets l_6 outside $[p_{i6}, p_{j6}]$. Now pick a different line l_b which is adjacent to either l_i or l_j in P_6. Say without loss of generality that l_b is adjacent to l_i in P_6 then l_b crosses $\Delta\{p_{ij}, p_{aj}, p_{ai}\}$, but l_b can't meet $[p_{ij}, p_{ja}]$ as this would prevent l_a forming an edge of P_6 so p_{ab} is in K, see figure 1.6.

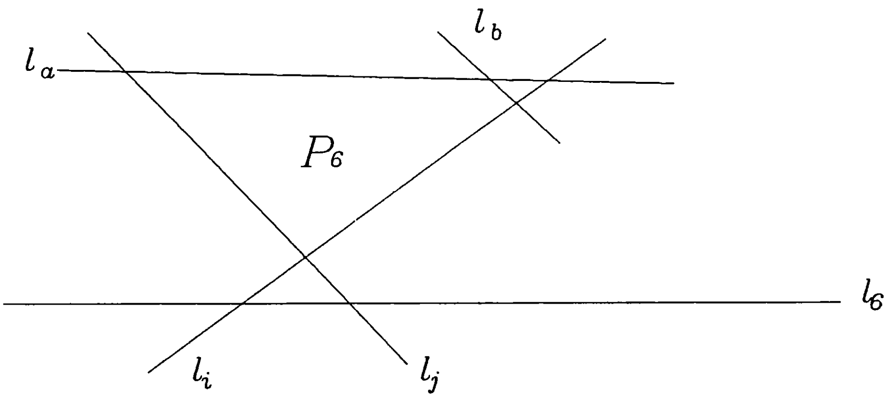


Figure 1.6: Theorem 9 case bi) five lines not in convex position.

The line l_b cuts only one of the quadrilaterals Q formed by l_a, l_i, l_j and

l_6 entering it through l_i at p_{ib}. In Q l_i is adjacent to l_a and l_j, yet l_b cannot leave Q at l_a as $p_{ab} \in K$, or at l_j as it cannot cut $[p_{ij}, p_{j6}]$ thus these five lines cannot be in convex position as l_b does not cut Q to form a 5-poly and misses the other quadrilateral—so no 5-poly can be formed there.
bii) If none of the other three lines meet K in a bounded line segment, then two of them must cross l_6 on the same side of $[p_{i6}, p_{j6}]$. These five lines are not in convex position as illustrated in figure 1.7.

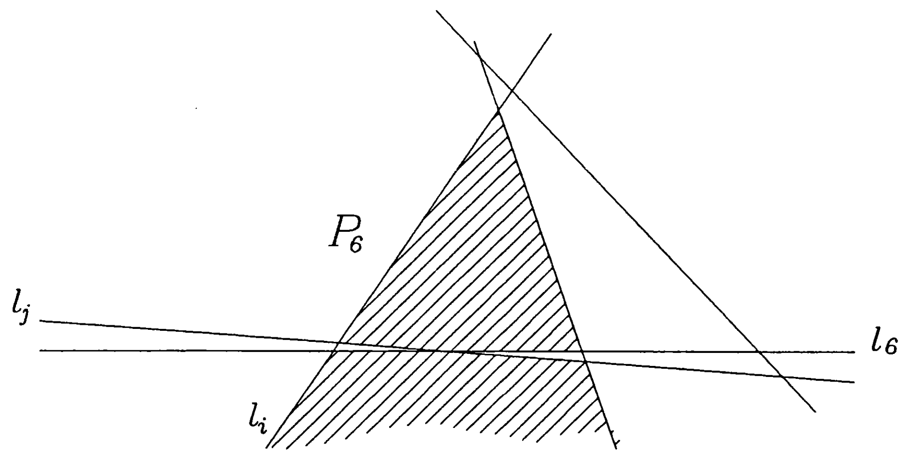


Figure 1.7: Theorem 9 case bii) five lines not in convex position.

Case (c): If there is one edge formed by line l_c between l_i and l_j, the other lines being l_a and l_b then:
ci) If either of the lines l_a and l_b, l_a say, meets K in a line segment then the argument used in case bi) holds.

If neither l_a nor l_b meet K in a line segment then if both l_a and l_b meet l_6 on the same side of $[p_{i6}, p_{j6}]$ and we have case bii). In fact in this particular case the 5-poly is unbounded so this can be excluded using remark 2. Now, assuming that p_{ai} and p_{bj} are vertices of P_6 there are two cases remaining. Using symmetry l_c crosses l_6 either in cii) $[p_{6j}, p_{6b}]$ or in ciii) $[p_{6b}, \infty]$
cii) Observe from figure 1.8 on page 35 that in this case l_b misses both quadrilaterals due to l_c, l_i, l_j, and l_6.
ciii) Observe from figure 1.9 on page 35 that here l_a misses the quadrilateral due to l_b, l_c, l_j, and l_6

For simplicity both of these cases have been drawn with l_a and l_b parallel. Note that remark 2 can be used to assume that $p_{ab} \in K$ however this fact has no actual bearing on the argument.

In every case some five lines are not in convex position. Thus if every five lines are in convex position there must be a 6-poly in the arrangement. □

Theorem 10. *Some n lines in general position in $\Re^2$ are in convex position if and only if every five of the lines are.*

Proof: The implication $\Rightarrow$ is obvious. The result is trivial when $n = 5$ and the result for $n = 6$ was proved in theorem 9, so assume here that $n \geq 7$.

Take n lines $l_1, \ldots, l_n$ that lie in general position in the plane with every five lines in convex position. Inductively assume that every $(n-1)$ of the lines are in convex position.

Using μ_2 map each of the lines l_i onto points p_i (and $\bar{p}_i$) on the sphere chosen so that $p_1, \ldots, p_{n-1}$ are the vertices of the admissible spherical $(n-1)$-gon P'_n corresponding to the $(n-1)$-poly P_n in the plane (note that this choice of normals is unique by corollary 3 since $n - 1 > 5$). Choose p_n so that it's the vertex of some admissible P'_j, $j \neq n$.

Remark 3: Thus p_n (as opposed to $\bar{p}_n$) will be a vertex of all admissible P'_i, $i \neq n$.

Proof of remark 3: Let i, $j \neq n$. In the plane P_i and P_j share $(n-2)$ lines including l_n so P_i and P_j each lie in an $(n-2)$-poly bounded by all of these lines. Such an $(n-2)$-poly is unique by corollary 3 as $(n-2) \geq 5$, and lies in the half-space due to l_n which corresponds to picking point p_n on the sphere. □

Either the points $p_1, \ldots, p_n$ are the vertices of an n-gon P' on the sphere or they are not.

(i) In the former case: If $k \in \text{int}P'_n$ then the n-gon is admissible and we are done. Otherwise the fact that P'_n is admissible gives us that $\{k, p_1, \ldots, p_{n-1}\}$ are in convex position, and we are assuming that $\{p_1, \ldots, p_n\}$ are in convex position. Now if $\{k, p_1, \ldots, p_n\}$ are in convex position P' is admissible and we are done. If not then $\{k, p_1, \ldots, p_n\}$ are not the vertices of an $(n+1)$-gon. Thus by Carathéodory's theorem either k lies inside the spherical hull of some three of the p_i's in which case $k \subseteq \text{int}(P')$ and P' is admissible. Or some p_i can be expressed as a combination of k and two other points: say that p_n lies inside the spherical hull of k, p_1 and

p_{n-1}. Consider P_i' where $i \neq 1, n-1, n$. By remark 3 P_i' must have p_j as a vertex for all $j \neq i$, yet then P_i' is not admissible as p_n cannot be a vertex of $\mathrm{sph}\{k, p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_n\}$. Thus P_i' does not correspond to an $(n-1)$-poly in the plane, a contradiction.

(ii) In the latter case: in other words when $\mathrm{sph}\{p_1, \ldots, p_n\}$ is not an n-gon, the fact that $\{p_1, \ldots, p_{n-1}\}$ are in convex position on the sphere implies that either p_n lies in the spherical hull of some three other points, or that p_i, $i \neq n$ lies in the spherical hull of p_n and some two other points. These cases are analogous if the roles of i and n are swapped. Thus without loss of generality $p_n \in \mathrm{sph}\{p_1, p_2, p_3\}$. Let $i \notin \{1, 2, 3, n\}$ and consider P_i', this polygon must have p_j as a vertex for all $j \neq i$ (by remark 3), yet p_n is not a vertex. □

Corollary 4. *The Carathéodory number for lines in the plane is six.*

1.4.4 Higher Dimensions

It is not clear whether such a Carathéodory number exists for higher dimensions. There may exist arrangements of n hyperplanes where the arrangement of every $n-1$ contains a rich cell yet the arrangement does not.

The result for $d = 2$ relies heavily on the fact that there is a unique cell bounded by n lines when $n \geq 5$. The example given in section 1.3 shows that this is not true in higher dimensions. If it could be shown for some $k(d)$ that in every suitable arrangement of $n \geq k$ hyperplanes two intersecting cells can be found each rich in a different $n-1$ of the hyperplanes, then the result follows for $k-1$.

As I observed earlier there is a strong connection with the equivalent projective problem, this is a question of McMullen:

> What is the largest number $\nu(d)$ such that any set of n points lying in general position in $\Re^d$ can be mapped by a permisable projective transformation onto the vertices of a convex polytope.

In fact the existence of such a projective transformation for n points in general position in $\Re^d$ is equivalent to the existence of an unbounded rich cell in an arrangement of $n-1$ hyperplanes in $\Re^d$ every d of which meet in a point and no $d+1$ of which have a point in common. This equivalence can be shown by using map μ_d to obtain $n-1$ points on S^d and letting the nth point correspond to the normal to the equator plane.

This question of McMullen was addressed by Larman [21] and later by LasVergnas [22], and bounds on $\nu(d)$ were obtained. In [21] it was shown that there is always such a transformation for $2d+1$ points. Thus every $2d$ hyperplanes in general position possess a (unbounded) rich cell. The fact that such an arrangement has a rich cell that is bounded is easy to see geometrically, as follows.

Lemma 11. *Every $2d$ hyperplanes in general position in $\Re^d$ have a (bounded) rich cell.*

Proof: Take some d of the hyperplanes, they meet in a unique point p, the other d meet at a unique point q. The arrangement of the hyperplanes meeting at q has 2^d cones all of which are rich. Point p must lie in the interior of one of these cones C_p say. Similarly q lies in $\text{int}(C_q)$ where C_q is a cone rich in the d planes meeting at p. Then $C_p \cap C_q$ is a rich cell of the arrangement. □

Larman showed that $\nu(3) = 7$ thus there is some arrangement of eight planes in $\Re^3$ that has no rich cells.

If there is a Carathéodory number it must be at least $\nu(d)$ or $\nu(d)+1$, however as noted above such a number may not exist. Conversely it is possible to attempt to improve the upper bound for $\nu(d)$ from its current value $\nu(d) < (d+1)(d+2)/2$, for $d \geq 2$ [22] to $\nu(d) \leq f(d)$ by finding an arrangement of $f(d)$ hyperplanes in general position that has no unbounded rich cell.

1.5 Helly's Theorem

It is interesting to consider other possible uses of the idea that is central to the previous work—that of contributing non-trivially to the boundary of a convex set. Helly's theorem [8] says that the intersection of n convex sets in $\Re^d$ is non-empty, provided every $(d+1)$ of the sets have this property, i.e. the intersection of every $(d+1)$ of the sets is non-empty; the number $(d+1)$ is known as the Helly number. There is a potential to use the above idea here, in order to create a different Helly-type theorem where convex sets contribute non-trivially to their intersection, actually forming part of its boundary. In this section I show that such a Helly-type number exists for halfspaces but that unfortunately it is not possible to obtain such a number for convex sets in general.

Lemma 12. *The Helly-type number for an intersection of some n halfspaces $H_1^+, \ldots, H_n^+$ in $\Re^d$, where each halfspace contributes (i.e. is not redundant) to the intersection, is $d+2$.*

Proof: One implication is trivial, so let every $d+2$ of the halfspaces bound a $(d+2)$-poly. Applying μ_d a set of n fixed points $p_1, \ldots, p_n$ on S^d are obtained, every $d+2$ of which are in (spherically) convex position, so by Carathéodory's theorem $p_1, \ldots, p_n$ are in (sph.) convex position. Observe that $p_1, \ldots, p_n$ is an admissable polygon: if k is in the interior of any $d+2$ of the points, we are done; otherwise k is in convex position with every $d+2$ of the points, thus every $d+2$ of $\{k, p_1, \ldots, , p_n\}$ are in convex position, and again the polygon of normals is admissable. Finally note that $d+1$ is certainly not enough as it is trivial that if the arrangement is in general position every $d+1$ normals will be in convex position. □

Such a Helly-type number does not exist for general convex sets because it does not exist for polyhedrons.

Example:
Assume the Helly-type number to be $n > 1$. Consider the intersection of any n distinct polyhedra $P_1, \ldots, P_n$ in $\Re^d$, where each P_i contributes to the boundary of the intersection. The intersection is itself a polyhedron P. Let $d = \min_v d(v, P)$ where $d(v, P)$ is the distance between P and a vertex v that is not on P. Define a polyhedron P' outside P whose facets are a distance $0 < d' < d$ from the facets of P. Observe that upon removing any P_i, P' will form a facet of the intersection of P' and $P_1, \ldots, P_{i-1}$,$P_{i+1}, \ldots, P_n$, as at least one facet has been removed. So every n of the polyhedra bound their intersection yet this is not true of $P_1, \ldots, P_n$,P'. This is a contradiction.

1.6 Summary

In this chapter I obtained several results on the theme of objects that form the boundary of a convex set. Firstly I showed that in an arrangement of hyperplanes the maximum number of rich cells, $f(d, n)$, is $\Theta(n^{d-2})$. Then I defined convex position for hyperplanes in $\Re^d$ and showed that the Carathéodory number for lines in the plane is five. The answer to this question in general dimensions is much more complicated, and such a number may not even exist. Finally I showed that a Helly number that does not involve any redundancy cannot be found for general convex sets, though $d + 2$ is the equivalent number for halfspaces.

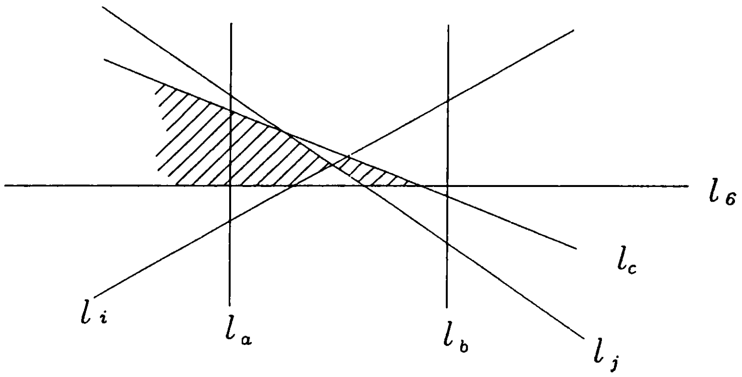


Figure 1.8: Theorem 9 case cii) five lines not in convex position.

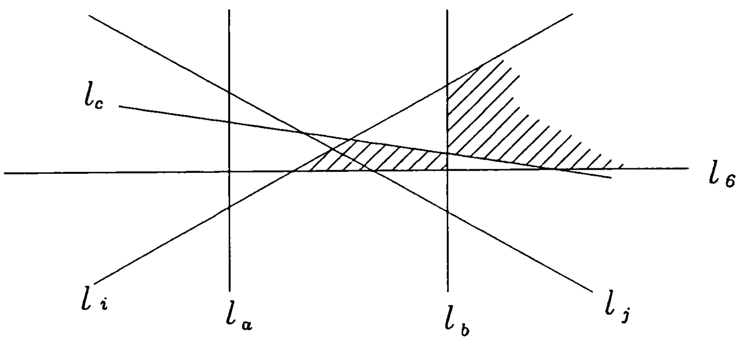


Figure 1.9: Theorem 9 case ciii) five lines not in convex position.

Chapter 2

The 180°Art Gallery Problem

2.1 Introduction

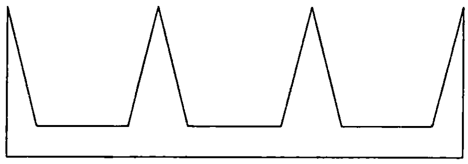
The original Art Gallery problem posed by Victor Klee in 1973 was:

> What is the minimum number $f(n)$ of guards needed to monitor any art gallery room of n sides if the guards are to be stationed at fixed points.

Here I will develop bounds on the number of guards that are required if their range of vision is restricted to 180°. Klee's question was answered by Chvátal who showed that $f(n) = \lfloor n/3 \rfloor$ [5]. The result is consequently known as 'Chvátal's Art Gallery Theorem'. Fisk gave an elegant proof of this theorem [13]. Fisk's proof is as follows.

It is known that the interior of a polygon on n vertices can be triangulated into $n-2$ triangles [25, p.12]. Consider the dual graph: the graph each of whose nodes corresponds to a triangle and where two nodes are joined by an edge if the corresponding triangles share a diagonal of the polygon. The dual graph has $(n-2)$ nodes each of degree at most three, and is a tree (it can have no cycles as a cycle would imply that the polygon had a vertex in its interior). Removing a leaf of this tree is equivalent to removing a triangle – a vertex and its two incident edges – from the edge of the polygon, this can always be done. Now consider the triangulation of the polygon. This is a planar graph on n vertices and is 3-colourable (we can remove one vertex and its two incident edges and inductively three colour the reduced graph, the removed vertex can be coloured as it is adjacent to only two other vertices). Any point in the polygon lies in some triangle, and each triangle must have all three of the colours on its vertices. Let the least used colour be 'red'. The whole polygon can be seen if a guard is stationed at each 'red' vertex as every triangle can be seen from its own red vertex. A simple counting argument shows that 'red' can be used no more than $\lfloor n/3 \rfloor$ times.

The bound $\lfloor n/3 \rfloor$ is sometimes necessary. This is illustrated by an example of Chvátal [5]:
A comb with k prongs has $3k$ edges, and requires k guards.



Many variants of the original problem have been considered. For example allowing the guards to patrol an edge or a line segment of the polygon. Many of these results are collected in [25].

At the 1992 Computational Geometry meeting in Barbados J. Urrutia asked the following variant of the problem.

> What is the minimum number $f_{180}(n)$ of guards needed to monitor any art gallery room of n sides if the guards are to be stationed at fixed points, and their range of vision is restricted to 180°?

Clearly the original art gallery problem will provide a lower bound of $f_{180}(n) \geq \lfloor n/3 \rfloor$, as at least as many 180°guards are required as 360°guards. The previous example shows the necessity of this lower bound; a comb with k prongs requires k 180°guards. The 360° problem also yields the trivial upper bound $f_{180}(n) \leq \lfloor 2n/3 \rfloor$. In section 2.4 below an optimal bound is obtained for monotone polygons. A monotone polygon with r reflex vertices requires $f_{180}(n) = \lfloor n/3 \rfloor$, and $f_{180}(n) \leq \lfloor r/2 \rfloor + 1$ 180°guards, results which interestingly match exactly the bounds attained when the range of vision is not restricted [1]. I will also show that in general $f_{180}(n) \leq \lfloor (4n+1)/9 \rfloor$; this result has since been extended and the extensions are described in section 2.3.1. Additionally in section 2.5.1 I obtain several results about the number of guards that are required for other restrictions on the angle of visibility.

2.1.1 Definitions:

A *polygon* P is defined to be a simple polygon, that is a closed finite connected region of the plane, bounded by the vertices $v_1, \ldots, v_n$ and the edges $[v_1, v_2], [v_2, v_3], \ldots, [v_{n-1}, v_n], [v_n, v_1]$ where no non-consecutive edges share a point. The boundary is a Jordan curve and divides the plane into two distinct regions, the interior *int* and the exterior of P.

A vertex of P shall be called *reflex* if it has an internal angle greater than 180°, and *convex* otherwise.

A point $x \in P$ *sees* $y \in P$, or y *is visible from* x if the line segment $[x, y] \subseteq P$. The line of sight is not blocked by grazing the boundary.

A *star polygon* is a polygon P where there exists some point $x \in P$ such that every point of P is visible from x, the set of all such points x is called the *kernel.*

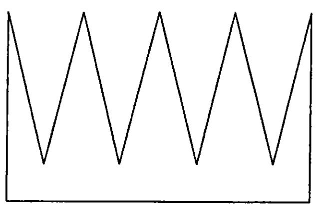
A *convex fan* centre v is a star polygon whose kernel includes a convex vertex v, [26, p.171] we extend this definition to also include star polygons P whose kernel includes a point $v \in (u, w)$ where u and w are adjacent vertices of P i.e. $[u, w] \subset \partial P$. We can reformulate the problem as:

> What is the smallest integer $f_{180}(n)$ such that any simple closed n-gon in the plane can be covered by $f_{180}(n)$ convex fans?

2.1.2 Observation:

Fisk's proof illustrates that for the general (360°) Art Gallery Problem it is appropriate to restrict the location of guards to vertices. This cannot be done here. Consider a 'saw' with k teeth, here $k = 5$:

This polygon has $n = 2k + 1$ vertices and requires $k = \lfloor n/2 \rfloor$ 180° vertex-guards, but only $\lceil k/2 \rceil = \lceil (n-1)/4 \rceil$ 180° guards placed opposite every other reflex vertex.

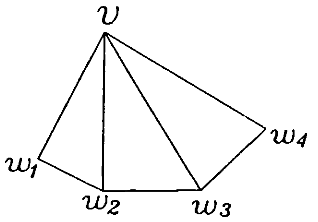


In Section 2.2 I obtain bounds on $f_{180}(n)$ for small values of n: $n = 5, 8$. I also give the number of guards required for particular types of polygon. In Sections 2.3 and 2.4 these results are used to prove the main theorems.

2.2 Preliminary Results

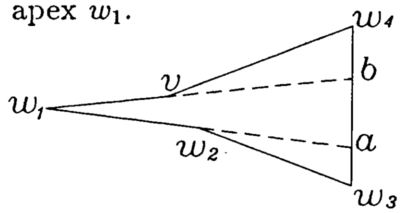
Lemma 13. *A convex fan, a convex n-gon, an n-gon with only one reflex vertex and a 5-gon each require exactly one 180° guard.*

Proof: Clearly a polygon requires at least one guard.
(i) The lemma is trivially true for a convex fan by its definition, and for a convex n-gon.
(ii) Let P be an n-gon with a single reflex vertex. Extend a ray from this vertex bisecting the internal angle, until it first meets ∂P at p. The ray divides P into two convex polygons each of which can be seen from p. The internal angle at p is at most 180°, so P requires one 180° guard which can be placed at p.
(iii) Now consider a 5-gon Q. Triangulate Q and 3-colour the resulting graph on five vertices. One of the colours is used once only at a vertex v which lies on all three triangles.
Label the vertices as shown:



If v is convex we have a convex fan which requires one guard. Otherwise v is reflex, if no other vertex w_i is reflex then by (ii) above one guard will suffice, and we are done.

Note that a 5-gon can have at most two reflex vertices[1]. As w_1 and w_4 are necessarily convex, wlog let w_2 be the second reflex vertex. Consider the cone generated by $[v, w_2]$ with apex w_1.



Let the cone meet $[w_3, w_4]$ in a line segment $[a, b]$ as illustrated. If a guard

[1]Any polygon P must have at least three convex vertices as follows: If P has n vertices it can be cut into $(n-2)$ triangles, each triangle contributes π to the internal angle so the total internal angle of P is $(n-2)\pi$. Each of the reflex vertices has by definition an internal angle strictly greater than π, so to avoid contradiction there can be at most $(n-3)$ reflex angles.

is placed at $g \in [a, b]$ then w_1, w_2 and v are visible from g. The internal diagonals $[g, v]$, $[g, w_1]$, $[g, w_2]$ form a triangulation of Q, g lies on the boundary of every triangle and the angle at g is 180°, hence Q is a convex fan centre g. Q requires one guard. □

Corollary 5. *A 4-gon requires one 180° guard.*

Corollary 6. $f_{180}(n) \leq \lfloor n/2 \rfloor$

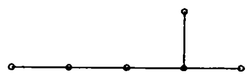
Proof: This is true $n = 3, 4$. Let $n \geq 5$, recall that the dual graph of the triangulation is a tree with maximal vertex degree three. We can always remove 2 or 3 vertices from the edge of this tree: simply start at any node and using a Depth First Search find w the furthest vertex, if the parent v of w has degree two then we can remove v and w from the tree, this equates to removing a 4-gon from the edge of the polygon; if the parent of w has degree three then we can remove three vertices from the tree, which equates to removing a 5-gon from the edge of the polygon. The result follows by induction. The removed 4- or 5-gon requires one guard. An $(n-2)$- or $(n-3)$-gon remains. □

Lemma 14. *An 8-gon P requires at most two guards.*

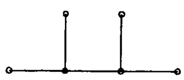
Proof: Triangulate P and consider the dual graph of the triangulation. Recall that this is the graph whose nodes represent the triangles, and where two nodes are joined by an arc if the corresponding triangles are adjacent (share a diagonal of P). As explained in the introduction this graph will be a tree on six nodes with maximal degree 3. There are only four distinct such trees.

a)

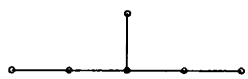
b)



c)

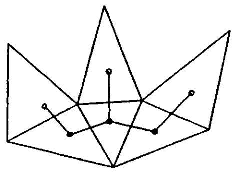
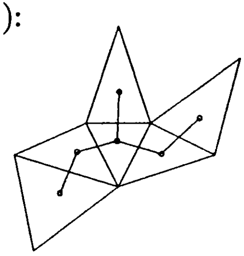
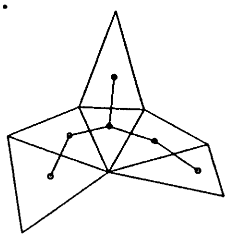


d)



In cases a), b) and c) it is possible to remove an arc so as to leave two connected trees each with three nodes. This corresponds to dividing P into two 5-gons by cutting along an internal diagonal. By lemma 13 each 5-gon can be covered by one 180° guard so P requires at most two guards in total.

Only in case d) is this not possible. Here there are three possibilities for the arrangement of the triangles.

(i):  (ii):  (iii): 

Case (i)

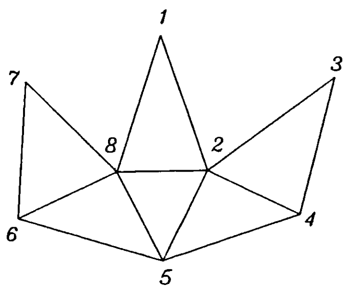
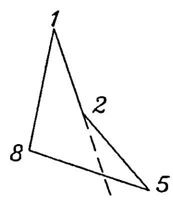


Figure 2.1: Lemma 12 case (i).

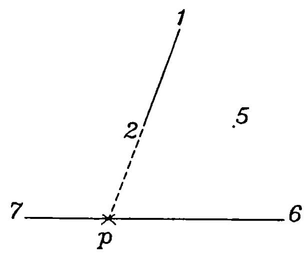
Label the vertices as shown. Vertices 1, 3 and 7 are necessarily convex. If either of 2 or 8 are convex the 8-gon requires at most two guards: If 2 is convex {2,3,4,5,8,1} is a convex fan centre 2 and requires one 180°guard, there remains the 4-gon {5,6,7,8} which can be covered by one guard. The case for 8 is symmetrical. So we assume that both 2 and 8 are reflex vertices.

If 1 can 'see' 5 i.e. if (1,5)$\subset$ P then [1,5] cuts P into two 5-gons {1,2,3,4,5} and {5,6,7,8,1} each of which requires one 180°guard, and we are done. Otherwise either angle $1\hat{2}5$ or angle $1\hat{8}5$ is reflex and blocks the line of sight. Due to symmetry we can assume that $1\hat{2}5$ is reflex. Extend edge $\overrightarrow{12}$ until it first meets ∂P at p. The ray must pass through (5,8), so $p \in (5,6]$ or $p \in (6,7)$. (The ray cannot hit ∂P at [7,8] as 8 is reflex.)



If $p \in (5,6]$ the ray divides P into two 5-gons each requiring one guard, and we are done. So assume $p \in (6,7)$.

a) If 6 is convex then $Q = \{1,2,5,6,7,8\}$ is a convex fan centre p.
$(p,8) \subset P$ as $p \in \Delta\{6,7,8\}$.
$(p,1) \subset P$ by construction.
Consider $R = \{2,p,6,5\}$. If 5 is convex then R is a convex quadrilateral, so $(p,5) \subset R \subseteq P$. Otherwise 5 is reflex: then $(2,6) \not\subset R$ so $(p,5) \subset R$ –it must be the diagonal in the triangulation of R. So $(p,5) \subset P$.



Internal diagonals [p,8], [p,1] and [p,5] divide Q into four triangles each of which have p on their boundary, the angle at p is 180°. Q and the remaining quadrilateral $\{2,3,4,5\}$ each need one guard. P requires at most two guards.

b) If 6 is reflex: As the vertex 2 can 'see' 5 and p, we can consider the triangulation of quadrilateral {2,5,6,p}. As 6 is reflex the triangulation cannot contain [5,p], hence $[2,6] \subset P$, and divides P into two 5-gons, thus P requires at most two guards. Case (i) is proved.

Case(ii)

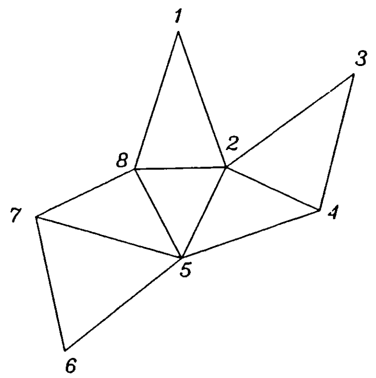
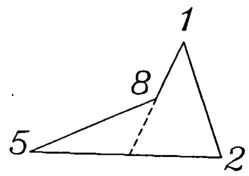


Figure 2.2: Lemma 12 case (ii).

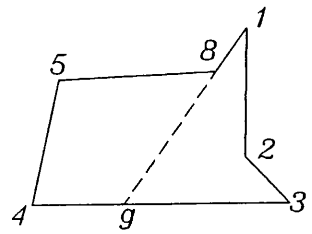
As for case (i) if the angle at 2 is convex or if (5,1)$\subset$ P at most two guards are required. So assume that 2 is reflex, there are two possibilities either angle $1\hat{2}5$ or angle $1\hat{8}5$ is reflex.

a) Angle $1\hat{8}5$ is reflex:
Extend $\overrightarrow{18}$ until it first hits ∂P at g. The ray passes through [2,5); $g \notin [2,3]$ as the angle at 2 is reflex, so $g \in (3,4)$ or [4,5].



If $g \in [4,5]$ then $[1,g]$ cuts P into two 5-gons and we are done.

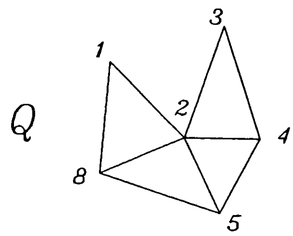
If $g \in (3,4)$ then provided $[g,5] \subset P$, {1,2,3,4,5,8} will be a convex fan centre g:
$[g,2) \subset P$ as $g \in \Delta\{2,3,4\}$
$[g,1] \subset P$ by construction.



This convex fan needs one guard, the remainder of P, {5,6,7,8}, can be covered by one further guard.

On the other hand if $[g,5] \not\subset P$ then this implies that $[4,8] \subset P$ as a triangulation of $\{g,4,5,8\}$ must exist. Then [4,8] cuts P into two 5-gons, hence in this case at most two guards are required.

b) Angle $1\hat{2}5$ is reflex, and $1\hat{8}5$ is convex.
Consider the 6-gon $Q = \{1,2,3,4,5,8\}$.

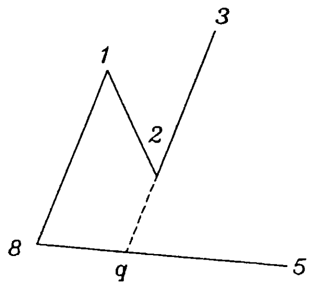


If $(4,8) \subset Q \subseteq P$ then [4,8] divides P into two 5-gons, and P requires at most two guards. If not then either bi) angle $4\hat{5}8$ is reflex, or bii) angle $4\hat{2}8$ is reflex.

bi) Extend $\overrightarrow{45}$ until it first hits ∂Q at p. The ray passes through (2,8) and so $p \in [1,8]$. Thus $[4,p]$ cuts P into two 5-gons and we are done. (Note that $p \notin [1,2]$ as then $1\hat{2}5$ would be the angle at a vertex of the triangle $\{p,2,5\}$ which contradicts $1\hat{2}5$ being reflex.)

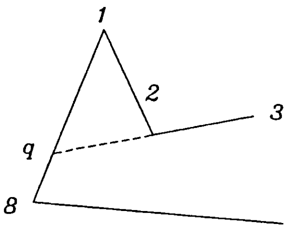
bii) Extend $\overrightarrow{32}$ until it first hits ∂Q at q. Then Q is a convex fan centre q: To prove this we observe that either $q \in [5,8]$ or $q \in [8,1]$, ($q \notin [4,5]$ as then $1\hat{2}5 + 3\hat{2}q > 360°$, a contradiction).

If $q \in [5,8]$



$(q,1) \subset Q$ as $\{1,2,q,8\}$ is a convex quadrilateral,
$(q,3) \subset Q$ by construction.
$(q,4) \subset Q$ as in $\{q,3,4,5\}$ the angles at 3, q, and 5 are convex (recall it is assumed here that $4\hat{5}8$ is convex).
Hence $[q,1]$, $[q,3]$, and $[q,4]$ divide Q into four triangles.

If $q \in [8,1]$



$(q,3) \subset Q$ by construction,
$(q,5) \subset Q$ as $\{2,5,8,q\}$ is a convex quadrilateral,
$(q,4) \subset Q$ as in $\{q,2,4,5\}$ the angles at q, 2 and 5 are convex,
$[q,3]$, $[q,4]$, and $[q,5]$ divide Q into four triangles.

Q requires one 180°guard at q, one further 180°guard can cover {5,6,7,8}. Case (ii) is proved.

Case (iii)

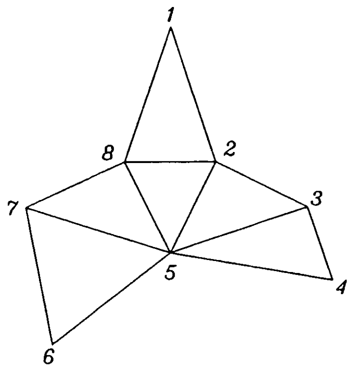


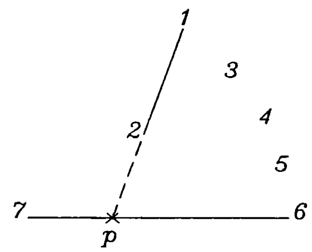
Figure 2.3: Lemma 12 case (iii).

If 5 is convex then {5,6,7,8,2,3,4} is a convex fan centre 5; at most two guards will be required for P. Assume then that 5 is reflex. As in previous cases if $(5,1) \subset P$ at most two guards are required, so (using symmetry) we can without loss of generality restrict attention to the case when angle $1\hat{2}5$ is reflex. Extend $\overrightarrow{12}$, the ray passes through (5,8) and first meets ∂P at $p \in [5,6]$, [6,7], or [7,8].

a) If $p \in [5,6]$ then [1,p] divides P into two 5-gons and we are done.

b) If $p \in [6,7]$ consider $Q = \{2,3,4,5,6,p\}$

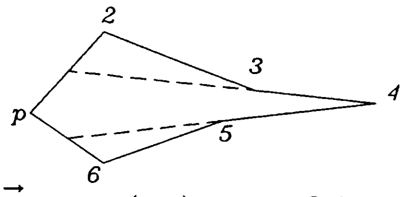
We have assumed that 5 is reflex. The angles of Q at 2 and p must be convex, as must those at 4 and 6.



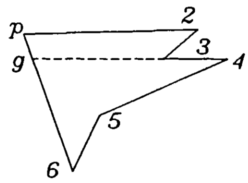
If 3 is convex Q is a 6-gon with one reflex vertex, and by Lemma 13 requires one guard.

If 3 is reflex consider the cone C generated by [3,5] with apex 4.

bi) If $p \in C$ then (p,3), (p,4), and (p,5)$\subset Q$.
Q is a convex fan centre p.



bii) If $p \notin C$ then wlog p lies above $\overrightarrow{43}$. Let $\overrightarrow{43}$ meet (p,6) at g. Q is a convex fan centre g.
$(g,2) \subset Q$ as $\{g,3,2,p\}$ is convex
$(g,4) \subset Q$ and $(g,5) \subset Q$ as $g \in C$.

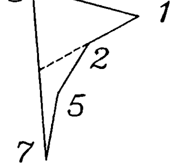


Thus Q requires one guard. The quadrilateral $\{1,p,7,8\}$ remains and this can be covered by one further guard.

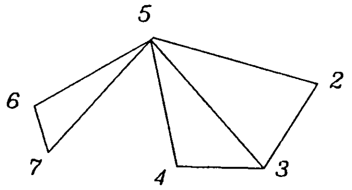
c) The final case to consider is if $p \in [7,8]$ (thus 8 is a convex angle).

Consider {1,2,5,7,8} if 2 can 'see' 7 then we can retriangulate replacing [8,5] with [2,7] to get an instance of case (ii), otherwise angle $2\hat{5}7$ is reflex (it can't be $2\hat{8}7$ as $2\hat{8}7$ is convex).

{8,1,2,5,7} is a convex fan centre 8 and requires one guard.



It is possible to place a guard at 5, so that the guard can 'see' {5,6,7} and {5,2,3,4} within 180°. These two areas together form a (degenerate) convex fan centre 5.



□

Corollary 7. *A polygon with 6 or 7 vertices can be seen by at most two 180° guards.*

2.3 The Result For General Polygons

We can now establish an upper bound for $f_{180}(n)$.

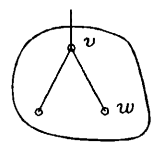
Theorem 15.

$$f_{180}(n) \leq \lfloor \frac{4n+1}{9} \rfloor$$

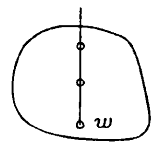
A polygon P with $n \geq 3$ vertices requires at most $\lfloor (4n+1)/9 \rfloor$ 180° guards.

Proof: The result is valid $3 \leq n \leq 8$, so let $n \geq 9$ here. Triangulate P and construct the dual graph of the triangulation. This graph is a tree T on $(n-2)$ nodes with maximal degree three. For trees T with $|T| \geq 3$ the following shows that it is always possible to prune a clump of either 3 nodes, 5 nodes, 6 nodes, $(6+1)$ nodes, or $(4+5)$ nodes from the tree leaving behind a connected tree on fewer nodes:

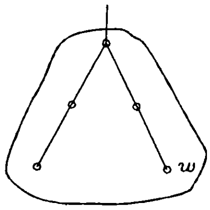
Root T at any node and perform a depth first search to locate the node w which is furthest from the root. Let the parent of w be v. If $\deg(v) = 3$ then we may remove an 'A clump' of three nodes.



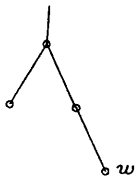
Otherwise $\deg(v) = 2$. Let u be the parent of v, then if $\deg(u) = 2$ or 1 we can remove a 'B clump' of three nodes.



Otherwise $\deg(u) = 3$. There are two possible shapes for the clump of nodes beneath u. Either we have a 'C clump' in which case we can remove

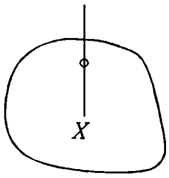


A 'C clump'

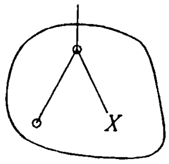


An 'X clump'

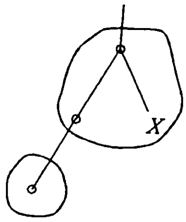
5 nodes, or we have an 'X clump'. Consider those trees with an X clump, the possibilities are limited by w being the furthest node. In all but four cases we can cut an A, B, or C clump from the tree, the four cases are illustrated below.



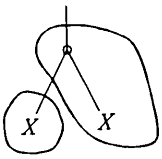
Remove 5 nodes



Remove 6 nodes



Remove $(6 + 1)$ nodes



Remove $(5 + 4)$ nodes

This operation of pruning r nodes from T is that of cutting P along one of the internal diagonals of the triangulation to remove an $(r + 2)$-gon, an $(n - r)$-gon remains.

Let the operation be applied repeatedly until there are at most two nodes remaining. Say 3 nodes have been pruned α times; 5 nodes, β times; 6 nodes, γ times; $(6 + 1)$ nodes, δ times; and $(4 + 5)$ nodes, η times. Then we have divided P into α 5-gons each requiring one guard (by lemma 13), β 7-gons and γ 8-gons each requiring two guards, δ 8-gons and triangles requiring a total of three guards, and η 6-gons and 7-gons each pair requiring a total of four guards. At most a quadrilateral remains, this would require one guard.

$$|T| = n - 2 \text{ so } 3\alpha + 5\beta + 6\gamma + 7\delta + 9\eta \leq n - 2$$

$$\begin{aligned}
\text{number of guards} &= \alpha + 2(\beta + \gamma) + 3\delta + 4\eta + 1 \\
&\leq \frac{n - 2 + \beta + 2\delta + 3\eta}{3} + 1 \\
&\leq^2 \frac{4n - 8}{9} + 1 \\
&= \frac{4n + 1}{9} \\
\text{hence } f(n) &\leq \lfloor \frac{4n + 1}{9} \rfloor
\end{aligned}$$

□

Thus $\lfloor n/3 \rfloor \leq f_{180}(n) \leq \lfloor (4n + 1)/9 \rfloor$.

2.3.1 Extensions Of This Result

The result described in this section has been subsequently extended, firstly by G. Csizmadia [6], and more recently by Csizmadia and G. Tòth [7].

Let a U-configuration be the configuration of four triangles shown in figure 2.4. Csizmadia proved that one guard can monitor a U-configuration

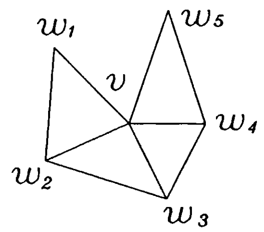


Figure 2.4: A U-configuration

except for perhaps a triangle that has $[w_2, w_3]$ as one of its sides. He uses this result to prove that $f_{180}(11) = 3$ by case analysis. He then obtains a bound of $f_{180}(n) \leq \lfloor (5n + 1)/12 \rfloor$ by inductively removing connected portions of the dual tree.

More recently Csizmadia and G. Tòth claim to have proved an upper bound of $\lceil \frac{2}{5}(n-2) \rceil = \lfloor 2n/5 \rfloor$. I have seen an (incomplete) sketch of the proof [7]. The first step is to classify all polygons of six vertices (for example figure 2.5 on page 54) where it is not possible for one guard to cover all but one triangle that is adjacent to the rest of the tree. Using this, and investigating the different tree endings (of up to fifteen nodes) that might cause problems. They claim that for any polygon P a part of P can be removed so that either a) the removed part can be seen by one guard and the rest of the polygon has three less sides, or b) the removed part requires two guards and the rest of P has five less sides. This yields the bound $\lfloor 2n/5 \rfloor$.

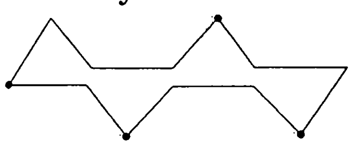
2.4 The Result For Monotone Polygons

A chain of vertices $p_1, \ldots, p_k$ is called *monotone* with respect to a line L if the projections of $p_1, \ldots, p_k$ onto L retain the ordering of the chain.

A polygon P is called *monotone* if its vertices can be partitioned into two chains monotone with respect to the same line L. Such a polygon P is also called *convex in one direction* as any line perpendicular to L intersects P in a line segment.

Theorem 16. $\lfloor n/3 \rfloor$ *180°guards are sometimes necessary and always sufficient to cover a monotone polygon P on n vertices, i.e. $f_{180}(n) = \lfloor n/3 \rfloor$ for a monotone polygon.*

Proof:
Necessity



A polygon consisting of k 'triangles' joined as shown has $3k$ edges and requires k guards.

Sufficiency

By Lemmas 13 and 14 $\lfloor n/3 \rfloor$ guards are sufficient when $n \leq 8$. Let $n \geq 9$ we inductively cut P into a 5- or 6-gon requiring one guard, and an $(n-3)$-gon. Thus P requires $1 + \lfloor (n-3)/3 \rfloor = \lfloor n/3 \rfloor$ guards.

Let P be monotone with respect to the x-axis, P is formed by a 'top' and a 'bottom' chain of vertices. Order the vertices of P $1, \ldots, n$ in the positive x-direction The order is unimportant when more than one vertex has the same x-coordinate, and the method used is still valid in this case. Without loss of generality 5 is a 'bottom' vertex.

(i) If 5 can 'see' the last top vertex v where $1 \leq v \leq 4$ then cut P at [5,v]. This divides P into the polygon {1,2,3,4,5} which requires one guard, and P' which has $(n-3)$ vertices, and requires $(\lfloor n/3 \rfloor - 1)$ guards by induction.

(ii) If not then a reflex vertex i, $v < i < 5$ must block the line of sight from 5 to v, (hence i must be a bottom vertex or monotonicity would be violated).

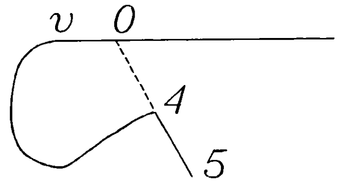
There are four possibilities:

a) 'top' chain $\{1,2,3,\ldots,n\}$ 'bottom' chain $\{1,4,5,\ldots,n\}$ $i = 4$,
b) 'top' chain $\{1,3,\ldots,n\}$ 'bottom' chain $\{1,2,4,5,\ldots,n\}$ $i = 4$,
c) 'top' chain $\{1,2,\ldots,n\}$ 'bottom' chain $\{1,3,4,5,\ldots,n\}$ $i = 3$ or 4,
d) 'top' chain $\{1,\ldots,n\}$ 'bottom' chain $\{1,2,3,4,5,\ldots,n\}$ $i = 2$ or 3 or 4.

Note that a vertical line meets P in a line segment, and a vertical line which passes through a reflex vertex resolves it into two convex vertices.

Cases (a), (b) and (c) when $i = 4$.

Extend the edge $\vec{54}$. Let it hit the top side of P at 0, 0 lies between v and the next top vertex.[3]



Cutting P at [0, 5] divides P into the 5-gon {0,1,2,3,4} and the $(n-3)$-gon $\{0,5,6,7,\ldots n\}$.

Inductively at most $\lfloor n/3 \rfloor$ guards are required.

(c) When $i = 3$:

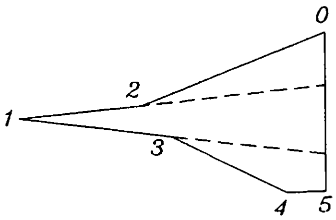
Here 4 must be a convex vertex (else 3 lies to the left of $\vec{54}$ which would imply $i = 4$ the case dealt with above). Vertex 1 is also convex. Let 0 be

[3]Vertex v lies to the left of $\vec{54}$ as 4 prevents 5 seeing v; $\vec{54}$ is in the negative x-direction and so hits the top edge to the left of 4, i.e. to the left of the next top vertex.

the point where the vertical line through 5 meets the top side of P. Cut P along [5,0]. The angles created at 5 and 0 are convex, or else monotonicity would be contradicted.

If 2 is convex then {0,2,1,3,4,5} is a 6-gon with one reflex vertex (i.e. the vertex at 3), which requires one guard by Lemma 13. The $(n-3)$-gon {0,5,6, ... ,n} remains and we can induct.

If 2 is reflex then consider the cone C generated by [2,3] with apex 1: 0,4 and 5 lie outside the cone because 2 and 3 are reflex, and if $5 \in C$ then 5 can 'see' 2 as then $(5,2) \subseteq C$ i.e. 5 can see v which would be case (i) above.

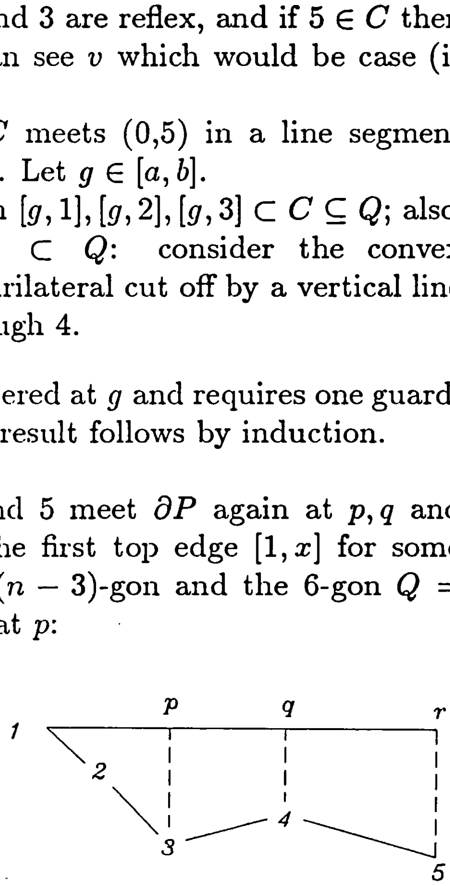


$=Q$

So C meets (0,5) in a line segment $[a, b]$. Let $g \in [a, b]$.
Then $[g, 1], [g, 2], [g, 3] \subset C \subseteq Q$; also $[g, 4] \subset Q$: consider the convex quadrilateral cut off by a vertical line through 4.

Hence {0,2,1,3,4,5} is a convex fan centered at g and requires one guard, and $\{0,5,6, \dots, n\}$ is an $(n-3)$-gon. The result follows by induction.

(d) Let the vertical lines through 3,4 and 5 meet ∂P again at p, q and r respectively. Then p, q and r lie on the first top edge $[1, x]$ for some $x \in \{6, \dots, n\}$. Cut P at $[5,r]$ into an $(n-3)$-gon and the 6-gon $Q = \{r, 1, 2, 3, 4, 5\}$. Consider placing a guard at p:
$[p, 3] \subset Q$ (monotonicity),
$[p, 2] \subset Q$: a triangulation of $\{1,2,3,p\}$ exists; Hence either $[p, 2]$ or $[1, 3] \subset Q$. If $[1,3] \subset Q$ then 2 is convex, so $\{1,2,3,p\}$ is convex which implies that $[p, 2] \subset Q$,
$[p, 4] \subset Q$ as $\{p, q, 4, 3\}$ is convex.



If $[p, 5] \subset Q$ then Q is a convex fan centre p and requires one guard. Otherwise if $[p, 5] \not\subset Q$ then consider $\{p, 4, 5, r\}$. The angle $p\hat{4}5$ must be reflex, if we extend edge $\overrightarrow{54}$ it cuts (p, r) at a point g. In this case $[g, 5]$ cuts P into a 5-gon and an $(n-3)$-gon and we can induct. □

Theorem 17. *The number of 180° guards required for a monotone polygon P with r reflex vertices is at most* $\lfloor r/2 \rfloor + 1$.

This result is better than Theorem 16 whenever $r < 2\lfloor n/3 \rfloor - 2$.

Proof: Let P be monotone with respect to the x-axis. We draw a vertical line through each of the reflex vertices—this resolves each reflex vertex into two convex angles.

Assume for the moment that no two reflex vertices have the same x-coordinate. The lines cut P into $(r+1)$ convex pieces, which can be seen by $(\lfloor r/2 \rfloor + 1)$ 180° guards—place a guard opposite every other reflex vertex (in a horizontal sort). Each guard will be placed on an edge or at a convex vertex, and so will have a field of view less than 180°.

A maximum of two reflex vertices may share the same x-coordinate. Each time that two reflex vertices r_1 and r_2 share the same x-coordinate the above method would place a guard opposite one of them to cover the convex region K of P to the left or right of $[r_1, r_2]$ and the now degenerate convex region of P between the vertical lines through r_1 and r_2. Placing a guard anywhere on ∂K will cover the required area within 180°. □

Necessity

The comb of page 37 and the necessity example on page 49 are both monotone and each has $r = 2(n-3)/3$ reflex vertices and needs $\lfloor r/2 \rfloor + 1$ which is $\lfloor n/3 \rfloor$ 180°guards.

2.5 Conclusion.

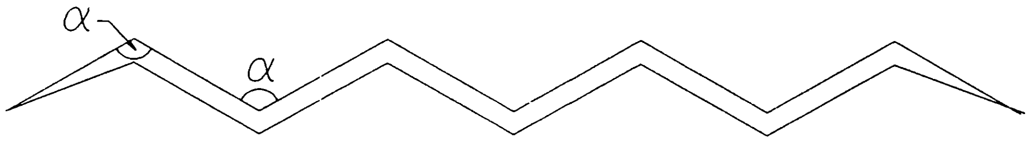
In this chapter I attained the non-trivial bound $f_{180}(n) \leq \lfloor (4n+1)/9 \rfloor$ for the 180° art gallery problem, a result which has since been extended. I also showed that for monotone polygons restricting the field of view from 360° to 180° does not affect the bounds obtained [1]. I strongly suspect that the true bound for this problem is $f_{180}(n) = \lfloor n/3 \rfloor$ in general. The current technique involves cutting off ever larger portions of the dual tree and limiting the numbers of guards that are required. This has lead to successive improvements but is unlikely to achieve a bound of $\lfloor n/3 \rfloor$. The example on 54 illustrates the difficulty—to achieve $\lfloor n/3 \rfloor$ using this method we must always be able to remove a portion seen by k guards so that at most $(n-3k)$ connected triangles (not necessarily of the original triangulation) remain; this is an example where this is not possible for $k = 1$. The problem with this technique seems to be its dependence on the triangulation, yet it should be noted that there do exist polygons which have a unique triangulation.

2.5.1 Other Related Results

It is feasible to consider what happens when other restrictions are placed on the angle of visibility. Let $f_\theta(n)$ designate the number of guards that are required to monitor an art gallery of n sides if each guard is allowed to survey a maximum angle of $\theta°$.

Claim 18. *If* $\theta < 180°$ *then* $f_\theta(n) \geq \lfloor n/2 \rfloor$.

Proof: The proof is by example. A 'zig-zag' with $2k$ vertices requires at least k θ-guards if the angle α of the zig-zag is strictly greater than θ.



□

Claim 19. $f_{60}(n) \leq (n-2)$

Proof: Each triangle of the triangulation has an angle that is at most 60°. □

Claim 20. *If* $\theta < 60°$ *then* $(n-2) \geq f_\theta(n)$.

Proof: If n is even, a zig-zag on n vertices with angle $180° > \alpha > 3\theta$ requires $(n-2)$ θ-guards. To see this note that each 'V' shaped region requires at least four guards, and each end triangle requires one guard.

If n is odd, then consider the same shape, but with one end flattened so that it forms a parallelogram. This parallelogram requires at least two guards, as all of its angles are at least 60°. □

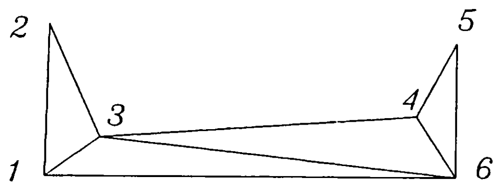


Figure 2.5: A polygon on six vertices. If this is joined to the tree via [1, 6] (or [3, 4]) it is not possible for a guard to see all but a single triangle that remains attached to the rest of the tree.

Chapter 3

The Diameter Problem

3.1 Introduction

In this chapter I give details of a (deterministic) algorithm that solves the diameter problem. This problem is one of the classical problems of computational geometry. The problem can be formulated as follows:

> Given a set A of n points in $\Re^3$ find x, $y \in A$ such that $\mathrm{d}(x, y) = \max\{\mathrm{d}(u, v) : u, v \in A\}$ in other words what are the farthest pair of points from a set of n points.

This question can be trivially answered in $O(n^2)$ time, by simply calculating the distances between all possible pairs of points and comparing the results.

The diameter question can be asked in any dimension. A lower bound for the problem is $\Omega(n \log n)$ [26], which is obtained by transforming the problem to set disjointedness. As a guide to the relative complexity several results describe how many diametral pairs (pairs of points that realise the diameter) there can be among n points in $\Re^d$. In [11] it is shown that there are at most n diametral pairs in $\Re^2$. In $\Re^3$ there can also be at most $O(n)$ such pairs (in fact $2n - 2$), this result is known as Vászonyi's conjecture and is demonstrated in [18,16]. However in $\Re^d$, $d \geq 4$ there can be as many as $O(n^2)$ diametral pairs [12,24]. The example that shows this is due to H. Lenz. For the construction in $\Re^4$ take two mutually orthogonal circles radius $1/\sqrt{2}$ around the origin, and distribute $n/2$ points within a small arc of each; the diameter of the set is one and $n^2/4$ pairs realise this.

The diameter problem has been solved in $\Re^2$; for instance by noting that the diameter must be realised by an antipodal pair of vertices of the convex hull, if the convex hull is constructed in $O(n \log n)$ time, then rotating a pair of parallel planes around its boundary to identify all antipodal pairs takes $O(n)$ time and yields the diameter, [26].[1] Notice that, in $\Re^d$ with $d \geq 4$, Lenz's example can be slightly perturbed so that $O(n^2)$ pairs nearly realise the diameter, thus $\Re^3$ is the most interesting case left open. It is hopeful that if pairs of points are chosen wisely, the number of pairs that need to be considered is relatively small.

The algorithm detailed below is compiled from the work of A.C. Yao [29], B. Chazelle [4], and D. Kirkpatrick [19]. Previously only a sketch of the algorithm existed, and it has never, to the best of my knowledge, been presented in a unified form. This algorithm has a running time of $O(n^{3/2} \log n)$. The best preceeding algorithm $\Re^3$ was due to A.C. Yao [29], he gives an algorithm that returns the diameter of n points in $\Re^3$ in a time of $O((n \log n)^{1.8})$.

3.2 General Outline And Yao's Algorithm

An algorithm described by Yao in [29] will be the top level algorithm; this applies a divide and conquer strategy using Chazelle's answer to another problem 'the post office problem' [4] which produces the farthest point, in some set B, from a query point x. The post office problem is usually stated for nearest point queries.

> The *post office problem for farthest point queries* is to: preprocess a set B of points so that given a test point x the furthest $y \in B$ from x can be quickly obtained.

If a large number of queries are to be made on a set B then the preprocessing time may be expensive, because if this yields a fast query time the net result may be quicker.

The post office problem was first formulated in two dimensions by Knuth [20]. In three dimensions there have been several results. Dobkin and Lipton [9] used a technique generalised from two dimensions that answered

[1] This type of technique will not work in $\Re^3$ because in this case there can be as many as $O(n^2)$ antipodal pairs.

a query in $O(\log n)$ time, given an extremely high preprocessing time of $O(n^{12} \log n)$. Yao [29] improved the $\Re^3$ result, obtaining a query time of $O((\log n)^2)$ with a substantially better preprocessing time of $O(n^5 \log n)$. This was the result that led to the $O((n \log n)^{1.8})$ diameter algorithm. Subsequently Chazelle produced an algorithm that had a query time of $O((\log n)^2)$ and a preprocessing time of $O(n^2)$. This is the algorithm that will be described below.

3.2.1 Yao's Algorithm For The Diameter Problem

(1) Divide A into $r = \lceil n/q \rceil$ sets $B_1, \ldots, B_r$ each with at most q points.
(2) Preprocess each B_i for farthest point queries.
(3) For each $x \in A$ and $1 \leq i \leq r$ find the point $y_{xi} \in B_i$ which is the furthest point from x in B_i.
(4) For each $x \in A$ compare the y_{xi}'s $1 \leq i \leq r$ to find the point z_x in A furthest from x.
(5) Find the longest such edge $\{x, z_x\}$.

This algorithm clearly returns the diameter of the set. If $P(n)$ and $Q(n)$ are the preprocessing and query times respectively for the post office problem on n points in $\Re^3$ then the total running time is:

$$rP(q) + nrQ(q) + nO(r-1) + O(n-1)$$

where the additional terms are the times for the various comparisons. Chazelle's algorithm gives $P(n) = O(n^2)$ and $Q(n) = O((\log n)^2)$. In section 3.5 I show that taking these time bounds, if we pick $q = n^{1/2} \log n$ the algorithm requires a total running time of $O(n^{3/2} \log n)$.

3.2.2 Outline Of Chazelle's Algorithm

Chazelle writes his algorithm to answer a nearest point query, but all the arguments adapt straightforwardly for farthest point queries. The algorithm hinges on a geometric observation about Voronoi diagrams.

The *farthest point Voronoi diagram* $\mathcal{V}(B)$ (hereafter abbreviated to Voronoi diagram) of a set $B = \{p_1, \ldots, p_n\}$ in $\Re^d$ is a division of the space into regions called *cells*. A cell is that part of space which is farther from a particular p_i than from any other points of B. The Voronoi cell of point

p_i is the set $\{p \in \Re^d : \mathrm{d}(p, p_i) > \mathrm{d}(p, s), s \in B - \{p_i\}\}$. The union of all such cells, each associated with one of the n points, and the faces between the cells make up the Voronoi diagram $\mathcal{V}(B)$ of B. Note that all faces and regions of $\mathcal{V}(B)$ are convex.

We could equivalently define $\mathcal{V}(B)$ to be the cell complex whose faces are equivalence classes of points under the relation $C(p) = \{s \in B : \mathrm{d}(p, s)$ is maximised$\}$ and where p and q are equivalent points if $C(p) = C(q)$ see [10, p.294].

The action of finding the farthest point p_i in B from a query point q is equivalent to that of locating q in $\mathcal{V}(B)$: q will lie in the Voronoi cell of p_i. Chazelle shows that if the points of B are ordered $p_1, \ldots, p_n$ with respect to the x direction; and the cells of $\mathcal{V}(B)$ are divided into two groups: those associated with $p_1, \ldots, p_{(n/2)}$; and those with $p_{(n/2)+1}, \ldots, p_n$; then the faces which lie between these two groups of cells are unique with respect to x. In other words a line parallel to the x-axis will touch exactly one of these faces.

Effectively these faces form a 'curtain' (not necessarily flat) of 2, 1, and 0 dimensional faces, which lie between the two groups of cells. On one side of this curtain lie points which are furthest from one of $p_1, \ldots, p_{(n/2)}$ and on the other points furthest from one of $p_{(n/2)+1}, \ldots, p_n$. Because the curtain is unique in the x-direction we can project it onto the yz plane to get a convex planar subdivision S[2].

We wish to locate the test point in a cell (or face) of the Voronoi diagram $\mathcal{V}(B)$. If the test point is $q = (x, y, z)$ then we project this into S, to get $q' = (y, z)$. If we locate q' in S, using a planar point location algorithm, we obtain the planar region(s) which it lies in. This corresponds to a facet(s), a two dimensional face, of $\mathcal{V}(B)$ which is part of the curtain. The facet lies between a cell due to a p_i, $1 \leq i \leq n/2$, and a p_j, $(n/2) + 1 \leq j \leq n$; and lies on the perpendicular bisector of p_i and p_j. The line L_q through q parallel with the x-axis passes through the curtain at this facet(s). The point q lies on the line either in the curtain or on one side of it.

If $\mathrm{d}(q, p_i) = \mathrm{d}(q, p_j)$ then q actually lies on this facet in the curtain. We have found the face of $\mathcal{V}(B)$ where q lies: we know p_i and p_j are both farthest from q.

If $\mathrm{d}(q, p_i) > \mathrm{d}(q, p_j)$ then q lies on the same side of the curtain as the

[2]This is planar because the faces are unique in the x-direction, the subdivision is convex because the faces of the curtain are convex as they are faces of the Voronoi diagram.

cell due to p_i (as it is on this side of the perpendicular bisector), and q will be found to lie among the cells due to $p_1, \ldots, p_{(n/2)}$, in other words its furthest point is one of $p_1, \ldots, p_{(n/2)}$. Similarly if $d(q, p_i) < d(q, p_j)$.

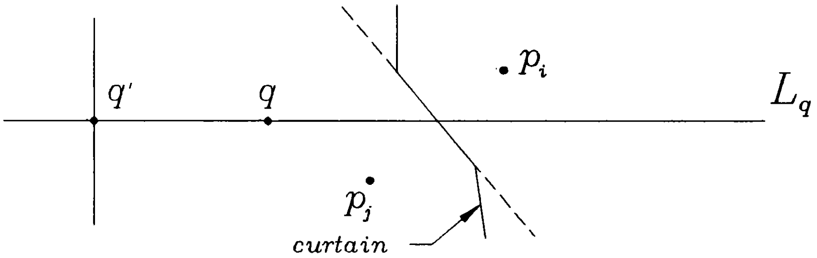


Figure 3.1: q lies among the cells due to $p_1, \ldots, p_{(n/2)}$.

The number of points of B that need to be considered has been (at least) halved. Effectively we have constructed a binary search among $p_1, \ldots, p_n$. The algorithm involves a preprocessing at each possible combination $p_r, \ldots, p_t$ of points that may be reached during the binary search. For each of these the Voronoi diagram must be built and S constructed and preprocessed for planar point location. After each step we have halved the number of possible candidates for the farthest point. An optimal algorithm, [19], is used to search S, this has linear preprocessing, and logarithmic query time. In section 3.4.3 I will show that this results in time bounds $P(n) = O(n^2)$ and $Q(n) = O((\log n)^2)$.

The algorithm is described from the bottom up. Firstly, in section 3.3 an algorithm of Kirkpatrick for the planar search will be described. Then in section 3.4 Chazelle's algorithm for the post office problem is described in detail. The information which is obtained in the course of Chazelle's algorithm needs to be converted to that required to perform Kirkpatrick's algorithm; section 3.4.2 will explain how to deal with this. Finally the time taken to perform the whole algorithm will be discussed in section 3.5.

3.3 Planar Search

David Kirkpatrick - Optimal Search In Planar Subdivisions

Chazelle suggests preprocessing and searching the plane using Kirkpatrick's algorithm [19]. This is a practical algorithm, that is optimal in terms of search time and space.

Kirkpatrick's algorithm is designed for searching triangular subdivisions. He describes how it can be extended to search a general planar subdivision, this will be discussed in section 3.3.3. The algorithm is described as it is presented in [19].

A *triangular subdivision* T is a finite convex planar subdivision (i.e. each line segment defining the subdivision is finite) all of whose regions, the external region included, are bounded by exactly three line segments. The *external region* is the unique unbounded region. Note that the subdivision can also be viewed as a planar graph, so graph theoretic ideas can also be used.

The algorithm exploits the fact that membership of a triangular region can be tested in constant time. If the three vertices are $u = (u_y, u_z)$, $v = (v_y, v_z)$ and $w = (w_y, w_z)$, then we can easily find an interior point: for example $(\frac{1}{3}(u_y + v_y + w_y), \frac{1}{3}(u_z + v_z + w_z))$. We check whether the test point p lies on the same side of each of the three edges as this interior point; only if this is true does p lie inside the triangle. This operation takes constant time.

If a convex planar subdivision has $|V|$ vertices then Kirkpatrick's algorithm allows us to locate a query point q' in $O(\log |V|)$ time, with $O(|V|)$ preprocessing and storage.

3.3.1 Basic Outline Of The Algorithm

Clearly we can't test membership of every triangle of the subdivision, as this would result in a linear query time. Instead, a search structure of triangles is constructed this is called the *subdivision hierarchy* over T. This is a sequence of triangular subdivisions each with sucessively fewer regions. We obtain these by repeatedly removing independent sets of vertices (from the previous subdivision) and retriangulating the regions thus left empty. This is done in a way which ensures that for each triangle in the new subdivision there are only so many 'parent' triangles in the previous subdivision, in other words each new triangle only overlaps a certain number of triangles on the layer beneath.

To search the region inside the bounding triangle start at the top subdivision (that with fewest triangles) and by testing membership of each of the triangles determine which contains the test point. Repeat this for the parents of that triangle in the layer beneath, and so on until finally the original subdivision T is reached and the answer is returned.

The hierarchy takes $O(|V|)$ time to construct and has height $\log(|V|)$, where $|V|$ is the number of vertices of T. At each stage we limit the number of triangles for which we have to test p's membership, to a maximum of a constant number.

The input that the algorithm requires is an *edge ordered representation* of T, this consists of the following:

1) The coordinates of the vertices v of T and an anticlockwise list of all directed edges of T source v.

2) A list $\{(v,w),(w,v)\}$ for each line segment v—w.

3) On edge (v,w) a pointer to the name[3] of the region immediately 'to the right of' (v,w), and a pointer between (v,w) and (w,v).

3.3.2 Theory Behind Kirkpatrick's Algorithm

Note that the size of a finite planar subdivision can be measured in terms of either the number of vertices, or the number of edges, as these are linearly related by Euler's formula [2, p.17]. This states that $f_2 - f_1 + f_0 = 2$, where f_k is the number of faces of dimension k. Each edge is in two regions, and we have a triangular subdivision which has (the maximal number of edges) exactly three edges per region $f_2 = \frac{2}{3}f_1$. Then if the number of vertices f_0 is n, and $n \geq 3$ we have that $f_1 = 3n - 6$ and $f_2 = 2n - 4$. We choose to measure $|T|$ using $|V|$ the number of vertices of T.

Theorem 21. *(Kirkpatrick) There is an $O(\log|T|)$ time $O(|T|)$ preprocessing and storage algorithm which will enable us to locate a point inside the triangular subdivision T.*

We search through a *subdivision hierarchy* on T which we can create in the required time. This is a sequence $T_1, \ldots, T_{h(|V|)}$ of triangular subdivisions such that each region R of T_{i+1} is linked to all regions R' of T_i for which $\text{int}R \cap R' \neq \emptyset$. The regions R' are called the *parents* of R in T_i. The first subdivision T_1 is T, and $h(|V|)$ is the *height* of the hierarchy.

The space required to store the hierarchy will be the sum of the spaces required for each of the individual subdivisions, which is $O(\sum_{i=1}^{h(n)} |T_i|)$; plus the space required for the links between the layers.

[3]Each region will be named by a pair $(p_i p_j)$ of points to reflect which facet of the curtain the region corresponds to.

The point location algorithm simply involves passing from $T_{h(|V|)}$ to T, and locating the test point q' at each level. Then looking at the parents of that region and locating q' in the level below. It is clear that this will locate q' in T.

Algorithm: Hierarchical Subdivision Search

$candidates_{h(|V|)}$:= regions of $T_{h(|V|)}$
R:=[4] region in $candidates_{h(|V|)}$ containing q'
i:= $h(|V|) - 1$
while $i > 0$ **do** $candidates_i$:= parents of (R)
  R:= region from $candidates_i$ containing q'
  i:= $i - 1$
report (region R)

The time taken to perform this algorithm is determined by the number of regions of which we have to test q''s membership: $O(\sum_{i=1}^{h(|V|)} |candidates_i|)$.

In the following section it is shown that it is possible to construct hierarchies where $h(|V|)$ and $|candidates_i|$ are limited. In other words we will limit the number of parents a region can have at each level, and also the number of levels.

Constructing The Subdivision Hierarchy

Firstly we show how to construct such a subdivision hierarchy of height two. In other words from T we construct in $O(|T|)$ time another triangular subdivision T' which is of smaller size (has less vertices) than T, and where we limit to d the number of parents regions of T' can have.

Lemma 22. *(Kirkpatrick) There exist constants $c, d > 0$ such that for any triangular subdivision T with $|T| > 3$ (i.e. non trivial) we can construct T' in $O(|T|)$ time where T' is a triangular subdivision and*
(i) $|T'| \leq (1 - \frac{1}{c}|T|)$
(ii) each region of T' has at most d parents in T.

Proof: (Kirkpatrick). We do this by first removing a vertex of T. Let v be any internal vertex of T, then there are deg(v) regions of T incident with

[4]By testing $q' \in R'$ $\forall R' \in$ candidates$_{h(|V|)}$.

v. We call the union of these regions the *neighbourhood* of v and denote this by $\text{nbd}(v)$. This is a star shaped polygonal region and has $\deg(v)$ bounding edges: each region in $\text{nbd}(v)$ is triangular and two of its edges are incident with v, the third edge of each triangle bounds $\text{nbd}(v)$.

We remove v and its $\deg(v)$ incident edges from T and retriangulate $\text{nbd}(v)$, this involves introducing $(\deg(v) - 3)$ new edges [25, p.12]. The result is a triangular subdivision on $|T| - 1$ vertices. In $\text{nbd}(v)$ we cover $\deg(v)$ regions of T with new triangles of T', so these new triangles overlap at most $\deg(v)$ regions of T. All the other regions remain fixed. No matter how we retriangulate, each region of T' intersects at most $\deg(v)$ regions of T.

Removing one vertex gives minimal simplification, but we can remove $v_1, \ldots, v_t$ a set of independent vertices in T to get our new triangular subdivision T'. Vertices are considered independent if they are non-adjacent in other words if $\text{nbd}(v_i)$ and $\text{nbd}(v_j)$ intersect in at most some boundary edges. After removing the vertices, all the vacated neighbourhoods can be retriangulated to form T'. Then T' is a triangular subdivision with $|T| - t$ vertices; each new region intersects at most $\max\{\deg(v_i) : 1 \leq i \leq t\}$ regions of T.

We can remove the edges and retriangulate in $O(|T|)$ time: Less than $|E|$ edges are removed and $|E| \propto |V| = |T|$. We replace these with less edges, $\sum \deg(v_i) - 3t$, than are removed. When we remove a vertex we leave a starshaped region, which it is known can be retriangulated in $O(|\deg(v)|)$ time where v is the vertex removed (see algorithms outlined in [27,14]). So the total time used to remove edges and retriangulate is $O(|T|)$ time.

To complete the proof of the lemma we need only identify $v_1, \ldots, v_t$ in $O(|T|)$ time, with $\deg(v_i) \leq d$, $1 \leq i \leq t$; and $t \geq |T|/c$ for some constants c and d. Lemma 23 below shows that we can find such a c and d but no attempt is made to optimise, as optimising may affect the time bounds adversely.

Lemma 23. *(Kirkpatrick) There exists $c, d \geq 0$ such that every planar graph with $|V| = n$ vertices has at least n/c independent vertices of degree at most d and we can find at least n/c of them in $O(n)$ time.*

Proof: (Kirkpatrick). We know that a triangulation on n vertices has the maximum number of edges possible for a planar graph, so a planar graph has at most $3n - 6$ edges. Each edge has two vertices so the average vertex degree is:

$$\frac{2 \cdot (3n-6)}{n} = 6 - \frac{12}{n} < 6.$$

Less than half of the vertices can have degree exceeding eleven.[5]

So consider the vertices of degree at most eleven. These form a set V' with $|V'| \geq \frac{1}{2}n$. We can identify these vertices in linear, $O(n)$, time by counting up to twelve edges out of every vertex and stopping. We can then eliminate these vertices straightforwardly to get an independent set by picking one element and eliminating its neighbours and so on. Each vertex in V' is adjacent to at most eleven vertices of V', so the resulting independent set has at least $|V'|/12 \geq n/24$ vertices, as $|V'| \geq n/2$.

So we have found $d = 11$ and $c = 24$ in $O(n)$ time. □

Thus to construct the subdivision hierarchy of lemma 22 takes $O(|T|)$ time: the vertices are identified in $O(|T|)$ time, and the edges are removed and the neighbourhoods retriangulated in $O(\sum \deg(v)) \leq O(|E|) = O(|T|)$ time (each edge can be removed at most once). □

We have found a subdivision hierarchy of height two. We iterate this procedure to obtain the required hierarchy.

Lemma 24. *(Kirkpatrick) $\exists c, d(>0)$ such that from any triangular subdivision T on n vertices, we can create an associated subdivision hierarchy $T_1, \ldots, T_{h(|V|)}$ in $O(|T|)$ time, with:*
(i) $|T_{h(|V|)}| = 3$,[6]
(ii) $|T_{i+1}| \leq (1 - \frac{1}{c})|T_i|$,
(iii) Each region in T_{i+1} has at most d parents in T_i.

Proof: (Kirkpatrick). Whenever $|T_i| > 3$, with $c = 24$, $d = 11$ we can find $|T_i|/c$ independent vertices in $O(|T_i|)$ time with the required property. Removing these and retriangulating takes $O(|T_i|)$ time. So the total time is of the order of

$$\sum_{i=1}^{h(|V|)} |T_i| \quad \leq \quad \sum_{i=1}^{h(|V|)} (1 - (1/c))^{i-1} |T_1|$$

[5] Less than half of the vertices can have degree twice the average or above as a vertex can't have zero degree.

[6] We can always reduce it if it's greater than three.

$$
\begin{aligned}
&= |T|(1 + (1 - (1/c)) + \cdots + (1 - (1/c))^{h(|V|)-1}) \\
&= |T|\frac{1 - (1 - \frac{1}{c})^{h(|V|)}}{1 - (1 - \frac{1}{c})} \\
&= |T|\frac{1 - (1 - \frac{1}{c})^{h(|V|)}}{1/c} \\
&< c|T| \\
&= O(|T|)
\end{aligned}
$$

as $h(|V|) > 0$. □

We can now prove theorem 21. To obtain the subdivision hierarchy the preprocessing time is $O(|T|)$ (lemma 24). The total space required to store the subdivisions and the links between the layers of the hierarchy is $O(\sum |T_i| + d \sum |T_i|) = O(|T|)$ (by the calculation above).

We need only show that the time required to perform a query is $O(\log |T|)$. First observe that $|T| = |T_1| \cdots |T_{h(|V|)}| = 3$ form at most a decreasing geometric progression with $|T_{i+1}| \leq \frac{c-1}{c}|T_i|$. Thus the subdivision hierarchy has height $O(\log |T|)$:

$$
\begin{aligned}
3 &= |T_{h(|V|)}| \leq ((c-1)/c)^{h(|V|)-1}|T| \\
|T| &\leq 3 \cdot (c/(c-1))^{h(|V|)-1} \\
\log |T| &\leq (h(|V|) - 1)C^{\,7} \\
h(|V|) &= O(\log |T|)
\end{aligned}
$$

Note that the top region $T_{h(|V|)}$ has one triangle. The time taken to perform the query is made up of testing membership of a triangle, a constant time question, in the following number of regions:

$$
\begin{aligned}
1 + \sum_{i=1}^{h(|V|)} |\text{parents}_R| &\leq 1 + \sum_{i=2}^{h(|V|)} d \\
&= 1 + O(\log |T|) \cdot d
\end{aligned}
$$

The algorithm query time is $O(\log |T|)$, which proves theorem 21. □

[7] Where $C = (\log(3) + \log c - \log(c-1))$ is a constant.

On completion of the algorithm we arrive at a triangle $\{v_1, v_2, v_3\}$, and the region that q' lies in must be named. If $(v_3)_y > (v_1)_y, (v_2)_y$; then we look at the region to the right (with respect to y) of $[v_1, v_2]$ and we have located the region that contains q'.

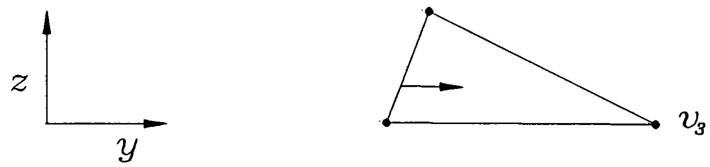


Figure 3.2: Naming the region that contains q'.

3.3.3 Searching The Convex Planar Subdivision

The planar search that has been described so far can only be used to search a triangular subdivision, such a subdivision is bounded and very specialised. During the course of Chazelle's algorithm we must locate a point in a general unbounded convex planar subdivision. This section describes how Kirkpatrick's triangular subdivision search is adapted to search this type of subdivision.

Assume an edge ordered representation (see section 3.3.1) of S is given, where S is a general unbounded convex planar subdivision. First we show how to create a triangular subdivision T from S by intersecting S with a large triangle. Kirkpatrick's algorithm can then be used to search the subdivision inside the large triangle. After that the method used for searching the external region is described.

Creating T from S

Claim 25 *(Kirkpatrick) The triangulation T of an unbounded convex planar subdivision S can be obtained in $O(|S|)$ time; $|T| = O(|S|)$.*[8]

The triangulation T of S will be formed by intersecting S with a large triangle, and then triangulating the resulting subdivision to obtain T. Recall that $|T|$ is the number of vertices of T. So in forming T we will gain the three vertices of the large triangle, and additional vertices where the

[8] T is a refinement of S so if we can locate a point in T we have located it in S.

unbounded edges intersect this triangle. Triangulating the region inside the large triangle does not change the number of vertices (it just increases the number of edges). So the number of vertices of T is at most $|V| + |E| + 3$, (assuming all edges are unbounded), where $|E|$ is the number of edges of S. As we have already observed Euler's relation implies that $|E| = O(|V|)$, so $|T| = O(|V|)$, in other words $|T| = O(|S|)$.

Assume an edge ordered representation of S. We must now show how an edge ordered representation of T can be obtained in the required time. As T is constructed new vertices and edges are formed, and the required information about each of these must be stored.

First we intersect the subdivision S with a triangle big enough to contain all of the vertices (all the intersections of the line segments) of the subdivision. It is easy to achieve this in the required time. For example one way of doing this is to find the maximum of $\mathrm{d}(0, v)$ over all vertices v of S, this will take $O(|V|)$ time; let this maximum be r', then all the vertices are contained in the closed ball $\bar{B}(0, r)$ for $r > r'$. We can find an equilateral triangle outside this ball: see figure 3.3. We need to find $h = r/(\sin \pi/6) = 2r$, and $c = r/(\tan \pi/6) = \sqrt{3}r$; so $v_1 = (-r, \sqrt{3}r)$, $v_2 = (-r, -\sqrt{3}r)$, $v_3 = (2r, 0)$. This orientation of the triangle was cho-

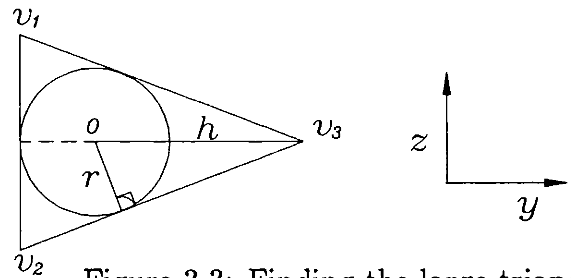


Figure 3.3: Finding the large triangle.

sen because in this case we need only name the regions 'to the right of' boundary edges along $[v_1, v_2]$.

Next, the unbounded edges must be intersected with the border of the large triangle $\Delta\{v_1, v_2, v_3\}$. All the required information about the new vertices must be obtained: that is their coordinates and the cyclic order of edges about them. We must also list the new line segments and name the regions to the right of the edges on $[v_1, v_2]$.

We assume that the information we have about the unbounded edges is the equation (and direction) of the half-line that forms each unbounded edge, the coordinates of the vertex that the edge radiates from, and a link to the names of the two regions lying on either side of the unbounded edge (v, ∞). It is not possible to just intersect all of the lines with the triangle and then order them as this would require an $O(|V| \log |V|)$ sort, and we need to achieve $O(|V|)$ time. One technique that achieves this time bound is described below.

The coordinates of the new vertices can be obtained in the following way: First pick an unbounded edge and intersect it with the large triangle $\{v_1, v_2, v_3\}$. To do this we intersect the edge (a half-line) with each of the lines bounding the large triangle and check whether the answer is in the relevant segment of that line. Look at the next edge anticlockwise and the next edge anticlockwise from the other end of that, until another unbounded edge is encountered; then intersect this edge with the large triangle, and repeat. Eventually we return to the initial edge. This method retains the cyclic ordering of the unbounded edges.

At each of the new vertices we need to record the anticlockwise order of the edges into that vertex. This can be incorporated into the above method. If edge (v, ∞) is the latest unbounded edge, and v' is its intersection with the boundary of the large triangle, then: create the edge (v, v') (in other words (v, v') and (v', v) with a pointer linking them). Let *prev* denote the last new vertex found and *next* denote the next new vertex that will be found. Create $(v', prev)$. The order of edges around v' is $(v', prev), (v', next), (v', v)$. We need to make special cases of the first unbounded edge that is picked; and whenever *prev* or *next* lies on a different edge of the large triangle from v': here set *prev* or *next* respectively to the relevant vertex v_i, $i = 1, 2, 3$. At this time we can also create the cyclic order of edges around v_i. Finally, if a newly created edge (v', w') (where $w' = prev$) lies along $\{v_1, v_2\}$ then the region to its right must be named. Simply create a pointer to the name common to the edges (v, ∞) and (w, ∞). This operation takes a constant time at each unbounded edge that is encountered, and involves a search along all edges that bound infinite regions, this can be at most $O(|E|)$ edges.

The previous operation clearly intersects S with a large triangle and obtains all the information about the new arrangement, the time taken is $O(|V|) + O(|E|) = O(|S|)$ time.

The final step necessary to obtain the triangulation of S is to triangulate the area inside the large triangle. Assume that the unbounded edges are distinguishable and excluded, and that we have an unordered list *LIST* of all bounded edges (v, w), each edge occurs in the list in both orientations. Also assume pointers to the associated regions to the 'right' $R(v, w)$, where $R(v, w) = \emptyset$ if $(v, w) \subseteq [v_2, v_3] \cup [v_1, v_3]$. Recall that the edge ordered representation contains an anticlockwise ordered list of edges source v, for each vertex. For ease of description I choose to represent the cyclic order of edges at v by two functions. Let $\mathrm{acw}_v(w)$ denote the next edge anticlockwise from (v, w) at v, with $\mathrm{cw}_v(w)$ defined similarly. Each of these functions can be stored in an array and I assume that these values can be updated and checked in constant time.

while $LIST \neq \emptyset$ **do:**
Pick edge (v, w) from *LIST*
    If $R(v, w) \neq \emptyset$ **and** $(v)_z > (w)_z$ **do:**
    $end := \mathrm{cw}_w(v)$
    $prev := v$
    $next := \mathrm{acw}_v(w)$
    $R := R(v, w)$
        while $next \neq end$ **do:**
        create $(w, next)$
        $R(w, next) := R$

        insert $(w, next)$ into cyclic order at w, before $(w, prev)$ by:
        $(\mathrm{cw}_w(prev) := next$ $\mathrm{acw}_w(next) := prev)$

        insert $(next, w)$ into cyclic order at next after $(next, prev)$ by
        $(X := \mathrm{acw}_{next}(prev)$
        $\mathrm{cw}_{next}(X) := w$ $\mathrm{acw}_{next}(w) := X$
        $\mathrm{cw}_{next}(w) := prev$ $\mathrm{acw}_{next}(prev) := w)$

        $prev := next$
        $next := X$
        end
    $\mathrm{cw}_w(prev) := \mathrm{end}$ $\mathrm{acw}_w(end) := prev$
    end
remove (v, w) from *LIST*
end

When we pick an edge from the list whose 'rightwards' adjoining region is triangular (this may have occured during the algorithm); or whose orientation is $(v)_z < (w)_z$; or if the edge lies on $[v_1, v_3]$ or $[v_2, v_3]$; we delete it from *LIST* in a constant time. There are $2|E|$ deletions from *LIST* in total. When we pick a suitable edge from *LIST* the region to its right is triangulated. The time to do this is linear in the number of edges of the region, and each edge is in two regions, so the total amount of time spent on triangulation steps is $O(|E|)$. (We only make $O(|V|)$ new edges at most as this is the size of a triangular subdivision.) Every region occurs in at least one label in the list, as each region is bounded, and so will be triangulated.

Thus a convex subdivision S can be triangulated to form T in $O(|V|)$ time, where $|V|(= |S|)$ is the number of vertices of S, so claim 25 stands.

Binary search of the external region

The planar subdivision encountered during the diameter algorithm is unbounded. At the initial stage of Kirkpatrick's algorithm we check whether the test point lies inside the largest triangle. If it does then Kirkpatrick's algorithm can be used to locate the point. This section deals with the case when the test point lies outside the large triangle. We need to be able to locate a point in this unbounded region in $O(\log |S|)$ time.

When we intersect the unbounded edges with the big triangle we can at the same time obtain an *acw* ordered circular list: $u', v', w', \ldots, v_1, \ldots, v_2, \ldots \ldots, v_3, \ldots, u'$ of all the vertices on the circumference of the triangle, each linked to the equation of the unbounded edge through that vertex. We have assumed that each unbounded edge has a pointer to the names $p_i p_j$, $p_k p_l$ of the two adjacent regions. Note that the lines do not intersect in the external region (by construction of the large triangle), so this ordering subdivides the space. The query takes the form of a binary search through this list.

When deciding that the point lies outside the large triangle we discover whether it lies on the 'correct' side of each of the bounding lines. The test point must lie on the correct side of at least one of the lines, for illustration let this side be $\{v_1, v_2\}$. Any edges radiating out from this side(s) of the triangle will be redundant in this search. All the vertices on this side (in the list between v_1 and v_2) can be removed and the cyclic list can be split here to give a list of vertices/edges that is suitable for a binary search. We can perform a binary search among these edges testing which side of an edge q' is on, until we locate q' in between two of them, they will have one

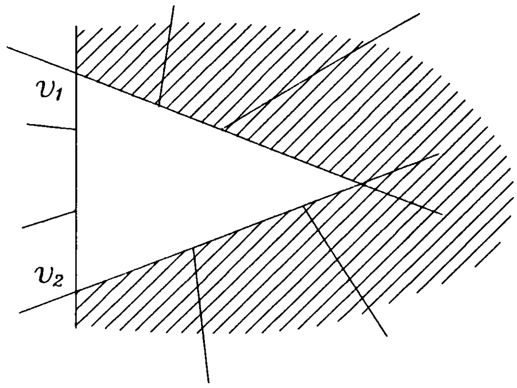


Figure 3.4: p must lie in the shaded region.

region label in common, and q' will be located.[9] Thus q' can be located in $O(\log|S|)$ time whether it lies inside or outside the triangular subdivision.

Given a convex planar subdivision S with $|S| = |V|$ the number of vertices of S it is possible, using Kirkpatrick's algorithm to locate a query point q' in S in time $O(\log|S|)$, with preprocessing and storage times of $O(|S|)$.

3.4 The Post Office Problem

B. Chazelle: How To Search In History

As I stated in the introduction, this paper [4] answers the 'farthest point post office query'; in other words in $\Re^3$ it will return the point from among $\{p_1, \ldots, p_n\}$ which is farthest from any given query point q in a query time $O((\log n)^2)$. The algorithm requires a preprocessing of $\{p_1, \ldots, p_n\}$ that takes time of $O(n^2)$, and storage $O(n^2)$. The algorithm will use the algorithm described in section 3.3 as a subroutine and can itself be used in

[9] In the case that q' is found to lie at the start/end of the list it is equally easy to find the name of the region containing q': simply find the next edge clockwise/anticlockwise from the first/last edge on the list and compare labels to name the region that contains q'.

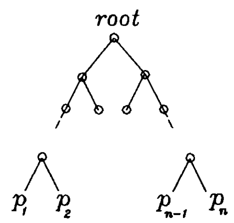


Figure 3.5: $\mathcal{T}_n$

the divide and conquer algorithm of Yao [29] (described in section 3.2.1) to give an algorithm that solves the diameter problem. The running times of Chazelle's algorithm will be calculated in section 3.4.3, and of the whole algorithm in section 3.5.

Recall from section 3.2 that answering the post office problem for a given query point q is equivalent to locating q in the farthest point Voronoi diagram. That section gave an overview of how the algorithm works, basically Chazelle's algorithm is a binary search through the farthest point Voronoi diagram of $\{p_1, \ldots, p_n\}$.

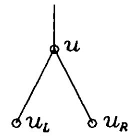
First assume that the points are ordered in the direction of one of the axes for example let $p_1, \ldots, p_n$ be such that $(p_i)_x < (p_j)_x$[10] whenever $i < j$. The search structure will be $\mathcal{T}_n$, a complete binary tree with objects $p_1, \ldots, p_n$ (in left to right order) as its leaves, see figure 3.5 above.

At each internal node u of $\mathcal{T}_n$ a second search structure is built, this will be the unbounded convex planar subdivision which is the projection of the 'curtain' $J(u)$. This subdivision will be searched using Kirkpatrick's algorithm. The Query Algorithm starts from the root of the tree and works its way down to the final answer/leaf, by performing a binary search through $\mathcal{T}_n$ that involves asking an $O(\log n)$ query (locating a point in $J(u)$) at each internal node visited, and from the result of this deciding which way to branch in the tree. The algorithm is based around a useful geometric property of Voronoi diagrams.

Let $I(u) = \{p_i : p_i \text{ is a leaf of the subtree rooted at } u\}$, where u is

[10]Strict inequality can be assumed here because there are a finite number of points so it is always possible to slightly perturb the x-axis to make the inequality strict.

an internal node of $\mathcal{T}_n$, and let u_L and u_R be the left and right sons of u respectively.



Let $J(u)$ be defined to be the set of faces in the Voronoi diagram of $I(u)$ whose 'construction points' lie in $I(u_L)$ and $I(u_R)$, in other words, if the notation $f_S(i,j)$ denotes the faces of $\mathcal{V}(S)$ that are supported by the bisector of p_i and p_j[11], then

$$J(u) = \{f_{I(u)}(i,j) : p_i \in I(u_L), p_j \in I(u_R)\}$$

So $J(u)$ consists of all of the faces of the 'curtain' (described in section 3.2) that lies between the space that is farther from one of $I(u_L)$ than any other point in $I(u)$, and the space that is farther from one of $I(u_R)$.

Consider the Voronoi diagram of the points $I(u) = \{p_r, \ldots, p_t\}$, and $J(u)$, those faces which are supported by bisectors between a point from $\{p_r, \ldots, p_s\}$ and a point from $\{p_{s+1}, \ldots, p_t\}$ where $s = (t-r)/2$, see figure 3.6 on page 73. $J(u)$ has a special property:

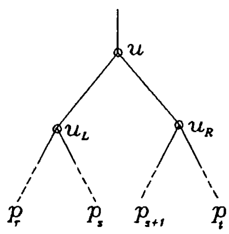


Figure 3.6: The nodes that lie beneath u in $\mathcal{T}_n$.

Lemma 26. *(Chazelle) Any line L parallel to the x-axis intersects one and only one face of $J(u)$.*

Though this may seem unlikely observe that $J(u)$ consists of only a few of the faces of $\mathcal{V}(I(u))$. Geometrically what has happened is that the

[11]Obviously the perpendicular bisector of p_i and p_j divides the space into regions farther from p_i than from p_j and vice versa.

Voronoi diagram $\mathcal{V}(I(u))$ has been partitioned into two distinct groups of cells, distinguished by greater and lesser x-values of the associated p_i's, and $J(u)$ is the faces that form the boundary between the two spaces. This lemma says that this boundary is unique with respect to a given (y, z). Note that since all of the faces of $J(u)$ are either facets or intersections of facets, we need only consider facets here.

Proof: (Chazelle). First given a line L parallel to the x-axis that *does* intersect some $f_{I(u)}(i, j)$ where $p_i \in I(u_L)$, $p_j \in I(u_R)$, we show this facet[12] is unique. Let this be the first such facet that point p hits as p travels along L in ascending x-order from $-\infty$. Notice that $i < j$, thus the vector $\vec{p_i p_j}$ has a positive x-coordinate: $(p_j)_x > (p_i)_x$, and this vector is normal to the facet $f_{I(u)}(i, j)$ by definition, as this facet is the perpendicular bisector of p_i and p_j.

As the point p travels along L it is initially furthest from points from $\{p_t, \ldots, p_{s+1}\}$. As it crosses the facet $f_{I(u)}(i, j)$ the farthest point from p changes from p_j to p_i. We now show that after this it can't have any p_k, $k \in I(u_R)$ as a farthest point.

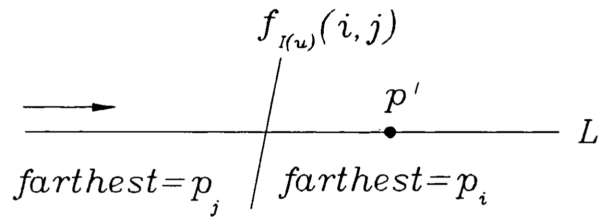


Figure 3.7: Line L parallel to the x-axis.

Let $p_i = (x_1, y_1, z_1) \in I(u_L)$, and let $p_k = (x_2, y_2, z_2)$ be any point in $I(u_R)$. The ordering of the points implies that $(p_k)_x > (p_i)_x$ so let $x_2 = x_1 + c$ for some $c > 0$.[13] Now let $p' = (x, y, z)$ be a point on L after the moving point has crossed $f_{I(u)}(i, j)$ and where p' is further from p_i than from any other point (see figure 3.7). Then in particular p' is farther from p_i than from p_k. If we use r_1^2 to denote $(y - y_1)^2 + (z - z_1)^2$, and r_2^2 for $(y - y_2)^2 + (z - z_2)^2$ then:

[12] Or intersection of facets.

[13] Recall that it was assumed that the points had distinct x-coordinates.

1) p' farther from p_i than p_k

$$\begin{aligned}(x-x_1)^2+r_1^2 &\geq (x-x_2)^2+r_2^2\\(x-x_2+c)^2+r_1^2 &\geq (x-x_2)^2+r_2^2\\r_1^2-r_2^2 &\geq (x-x_2)^2-((x-x_2)+c)^2\\&= -c^2-2c(x-x_2)\end{aligned}$$

If we now take another point p on L, $p=(x+k,y,z)$ which has p_k as its farthest point, then:
2) p farther from p_k than p_i

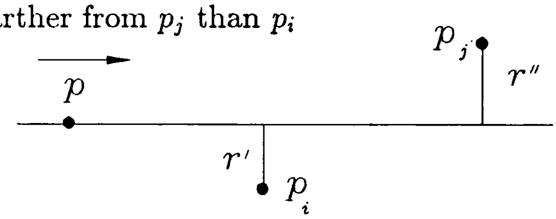
$$\begin{aligned}(x+k-x_2)^2+r_2^2 &\geq (x+k-x_1)^2+r_1^2\\&= (x+k-x_2+c)^2+r_1^2\\(x+k-x_2)^2-((x+k-x-2)+c)^2 &\geq r_1^2-r_2^2\\-c^2-2c(x+k-x_2) &\geq r_1^2-r_2^2\end{aligned}$$

3) Combining these gives:

$$\begin{aligned}-c^2-2c(x+k-x_2) &\geq -c^2-2c(x-x_2)\\2ck &\leq 0\end{aligned}$$

So as $c>0$ this implies $k\leq 0$. So after crossing $f_{I(u)}(i,j)$, p can't have any $p_k\in I(u_R)$ as a farthest point. Hence it can't (distinctly) cross another facet $f_{I(u)}(k,l)\in J(u)$. L intersects at most one face of $J(u)$. This proves the assertion.

Now it must be shown that every line L parallel to x-axis intersects one face of the curtain. Let L be any line parallel to the x axis (in other words y and z are fixed). Let a travelling point p start from $x=-\infty$. Initially the farthest point from p is in $I(u_R)$ (in fact it is p_t). Taking any two points p_i, p_j with $(p_i)_x=x'$, $(p_j)_x=x''$ such that $x''-x'>0$. Let r', r'' denote the perpendicular distances of p_i, p_j from L (it could be that $r'>>r''$). The following shows that it is always possible to find an x large enough that if $(p)_x=-x$ then p is farther from p_j than p_i



As $x''-x'$ is bounded and positive, and $(r'^2-r''^2)+(x'^2-x''^2)=D$ is bounded, it is possible to find an x with $2x(x'' - x') > D$ then:

$$\begin{aligned} r'^2 - r''^2 &< x''^2 - x'^2 + 2x(x'' - x') \\ x^2 + x'^2 + 2xx' + r'^2 &< x^2 + 2xx'' + x''^2 + r''^2 \\ (x + x')^2 + r'^2 &< (x + x'')^2 + r''^2 \\ \mathrm{d}(p, p_i) &< \mathrm{d}(p, p_j) \end{aligned}$$

Similarly we can find an x large enough such that $\mathrm{d}(p, p_i) > \mathrm{d}(p, p_j)$ if $(p)_x = x$. So initially the farthest of our points from p is in $I(u_R)$ and p will eventually end up having a farthest point in $I(u_L)$. Hence L must cross a face of $J(u)$.

So any line L parallel to the x-axis (with fixed y and z) passes through exactly one face of $J(u)$. This means that $J(u)$ can be directly projected onto the yz plane forming a planar graph, so that the projections of no two faces will intersect strictly (the boundarys may overlap). Hence $J(u)$ can be preprocessed for efficient planar searching (we can search the projection onto the yz plane).

3.4.1 Chazelle's Algorithm

Preprocess:
Sort points along x-axis $p_1, \ldots, p_n$
At each internal vertex u in $\mathcal{T}_n$:
  Create $\mathcal{V}(I(u))$.
  Find $J(u)$ from this graph (by a Depth First Search).
  Project $J(u)$ onto yz-plane and preprocess it for a planar point location (use Kirkpatrick's algorithm).
  Store this as $DS(J(u))$ (the data structure stored at vertex u).

Query:
Let the query point be $q = (x, y, z)$, locate this by:
  Perform a binary search through $\mathcal{T}_n$.
  Start at the root and at each vertex u encountered perform a planar point location of $q' = (y, z)$ in $DS(J(u))$. This returns a facet $f_{I(u)}(i, j)$, i.e. a pair $p_i p_j$.

Test $\max\{d(q,p_i), d(q,p_j)\}$
If i branch left, if j branch right, and continue.
If $d(q,p_i) = d(q,p_j)$ then return 'farthest is p_i and p_j'[a]. If leaf p_i is reached then return 'farthest is p_i'.

[a]as q lies on $f_{I(u)}(i,j)$ in this case

The test $\max\{d(q,p_i), d(q,p_j)\}$ tells us on which side of the division of the Voronoi diagram q lies. It simply tells us which half of L the point q lies on where L is cut into two half-lines by the perpendicular bisector of p_i and p_j at $f_{I(u)}(i,j)$. One half-line is farther from p_i, if p lies on this half-line then one of the points in $I(u_L)$ is the farthest point from q. We search then among that half of the points and their Voronoi diagram.

3.4.2 Preprocessing For Chazelle's Algorithm

The preprocessing involved in this algorithm is made up of several steps. Firstly the points must be sorted in the x-direction. Then at each internal node u, of the binary tree on these n vertices, the farthest point Voronoi diagram $\mathcal{V}(I(u))$ must be constructed. Finally this diagram must be searched to obtain the planar subdivision that is the projection of $J(u)$ into the yz-plane. The information obtained about this subdivision must be that necessary to search the subdivision using Kirkpatrick's algorithm.

Building the Voronoi diagram

We need to build the farthest point Voronoi diagram of $I(u)$ at each of the internal vertices u of $\mathcal{T}_n$. The next section describes how, if $|I(u)| = m$, the Voronoi diagram $\mathcal{V}(I(u))$ can be constructed in $O(m^2)$ time and storage.

The following argument is taken from Edelsbrunner [10, ch.13] which deals with nearest point Voronoi diagrams; the arguments can be straightforwardly adapted for farthest point Voronoi diagrams. To obtain the facial graph of the Voronoi diagram and the coordinates of its vertices, we first project the Voronoi diagram into space one dimension higher and show that it is equivalent to a polyhedron in this space. This polyhedron can be obtained by dualising the facets of the Voronoi diagram and finding the convex hull of the resulting points. The information obtained can then be converted into the required information about the original Voronoi diagram.

First we relate the Voronoi diagram $\mathcal{V}(B)$ of a set B of m points in $\Re^d$ to a polyhedron in $\Re^{d+1}$. Here we project $\mathcal{V}(B)$ from $\Re^3$ into $\Re^4$; we call the fourth axis *vertical* for convenience.

Consider the paraboloid $U : x^2 + y^2 + z^2 + K$, where K is a translation constant evaluated later. We define a transformation ε which maps each point $s \in B$ to a hyperplane in $\Re^4$: the tangent hyperplane to U at the point $U(s) = (s_1, s_2, s_3, (s_1^2 + s_2^2 + s_3^2 + K))$ which is the vertical projection of s onto U.

$$\varepsilon : s = (s_1, s_2, s_3) \rightarrow \varepsilon(s) : w = 2s_1x + 2s_2y + 2s_3z - (s_1^2 + s_2^2 + s_3^2) + K$$

If $s \in B$ ($\subseteq \Re^3$) and $p = (p_1, p_2, p_3)$ is any point in $\Re^3$ then let $\varepsilon(s, p)$ be the projection of the point p vertically onto $\varepsilon(s)$.

Lemma 27. *(Edelsbrunner)* $\mathrm{d}^2(p, s) = \mathrm{d}(U(p), \varepsilon(s, p))$

Proof:

$$\begin{aligned}
\mathrm{d}^2(p,s) &= (p_1 - s_1)^2 + (p_2 - s_2)^2 + (p_3 - s_3)^2 \\
\mathrm{d}(U(p), \varepsilon(s,p)) &= \mathrm{d}((p_1, p_2, p_3, (p_1^2 + p_2^2 + p_3^2 + K)) \text{ , } (p_1, p_2, p_3, (2s_1p_1 + \ldots \\
&\quad \cdots + 2s_3p_3 - (s_1^2 + s_2^2 + s_3^2) + K))) \\
&= \sqrt{(p_1^2 + p_2^2 + p_3^2 + K - (2s_1p_1 + \cdots + 2s_3p_3 - (s_1^2 + s_2^2 + s_3^2) + K))^2} \\
&= \mathrm{d}^2(p,s)
\end{aligned}$$

□

So there is a strong relationship between the distance between p and s in $\Re^3$ and the distance between the vertical projections of p onto U and onto $\varepsilon(s)$ in $\Re^4$.

We want to maximise $\mathrm{d}(p, s)$ to find the furthest $s \in B$ from p. In other words in $\Re^4$ we want to maximise $\mathrm{d}(U(p), \varepsilon(s, p))$ over $s \in B$. This means that the furthest $s \in B$ from p corresponds to the first hyperplane $\varepsilon(s)$ you meet as you increase x_{d+1} from $-\infty$, with $x_i = p_i$ for $i = 1, \ldots, d$ (the farthest one from the paraboloid). So we are interested in the space which lies below all of the hyperplanes.

It is possible to choose K large enough so that the origin lies below all of the hyperplanes, for this to be true all $\varepsilon(s)$ values should be greater than zero when $x = y = z = 0$. This holds if $K > \max\{s_1^2 + s_2^2 + s_3^2 : s \in B\}$

is chosen, K can be found easily in $O(m)$ time. With this value for K, $\varepsilon(s)^+$ is the halfspace lying towards the origin, and $P = \bigcap_{s\in B}(\varepsilon(s)^+)$, is the polyhedron lying below all of the hyperplanes.

Lemma 28. *(Edelsbrunner) There is a strong connection between $\mathcal{V}(B)$ and P.*

(i) If $\mathcal{P}(p)$ is the projection of $p \in \Re^3$ vertically onto the boundary of P then $\mathcal{P}(p) \subseteq \varepsilon(s) \iff \mathrm{d}(p,s)$ is a maximum over all $s \in B$.

(ii) For each k face f in $\mathcal{V}(B)$ there is a k-face f' in bd$P$ *and vice versa such that f is the vertical projection of f' into $\Re^3$.*

Proof: This follows from lemma 27, it is especially clear if you take the definition of Voronoi diagram as equivalence classes of points. □

The *facial graph* of a polytope or polyhedron is a graph with a node for each of its faces. The graph is layered by dimension of faces. A k-face f and a $(k+1)$-face g are joined by an edge if $f \subseteq \mathrm{cl}g$. This representation also includes the coordinates of its vertices and the supporting hyperplane of each of the facets. Hence if we build up the facial graph of the polyhedron P we have in effect created the facial graph of $\mathcal{V}(B)$ as the faces project directly into $\Re^3$. We also have bounds on the number of k-faces of $\mathcal{V}(B)$ following from the upper bound conjecture for P in $\Re^4$ [23].

Algorithms for convex hulls are much more familiar than those for intersections of halfspaces so the facial graph of the polyhedron is constructed by dualising the problem. By construction we have $\mathbf{0} \in P$. The *dual* or *polar set* [17,23,10] of a convex set P about $\mathbf{0} \in \mathrm{int}P$ is $P^* = \{y : <x,y> \leq 1, \forall x \in P\}$. This is an inclusion reversing map: if $Q \subset R$ then $R^* \subset Q^*$; which maps facets to vertices and k faces to $(d-k-1)$ faces. The dual of each face F of P is a face F^* of P^*, where $F^* = \{y : <x,y> = 1, \forall x \in F\}$. The latter relationship can be used to find all the possible vertices of P^* as they must each be related in this way to one of the hyperplanes. A total of m hyperplanes were projected onto the paraboloid, so if we perform this operation on each of these hyperplanes the dual faces will be m points in $\Re^4$. Finding the facial graph of the convex hull of these m points and an additional point at the origin, gives us full information about P. The point $\mathbf{0}$ is the dual of a facet[14] which cuts off all the unbounded faces at infinity:

[14]In this instance the polyhedron that is being considered is unbounded. Most of the

due to its construction (the projection onto a paraboloid) the polyhedron here is open.[15] The reason why it is necessary to have an additional vertex of P^* at the origin is as follows. Consider those points y of P^* which have inner product one when combined with points x of P. As $|x|$ increases, $|y|$ for the associated point in the dual decreases. We get a resulting cone into $\mathbf{0}$.[16] So for the dual we wish to find the convex hull of $m+1$ points, the origin being the extra point. Because the duality operation is inclusion reversing we can invert the facial graph and the facet equations obtained for the convex hull, and then delete the faces at infinity; the result is the polytope P and hence the $\mathcal{V}(B)$ we require.

Initially we had points $s \in B$, $s = (s_1, s_2, s_3)$. These points were mapped using $\varepsilon(s)$ to hyperplanes in $\Re^4$

$$w = 2s_1x + 2s_2y + 2s_3z - (s_1^2 + s_2^2 + s_3^2) + K$$

so

$$\begin{pmatrix} 2s_1 \\ 2s_2 \\ 2s_3 \\ -1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix} = s_1^2 + s_2^2 + s_3^2 + K$$

If $H = H(u, \alpha) = \{x :< x, u >= \alpha\}$, then the dual of this facet is $H^* = v = \{y :<x, y>= 1,\ \forall x \in H\} = u/\alpha$ and $v^* = H$ so

$$\varepsilon(s)^* = v_s = \frac{1}{s_1^2 + s_2^2 + s_3^2 - K}(2s_1, 2s_2, 2s_3, -1)$$

Standard algorithms [28,10] construct the facial graph of the convex hull of n points in $\Re^{d'}$, $d \geq 3$ in $O(n^{\lfloor (d'+1)/2 \rfloor})$ time, with $O(n^{\lfloor d'/2 \rfloor})$ storage. The upper bound conjecture [23] implies that this time and storage is optimal in even dimensions. Constructing the facial graph of the convex hull of our

literature deals with bounded polyhedrons and polytopes. In this case we do not have a strict dual (which is a one-to-one inclusion reversing map between faces of P and P^*) of P as extra faces will be obtained, on P these are all faces that lie on a facet that cuts off the unbounded polyhedron at infinity, and they correspond to the faces adjacent to the vertex at the origin in the polytope P^*.

[15]The vertical line through 0 meets all m planes as none of them are vertical, so the half-line beneath the last intersection point is contained in P: P is unbounded.

[16]The origin does not lie in the interior of the dual because if this were the case this would imply that P was bounded.

$m+1$ points in $\Re^4$, requires $O(m^2)$ time and storage. As $d = 3$, $|B| = m$ we have constructed $\mathcal{V}(B)$ in $O(m^2)$ time and $O(m^2)$ storage.

From this data we can immediately obtain the facial graph of P by deleting all of the faces into $\mathbf{0}$ and literally inverting the rest of the graph so that k-faces become $(d-k-1)$-faces; if we dualise the supporting hyperplane of each facet we get the coordinates of the corresponding vertex of P (we already know the equations of the hyperplanes of P). So we can construct P and hence by projecting vertically into $\Re^3$ we have $\mathcal{V}(B)$.

Note that though we were finding the graph of an unbounded polyhedron, by dualising we obtained the convex hull of a finite point set in other words the dual problem is bounded: Take any ball $B(\mathbf{0}, r) \subseteq \text{int}P$ then $B^*(\mathbf{0}, r) = B(\mathbf{0}, \frac{1}{r})$ and $P^* \subseteq B(\mathbf{0}, \frac{1}{r})$ as duality is inclusion reversing.

Obtaining $J(u)$

For the purposes of Chazelle's algorithm only data about $J(u)$ is required, not the actual Voronoi diagram. The following section describes a technique for acquiring this data in $O(m^2)$ time from the data obtained by the dual convex hull operation.

Assume the following information.

1) We have the facial graph of the Voronoi diagram, which includes additional faces at infinity, dual to faces containing $\mathbf{0}$ in the convex hull.

2) The 0-dimensional faces are labelled with their coordinates.

3) The 3-dimensional faces are labelled with the associated point p_i that they are farthest from.

The first part of this information can be obtained by inverting the facial graph of the dual convex hull. The second is obtained by dualising each of those facets not containing $\mathbf{0}$, to get a point in $\Re^4$ then simply deleting the fourth coordinate to obtain the projection into $\Re^3$. The cells of the Voronoi diagram can be labelled with p_i indicating which point $\varepsilon(p_i)^*$ was the equivalent vertex in the dual. The cell dual to $\mathbf{0}$ can be labelled ∞. This information can be obtained from the facial graph of the convex hull in the required time.

From this information we need to distinguish $J(u)$, and extract the information about its projection into the yz-plane that will enable us to perform Kirkpatrick's algorithm. As the vertices of $J(u)$ project directly into the yz-plane by simply removing their x-coordinate, and the line seg-

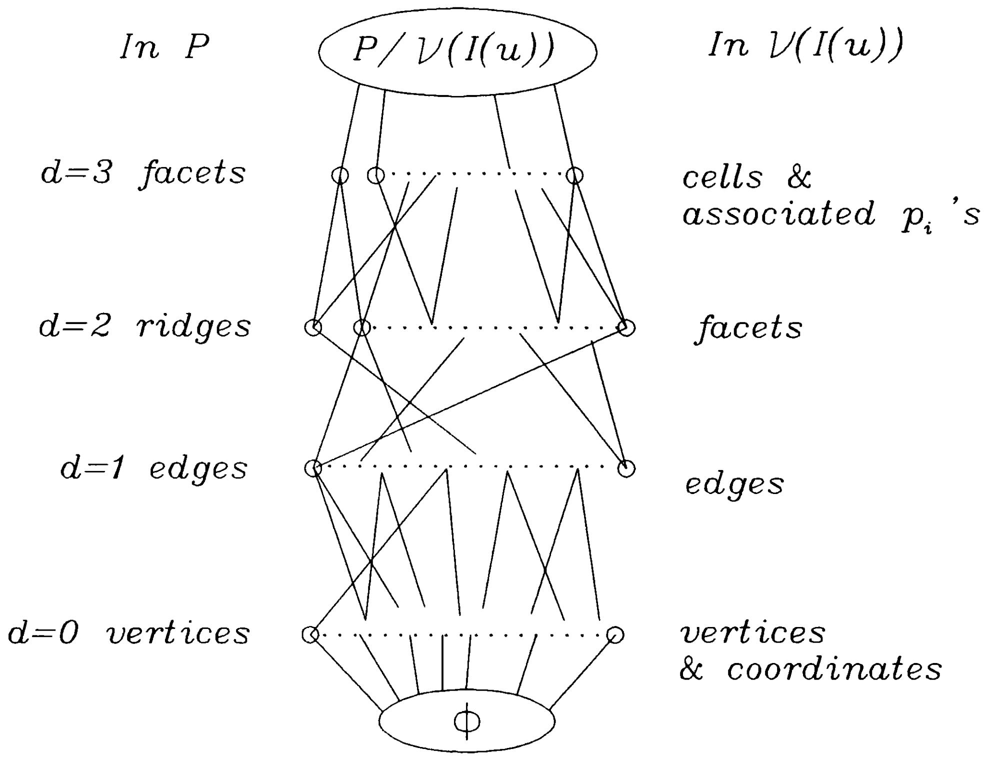


Figure 3.8: The facial graph of $\mathcal{V}(B)$.

ments will be represented by vertex pairs, hereafter the subdivision in the yz-plane will simply be referred to as $J(u)$.

Recall that to implement Kirkpatrick's algorithm we require the following information about $J(u)$.
1) The coordinates of vertices v (of $J(u)$) and an anticlockwise list of all directed edges (of $J(u)$) source v.
2) A list $\{(v,w),(w,v)\}$ for each line segment v—w (i.e. each edge of $J(u)$).
3) On edge (v,w) a pointer to the name of the region immediately 'to the right of' (v,w), and a pointer from (v,w) to (w,v).

Figure 3.8 shows the facial graph of $\mathcal{V}(I(u))$ that has been obtained. Firstly we need to select the relevant vertices and edges (and facets) of $\mathcal{V}(I(u))$. The initial technique we can use to obtain this information is a Depth First Search. The simplest description of this, ignoring unbounded edges is to move along the row of 2-dimensional faces and look at the superfaces[17] of each node here, asking 'am I in a p_i and a p_j cell with $i \in I(u_L)$, $j \in I(u_R)$?'. If the answer to this question is yes then we can search all edges on this 2-dimensional object (and their vertices). We find a vertex pair (v, w) for each bounded edge, create the edge, and create a pointer to the region name $p_i p_j$. If the edge has already been searched then we just need to add a second pointer this time to $p_i p_j$ (the edge has already been created).

This operation obtains the coordinates of all vertices on $J(u)$; the vertex pairs (v, w) that form the edges of $J(u)$; pointers from each edge to the facets $(p_i p_j)$ of $J(u)$ that contain it.

If $|I(u)| = m$ then $|\mathcal{V}(I(u))| = O(m^2)$: the upper bound conjecture implies that the number of incidences between $k/(k+1)$-faces of a polyhedron in $\Re^4$ is $O(m^2)$; if these incidences are summed over k the total number of incidences is $O(m^2)$, so this is the number of edges in the facial graph of $\mathcal{V}(I(u))$. The DFS to obtain $J(u)$ is $O(m^2)$. We need to search at least this much of the graph as $|J(u)| = O(m^2)$.[18]

The above description of the algorithm does not describe how the unbounded edges of the subdivision are dealt with, as these are not represented by a pair of vertices. A more detailed version of the algorithm, incorporating dealing with unbounded edges, is described below.

Locating Unbounded Edges.

Recall from page 68 that in order to implement the planar search algorithm we need to know which edges of $J(u)$ are unbounded, which vertex they leave from and in which direction, as well as the names of the adjacent regions. (We used this information to intersect them with a large triangle, and to search the area outside this triangle.) Recall also that we projected the (unbounded) Voronoi diagram into 4-dimensions and obtained a polyhedron with unbounded facets, ridges and edges. We dualised this and found a convex hull: a *bounded polytope*. What information has this yielded about the unbounded edges?

It is as if our polyhedron is cut off by an additional facet at infinity,

[17]The faces it is joined to in the row above.

[18]The maximum number of plane faces of $J(u)$ is $O(m/2 \cdot m/2)$, and as $J(u)$ is planar it also has $O(m^2)$ edges and $O(m^2)$ vertices, so the number of vertices $|V|$ by which the size of the subdivision $J(u)$ is measured is $O(m^2)$.

together with the ridges, edges, and vertices created where this facet intersects the unbounded faces. In the dual this facet transforms to a vertex v_0 at the origin, all the faces which contain this vertex are dual to faces at infinity in the polyhedron. We do not need to consider these faces at infinity as they do not lie on a $f_{I(u)}(i,j)$ hyperplane $i \in \{1,\ldots,(m/2)\}$, $j \in \{(m/2)+1,\ldots,m\}$.

Each unbounded edge of $J(u)$ has one vertex on the facet at infinity and one vertex in $J(u)$. If we can locate the relevant unbounded edges during the DFS, we can find out their directions by intersecting the facets that contain them, and by testing points either side of the vertex we can decide which is the correct half-line. This additional work takes $O(m^2)$ time for all unbounded edges (cf below).

During the search, as we did for bounded edges, we can obtain the names of the two regions $p_i p_j$ and $p_{i'} p_{j'}$, of $J(u)$ which border an unbounded edge. Each edge has precisely two such pairs because all edges lie between exactly two different plane faces $p_i p_j$ of $J(u)$ (a planar structure). The two regions are distinct and so at least three of the indices are different. For each unbounded edge in turn we intersect (in $\Re^3$) the two corresponding perpendicular bisectors, to obtain the equation of a line l along which the edge must lie, in constant time. We next decide which half-line out of v is the correct one. We move a distance ϵ_l along l from v and test for p_k, $k = 1,\ldots,m$ whether $\mathrm{d}(p_k, v+\epsilon_l) \leq \mathrm{d}(p_i, v+\epsilon_l)$, where p_i is one of the points that is farthest from v. If this is true then the point $v+\epsilon_l$ lies in the Voronoi diagram (on l) and this is the half-line (direction along l) that is required. If not the unbounded edge is the other half-line. The DFS also yields v the single (finite) vertex on the unbounded edge.

If $|I(u)| = m$ this operation takes $O(m)$ time for each unbounded edge, there are $O(m)$ unbounded edges[19]. The total time for this is $O(m^2)$.

[19] If we intersect the four dimensional polyhedron with a 3-plane which cuts all the unbounded faces, the result is a 3-dimensional polytope with vertices where the 3-plane cuts the unbounded edges, and facets where the unbounded facets of P cut the 3-plane there are at most m of these: P has at most m facets. Euler's relation for a 3-polytope says that $f_2 - f_1 + f_0 = 2$, where f_k is the number of faces of dimension k. There are at least three edges at each vertex, so if we count the edges by counting the vertices, we see that $f_1 \geq 3/2 \cdot f_0$ because each edge has been counted twice: once at each of its vertices. This implies that $f_2 - 3/2 f_0 + f_0 \geq 2$, so if $f_2 = m$ then $f_0 \leq 2m-4$, and $f_1 = f_2 + f_0 - 2 \leq 3m-6$. Hence this 3-polytope has at most $(2m-4)$ vertices.

Amended DFS of facial graph of Voronoi diagram

(1) Project each of the vertices into the yz-plane by removing the first coordinate of each.

(2) *DFS of faces at infinity:*
Mark facet at infinity ∞.
Using DFS mark all faces incident with this facet and faces on layers below with ∞.

(3) *Find the faces of $J(u)$:*
As explained in section 3.4.2 move along the faces f in the two dimensional row and if f is not marked infinity check the superfaces of f and ask 'am I in a p_i and a p_j cell $i \in \{1, \ldots, (m/2)\}$, $j \in \{(m/2)+1, \ldots, m\}$'.
For each facet where the answer to this question is *yes* search by DFS all edges of this face except those marked ∞ and:

(i) When we encounter an edge that has not yet been searched:
a) If both vertices are finite: create the edge as vertex pairs $(v, w)(w, v)$ with a pointer linking the pairs and a pointer from each to $p_i p_j$. Mark the edge as searched.
b) If one vertex is marked ∞: create the edge (v, ∞) with a pointer to $p_i p_j$; intersect the two perpendicular bisector planes linked to the edge to get the line of intersection; test which half-line from v is the correct one. Mark the edge as searched.

(ii) When an edge is encountered that has already been searched: create a pointer to $p_i p_j$.

In both cases (i) and (ii):
(iii) Check those vertices on the edge that are not marked ∞. If the label of a vertex is empty then: if the edge is finite label v and w with w and v respectively; if not label the finite node v with (v, ∞).[20] If the label of a vertex is not empty then it has already been searched during this $p_i p_j$ search. In this case link the new edge to the other $p_i p_j$ edge at that vertex using the label.

[20] It is necessary to be careful to distinguish between this edge and other unbounded edges from v.

At the end of each step set all the vertex labels to zero, and repeat for another 2-face.

(4) At each vertex v find the anticlockwise order of the edges around that vertex (see the section below).

(5) Create a pointer from each bounded edge to the name of the region that lies to its right (see the section below).

The DFS of steps (2) and (3) takes $O(|E|)$ time, where $|E|$ is the number of edges of the facial graph of $\mathcal{V}(I(u))$. We search each edge only once except in step 3iii) if the vertex is on an edge already searched, in this case the edges to the vertex nodes are traversed twice. All the other operations involved in steps (1), (2) and (3) take constant time. If $|I(u)| = m$ then $|E| = O(m^2)$ and performing the first three steps take a time of $O(m^2)$. The time requirement for steps (4) and (5) is considered below.

Anticlockwise order of edges around a vertex

Now we've been able to obtain all the vertices of $J(u)$ and all the edges: bounded edges represented by a vertex pair; unbounded edges by a half-line, and one vertex. Each edge has a pointer to two 2-dimensional regions.

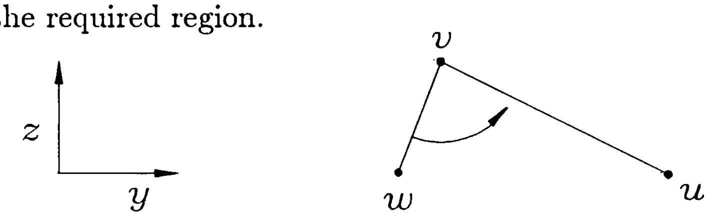
At each vertex v, an anticlockwise list of all directed edges source v is required. We know that no two regions on our convex planar graph have the same name (as this would contradict convexity). The name of a region which is adjacent to a vertex occurs in two consecutive edges in our cyclic ordering (the edges that the region lies between at the vertex). We have linked these.

We can take the vertices of $J(u)$ in turn. At each vertex, pick any edge out of the vertex, choose one of the labels on it and find the other edge from v with that label. Compare the tangents/slopes of these edges to check which order is anticlockwise, then taking them in that order, look at the other label on the second edge, and find the other edge from v with that label. Repeat this process until the first edge at that vertex is reached. We take each edge twice, once at each vertex. If $|E|$ is now used to denote the number of edges of $J(u)$, then this operation takes $O(|E|)$, which is $O(m^2)$, time.

Naming The Region To The Right Of Each Edge

Now all we need is to label each bounded edge (v, w) with the name of the region immediately to its 'right': $p_i p_j$. Where we take 'right' to mean in the positive y-direction (we may assume no edges are parallel to the y-axis).

for arc (v, w): if $(v)_z > (w)_z$ then find (v, u) the next arc anticlockwise from (v, w) at v. (v, w) and (v, u) each have two region names, the region common to both is the required region.



We do this for each edge taking a total of $O(|E|) = O(m^2)$ time.

In summary, at each vertex u of the binary tree with $|I(u)| = m$ we have taken $O(m^2)$ time to acquire all the information needed to perform Kirkpatrick's algorithm to search $J(u)$ which is a subdivision with $|V| = O(m^2)$ vertices.

3.4.3 Time Bounds For Chazelle's Algorithm

Preprocessing

1) Sorting points along x-axis takes a time of... $O(n \log n)$
2) *At each vertex u*

If the subtree from u has m leaves, $|I(u)| = m$ and this is the number of points we will be considering when we preprocess at u. Using the method described in section 3.4.2 we form the facial graph of $\mathcal{V}(I(u))$ the Voronoi diagram of the leaves under u. This was shown to take time... $O(m^2)$
We then obtain $J(u)$ from this using the algorithm described in section 3.4.2, this was shown to take time... $O(m^2)$

Then we preprocess the projected $J(u)$ data for Kirkpatrick's algorithm, this is linear in the number of vertices of the planar subdivision. We have $|J(u)| = O(m^2)$ vertices. This takes time... $O(m^2)$

So the total preprocessing at each vertex u with $|I(u)| = m$ is $O(m^2)$. As we have assumed that we are searching a binary tree we can assume that $n = 2^k$ for some k, then within the tree: 2^r vertices u have $|I(u)| = (n/2^r)$

leaves beneath them, where $r = 0, 1, \ldots, \log n$. If $T(m)$ is the time required for preprocessing when you have m leaves (this is $O(m^2)$),the total time is of order

$$\sum_{0}^{\log n} 2^r T(n/2^r) = n^2 \sum_{0}^{\log n} 1/2^r = n^2 \cdot 2(1 - (1/2n)).$$

Thus the total preprocessing is $O(n^2)$.

Query Time

Each planar point location is performed in a subdivision of size $|S| = |J(u)| = O(m^2)$ which is at most $O(n^2)$. Using Kirkpatrick's algorithm each query takes time of order $O(\log |S|) = O(\log(n^2)) = O(\log n)$ at each of the $\log n$ vertices that are considered as we move down the binary tree. Thus the total query time for each farthest point post office query is $O((\log n)^2)$.

Storage

Similarly the storage requirment is $O(m^2)$ for $\mathcal{V}(I(u))$ and $O(m^2)$ for Kirkpatrick's algorithm so the total storage, when this is all summed, is $O(n^2)$.

3.5 Time Bounds For The Diameter Problem

The algorithm answers the post-office problem within a time of $P(n) = O(n^2)$ for preprocessing, and $Q(n) = O((\log n)^2)$ for each query; it requires $O(n^2)$ storage. All that remains is to prove the claim of section 3.2 that with these time bounds the algorithm solves the diameter problem in a total of $O(n^{3/2} \log n)$ time. Recall that Yao's algorithm requires a total time of $rP(q) + nrQ(q) + O(n(r-1)) + O(n-1)$; where r is the number of subsets that the original set was divided into, and q is the maximum number of elements in each subset: $r = \lceil n/q \rceil$. The order of the time is considered below:

$$rq^2 + nr(\log q)^2 + n(r-1) + (n-1) = \frac{n}{q}q^2 + n\frac{n}{q}(\log q)^2 + (n(\frac{n}{q} - 1)) + (n-1)$$

If we pick $q = n^{\frac{1}{2}} \log n$, then the first term of this becomes $n^{\frac{3}{2}} \log n$, the second term becomes

$$\begin{aligned} \frac{n^{\frac{3}{2}}}{\log n}(\log(n^{\frac{1}{2}} \log n))^2 &= \frac{n^{\frac{3}{2}}}{\log n}(\frac{1}{2} \log n + \log\log n)^2 \\ &= \frac{n^{\frac{3}{2}}}{\log n}(\frac{1}{4}(\log n)^2 + \log n \log\log n + (\log\log n)^2) \end{aligned}$$

As $\log n > \log\log n$ the second term is also of order $O(n^{\frac{3}{2}} \log n)$. The third and fourth terms have lower orders. So the overall running time of this algorithm is $O(n^{\frac{3}{2}} \log n)$.

3.6 Summary

In this chapter I gave a detailed description of an algorithm that requires a worst-case time of $O(n^{3/2} \log n)$ to answer the 3-dimensional diameter problem. Previously only a sketch of this algorithm existed, so this is the first time that the constituent parts have been presented together in a unified form. For a long time this has been the best known algorithm, however during the preparation of this thesis I have heard reports that this longstanding problem has finally been completely answered by the discovery of an algorithm with an optimal time bound of $O(n \log n)$.

Bibliography

[1] A Aggarwal, *The art gallery theorem: its variations applications and algorithmic aspects*, Ph.D. thesis, John Hopkins Univ. (1984)

[2] B Bollobas, *Graph theory, an introductory course*, Graduate Texts in Mathematics, Springer Verlag, (1979)

[3] H Bunting, I Barany, DG Larman, J Pach, *Rich cells in an arrangement of hyperplanes*, to be published, (1994)

[4] B Chazelle, *How to search in history*, Found. Computation Th., Lecture notes in computer science, 158, Springer Verlag, (1983), 52-63

[5] V Chvátal, *A Combinatorial Theorem in Plane Geometry*, J. Combin. Theory Ser. B **18**, (1975), 39-41

[6] G Csizmadia, *Note on the art gallery theorem*, manuscript, (1992)

[7] G Csizmadia, G Toth, *Art gallery problem*, manuscript, (1993)

[8] L Danzer, B Grünbaum, V Klee, *Helly's theorem and its relatives*, Proc. Symp. Pure Math., **7** (convexity), (1963), 101-180

[9] D Dobkin, RJ Lipton, *Multidimensional searching problems*, SIAM J. Comput., Vol. **5** No.2, (1976), 181-186

[10] H Edelsbrunner, *Algorithms in combinatorial geometry*, Springer Verlag, (1987)

[11] P Erdös, *On sets of distances of n points*, Amer. Math. Monthly, **53**, (1946), 248-250

[12] P Erdös, *On sets of distances of n points in Euclidean space*, Publ. Math. Inst. Hungar. Acad. Sci., **5**, (1960), 165-169

[13] S Fisk, *A Short Proof of Chvátal's Watchman Theorem*, J. Combin. Theory Ser. B, **24**, (1978), 374

[14] A Fournier, DY Montuno, *Triangulating simple polygons and equivalent problems*, ACM Trans. on Graphics, Vol. **3** No. 2, (1984), 153-174

[15] JE Goodman, R Pollack, *Foundations of a theory of convexity on affine Grassman manifolds*, to be published, (1993)

[16] B Grünbaum, *A proof of Vázsonyi's conjecture*, Bull. Research Council of Israel, Section A, Vol 6A (1956), 77-78

[17] B Grünbaum, *Convex Polytopes*, Wiley, (1967)

[18] A Heppes, *Beweis einer vermutung von A Vázsonyi*, Acta Mathematica, Hung. Acad. Sci, **7**, (1956), 463-466

[19] D Kirkpatrick, *Optimal search in planar subdivisions*, SIAM J. Comput. **12**, (1983), 28-35

[20] DE Knuth, *The art of computer programming, Vol. 3 Sorting and Searching*, Addison Wesley, Reading, Mass., (1973)

[21] DG Larman, *On sets projectively equivalent to the vertices of a convex polytope*, Bull. London Math. Soc., **4**, (1963), 6-12

[22] M LasVergnas, *Hamilton paths in tournaments and a problem of McMullen on projective transformations in* $\Re^d$, Bull. London Math. Soc., **18**, (1986), 571-572

[23] P McMullen, GC Shephard, *Convex polytopes and the upper bound conjecture*, London Math. Soc. Lecture notes series, vol. 3, Cambridge University Press, (1971)

[24] W Moser, J Pach, *Recent developments in combinatorial geometry*, New Trends in Discrete and Computational Geometry, J Pach (Ed.), Springer Verlag, (1993)

[25] J O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, (1987)

[26] FP Preperata, MI Shamos, *Computational Geometry*, Springer Verlag, (1985)

[27] AA Schoone, J Van Leeuwen, *Triangulating a star-shaped polygon* Tech. Rep. RUVCS-80-3, Univ. of Utrecht, (1980)

[28] R Seidel, *Output-size sensitive algorithms for constructive problems in computational geometry*, PhD Thesis, Cornell University, (1986)

[29] AC Yao, *On constructing minimum spanning trees in k-dim spaces and related problems*, SIAM J. Comput., Vol. **11** No. 4, (1982), 721-736