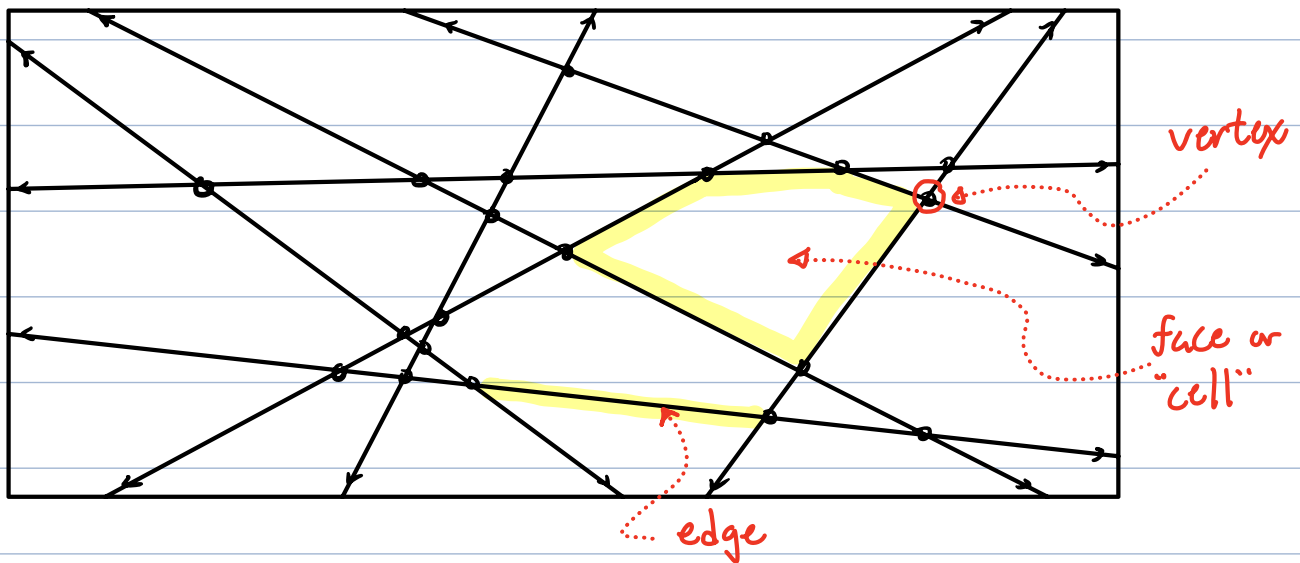


CMSC 754 - Computational Geometry

Lecture 13 - Line Arrangements

Arrangement:

Given a set $L = \{l_1, \dots, l_n\}$ of lines in \mathbb{R}^2 (generally $(d-1)$ -dim hyperplanes in \mathbb{R}^d), they subdivide the plane into a cell complex called the arrangement of L , or $A(L)$.



Combinatorial Properties:

Lemma: Given n lines L in gen'l position in \mathbb{R}^2 :

(i) $A(L)$ has $\binom{n}{2} = \frac{1}{2} \cdot n(n-1)$ vertices

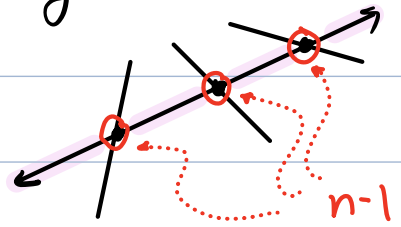
(ii) $A(L)$ has n^2 edges

(iii) $A(L)$ has $\binom{n}{2} + n + 1 = \frac{1}{2}(n^2 + n + 2)$ cells

Proof:

(i) Each pair intersects once = $\binom{n}{2}$ ✓

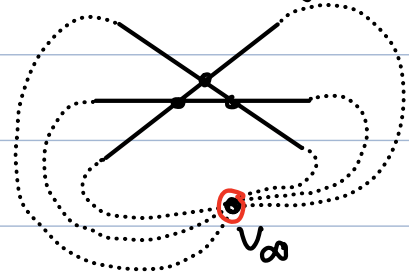
(ii) Each line is split by $n-1$ others into n edges
 $\Rightarrow n^2$ total \checkmark



(iii) Add a vertex at ∞ of degree n to tie off all unbounded edges

$$v = \binom{n}{2} + 1$$

$$e = n^2$$



By Euler's formula:

$$v - e + f = 2$$

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

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

$$\Rightarrow f = 2 + n^2 - \frac{n(n-1)}{2} - 1$$

$$= \frac{1}{2}(n^2 + n + 2) \checkmark \quad \square$$

[In \mathbb{R}^d , complexity is $\Theta(n^d)$]

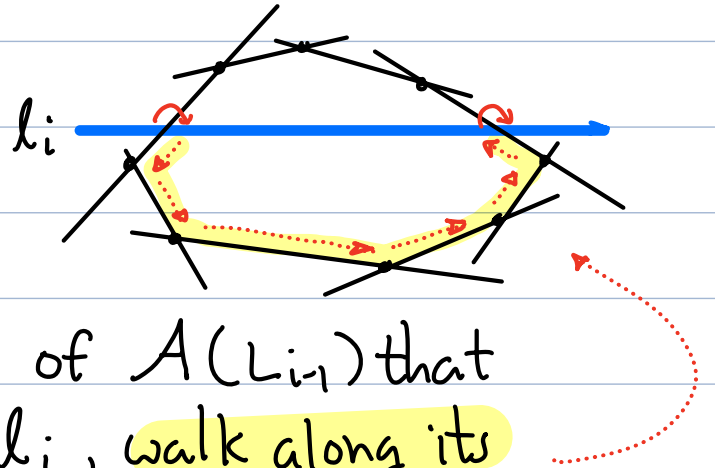
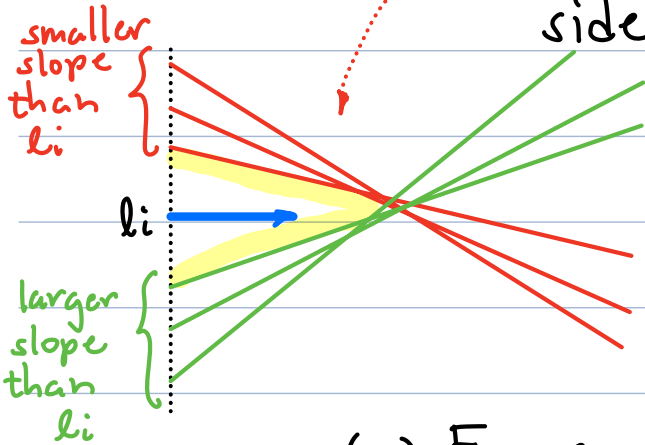
Incremental Construction: (not randomized)

Idea: Add lines one by one (in any order)
 Update the structure after each

Notation: $L_i = \{l_1, \dots, l_i\}$

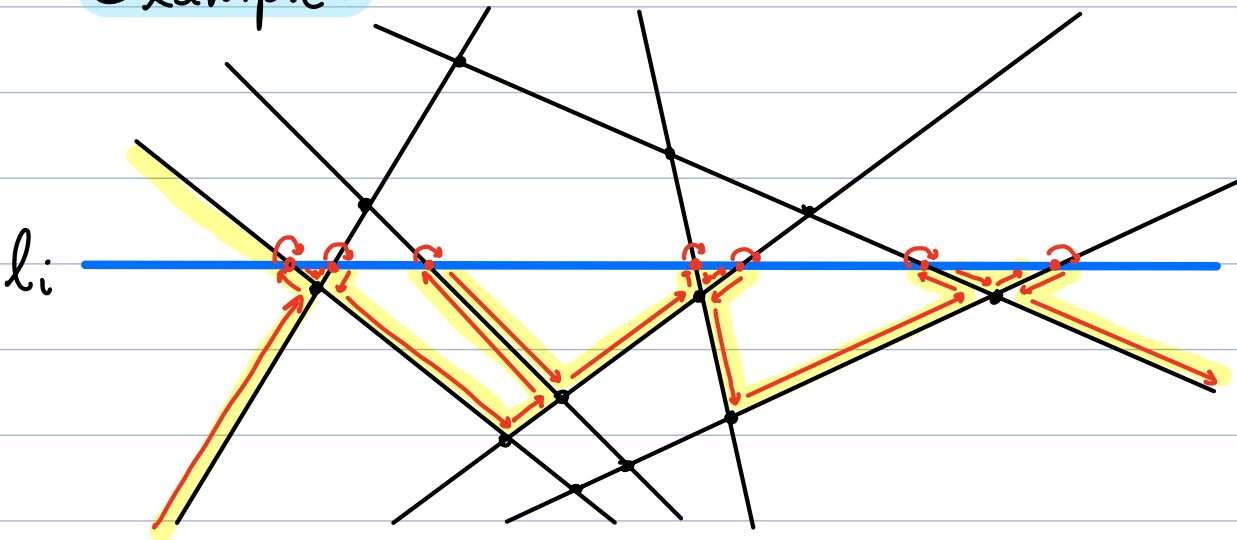
How to add the i^{th} line? l_i

(1) Find the unbounded cell on left side where l_i starts (slope based)



(2) For each face of $A(L_{i-1})$ that intersects l_i , walk along its lower boundary to determine where it exits this cell

Example:



- Once we know entry-exit points on each face - we update arrangement in $O(i)$ time (DCEL)
- How long to crawl around edges?

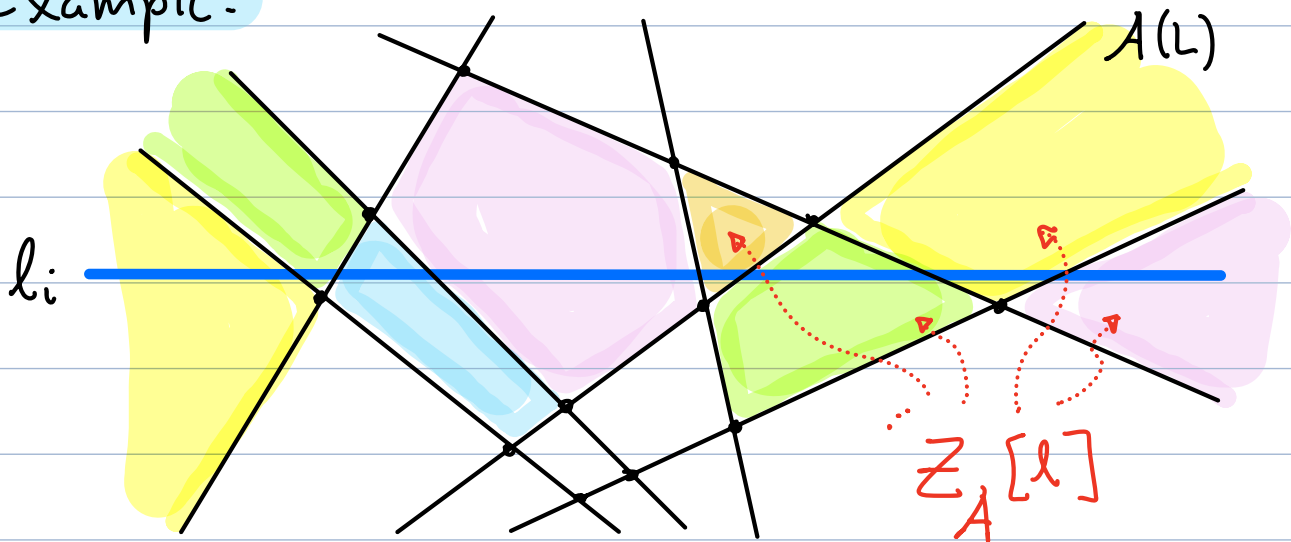
Naive analysis: On adding l_i

- l_i crosses i cells
- each cell may have as many as $i-1$ edges
- crawl takes $O(i(i-1)) = O(i^2)$ time
- total time $\approx \sum_{i=1}^n i^2 = O(n^3)$

Can it really be this bad?

Zone: Given an arrangement $A = A(L)$ and a line $l \notin L$, zone of l in A , $Z_A(l)$ is the set of cells of A that l intersects.

Example:

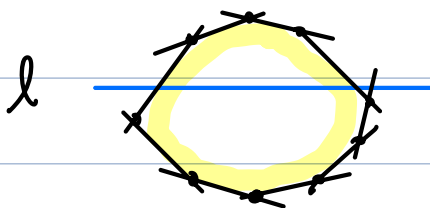


Obs: Crawl time \leq no. of edges on the zone of l_i in $A(L_{i-1})$ [$Z_{A(L_{i-1})}(l_i)$]
 \rightarrow We'll show this is $O(i)$ not $O(i^2)$

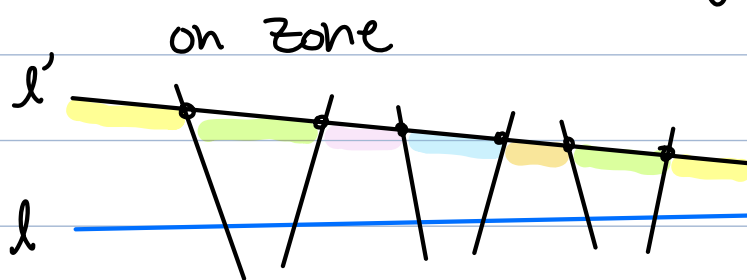
Theorem: (Zone Theorem) Given an arrangement $A(L)$ where $|L| = n$ and any line $l \in L$, the number of edges in $Z_A(l) \leq 6n$

How to prove this?

cell by cell? Some cells have high complexity



line by line? Some lines appear many times



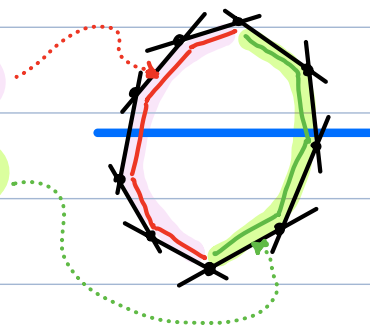
Our approach:

- Partition edges of zone into two classes (left side + right side)
- Show (by induction) at most $3n$ of each

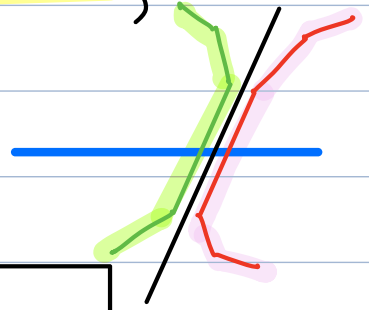
A zone edge is:

left bounding: on left side of cell

right bounding: on right side of cell



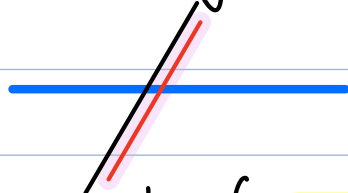
Note: Some edges appear twice in the zone, both as left/right bounding



Claim: At most $3n$ left-bounding edges.

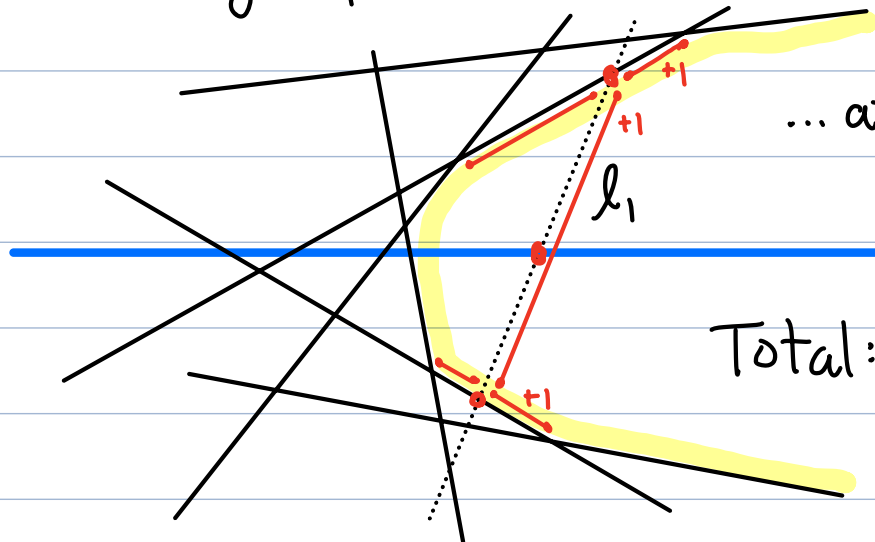
Proof: By induction on n

$n=1$: Just one LB edge $1 \leq 3 \cdot 1 \checkmark$



$n \geq 2$: I.H. arrangement of $n-1$ lines has $\leq 3(n-1)$ LB edges in zone

- Let $l_1 \in L$ be rightmost line to cross l
- Removing $l_1 \Rightarrow$ at most $3(n-1)$ LB edges
- Adding l_1 back creates...



... at most 3 new LB edges

Total: $\leq 3(n-1) + 3$
 $= 3n \quad \square$

Thm: Given a set L of n lines in \mathbb{R}^2 ,
 $A(L)$ can be built in time $O(n^2)$
[and has size $O(n^2)$... so this is optimal]

Proof: - Apply incremental construction

- Inserting l_i takes time \sim no. of
edges in $\sum_{A(L_{i-1})} (l_i) \leq 6(i-1)$

- Total time $\leq \sum_{i=1}^n 6(i-1) = 6 \sum_{i=0}^{n-1} i = O(n^2)$

Applications:

Line arrangements can be used to solve
many problems - mostly $O(n^2)$ time
- often using duality

How to process an arrangement?

- Build it + traverse it like a graph
 $O(n^2)$ time, $O(n^2)$ space

- Plane sweep

$O(n^2 \log n)$ time, $O(n)$ space

- Topological plane sweep

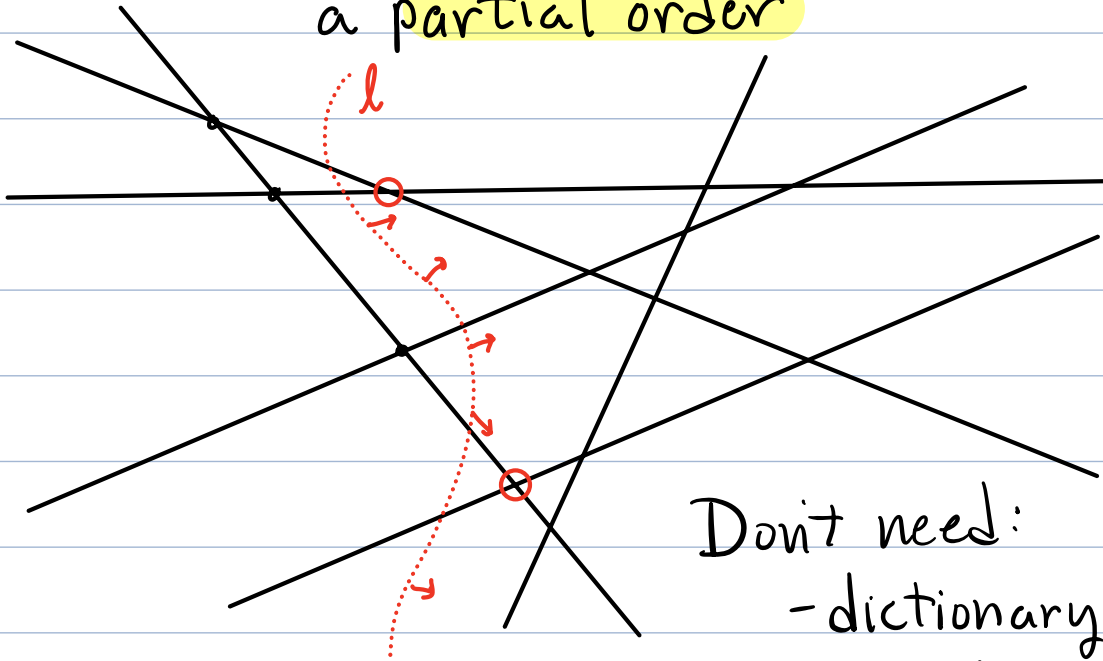
$O(n^2)$ time, $O(n)$ space

Not covered, but applicable pretty much
whenever plane sweep is.

... you may
assume
this

Topological plane sweep:

- A relaxed version of plane sweep
- Vertices are not swept in strict left to right order, but based on a partial order



Don't need:

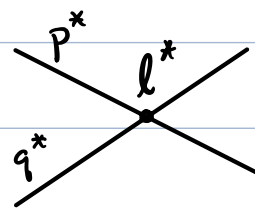
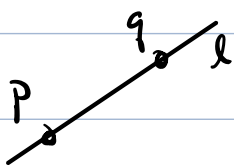
- dictionary
- priority queue

⇒ saves $\log n$ factor

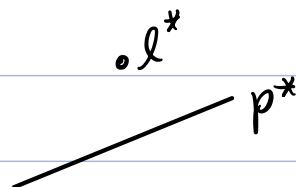
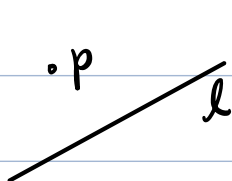
Recall: Dual transformation

$$p = (a, b) \longleftrightarrow p^* : y = ax - b$$

$$l : y = ax - b \longleftrightarrow l^* : (a, b)$$



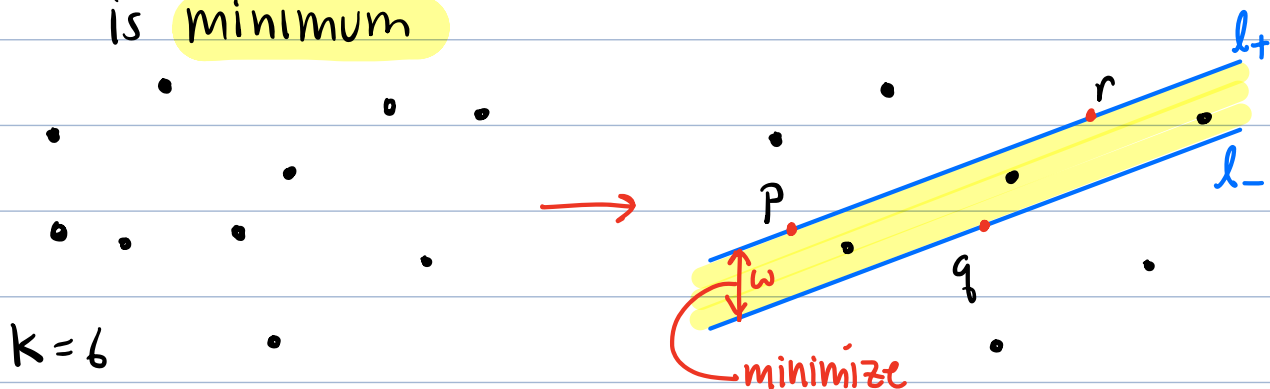
incidence preserving



order reversing

Narrowest k -corridor:

- Given a set $P = \{p_1, \dots, p_n\}$ in \mathbb{R}^2 and integer $3 \leq k \leq n$, find pair of parallel (non vertical) lines that enclose k pts so that vertical distance between lines is minimum



Primal form:

- Let l_+ + l_- be upper + lower lines of "slab"

$$l_+ : y = ax - b_+ \quad b_+ \leq b_-$$

$$l_- : y = ax - b_-$$

- Vertical width: $w = b_- - b_+$

- k pts of P lie on or between l_- + l_+

Local optimality:

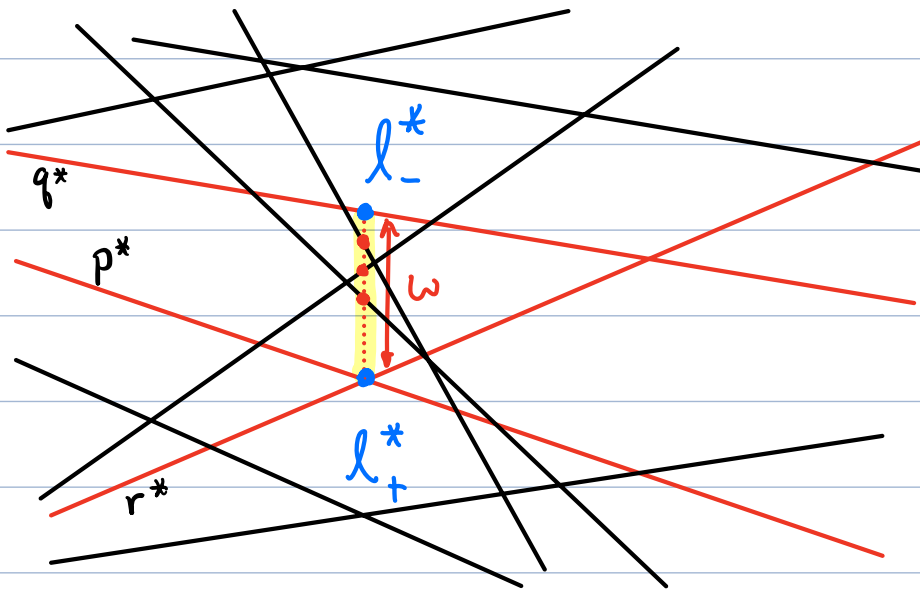
3 pts of P will lie on l_+ + l_- , 2 on one edge + 1 on other

- If 0, 1, or 2 can make width smaller
- If 4 or more - not gen'l position

Dual form:

- $l_+^* + l_-^*$ are pts (a, b_+) + (a, b_-)
- vertical distance $b_- - b_+$
- k lines of P^* pass through or between these pts

vertical line segment

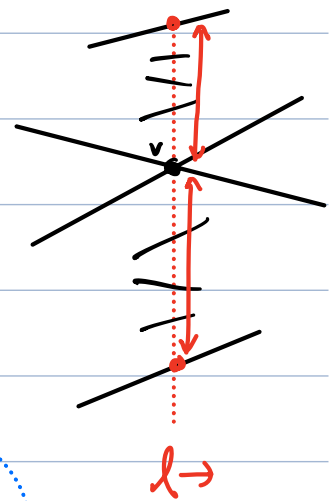


Local optimality:

3 lines will pass through $l_-^* + l_+^*$ with 2 on one side + one on other

Narrowest-Corridor (P, k) :

- (1) $P^* \leftarrow$ dual lines of P
- (2) Plane sweep through P^* .
- (3) On arriving at each vertex v , compute vertical distance to lines $k-2$ above + $k-2$ below
- (4) Return smallest such distance



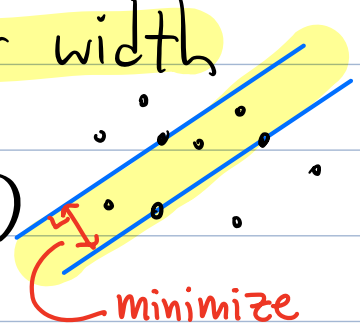
Correctness: (Argued above)

Can access in $O(1)$ time since sweep line can be stored in array

Time: $O(n^2 \log n)$ time + $O(n)$ space

↳ can reduce to $O(n^2)$ by topol. plane sweep.

Aside: It is easy to generalize this to minimize perpendicular width (Just apply a correction factor when computing widths)

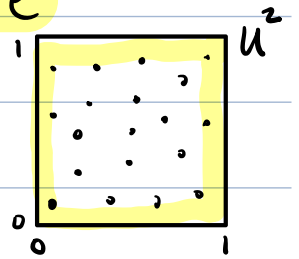


Halfplane Discrepancy:

Let $U = [0, 1]^2$ denote the unit square

Given n pts $P = \{p_1, \dots, p_n\} \subset U$,

how close is P to being uniformly distributed over U ?



Idea:

For any halfplane h , let

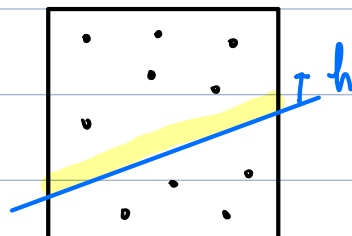
$$\mu(h) = \text{area}(h \cap U^2)$$

$$[0 \leq \mu(h) \leq 1]$$

the fraction of P in h

$$\mu_p(h) = |h \cap P| / |P|$$

$$[0 \leq \mu_p(h) \leq 1]$$



$$\mu(h) = 2/3 = 0.666\dots$$

$$\mu_p(h) = 6/10 = 0.6$$

If P is uniformly distrib., we expect

$$\mu(h) \approx \mu_P(h) \quad \forall h$$

To measure how uniform is P , define:

$$\Delta(P) = \max_h |\mu(h) - \mu_P(h)|$$

Called the halfplane discrepancy of P $[0 < \Delta(P) \leq 1]$
can't be perfect

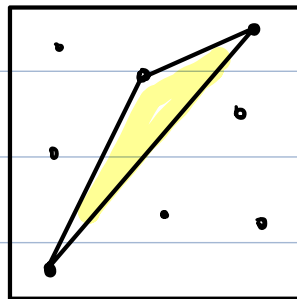
Questions:

- Given $P \subset U^2$, what is $\Delta(P)$?
- How low can $\Delta(P)$ be for any set of size n ?
- How to generate optimally uniform set P_{opt} of a given size n ?
($\Delta(P_{\text{opt}})$ is min. possible)
- Other measures of discrepancy?
 - Triangle discrepancy
 - Heilbronn's Triangle Problem:

Given any set of n pts P in U^2 ,
how large can the min area
triangle be?

(conj: $O(1/n^2)$)

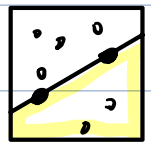
Open for a century!



Computing $\Delta(P)$ for a set $P \subset U^2$.

- Key: Identify $O(n^2)$ candidates for halfplane that maximizes discrepancy.
- Compute discrepancy for each
 - Return the max

Lemma: Given pt set P , let h be halfplane of max discrepancy. Let l be h 's bounding line. Either:



- (i) l passes through pt $p_i \in P$, and p_i is midpoint of $l \cap U^2$
- (ii) l passes through two pts of P .

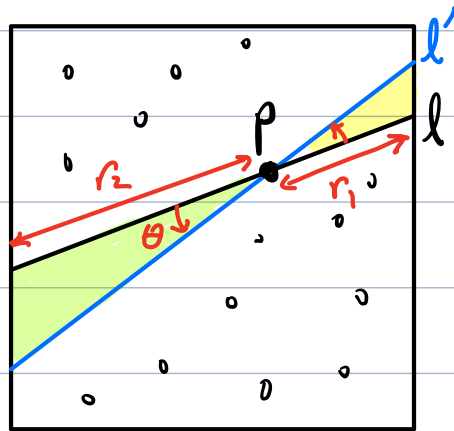
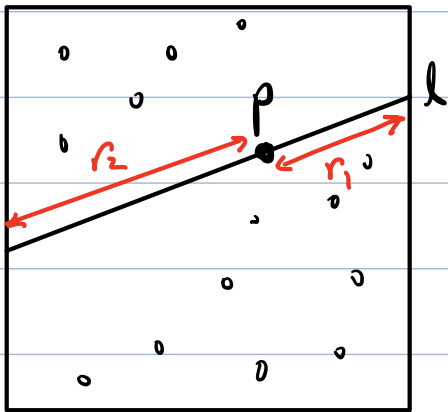
Proof:

Approach: Consider any line l . We'll show unless it satisfies (i) or (ii) we can perturb it to increase discrepancy.

Case 1: l passes through no pt of P - perturbing l up or down increases discrepancy.



Case 2: l passes through a pt $p \in P$, but p is not midpt of $l \cap U^2$



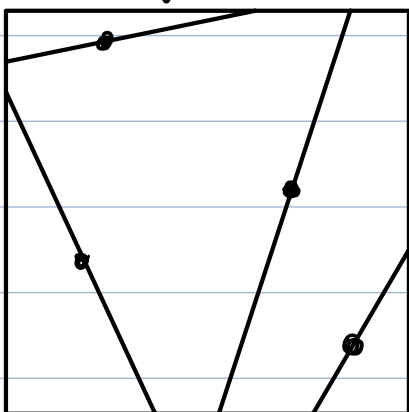
p splits $l \cap U^2$ into two segments of lengths r_1 and r_2 . Since p is not midpt, may assume w.l.o.g. $r_2 > r_1$

If we rotate l by small angle θ about p we increase/decrease area by \sim

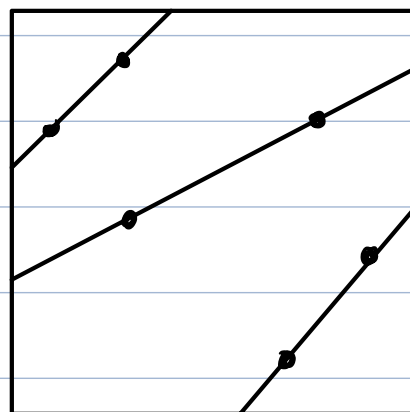
$$r_2^2 \cdot \theta - r_1^2 \cdot \theta = (r_2^2 - r_1^2) \theta > 0$$

Some small rotation will increase discrepancy. □

Type (i)



Type (ii)



Computing $\Delta(P)$:

Type (i):

- for each $p_i \in P$, compute lines l
s.t. p_i on midpt of $l \cap U^2$
- Count no. of pts on either side of l
 $\rightarrow n$ pts; $O(1)$ lines each; $O(n)$ time
to count $\Rightarrow O(n^2)$ time

Type (ii):

- Dualize P to P^*
- Perform plane sweep of arrangement $A(P^*)$
- For each vertex of arrangement
maintain no. of lines above +
below on sweep line
- Compute discrepancy in $O(1)$ time
for each vertex

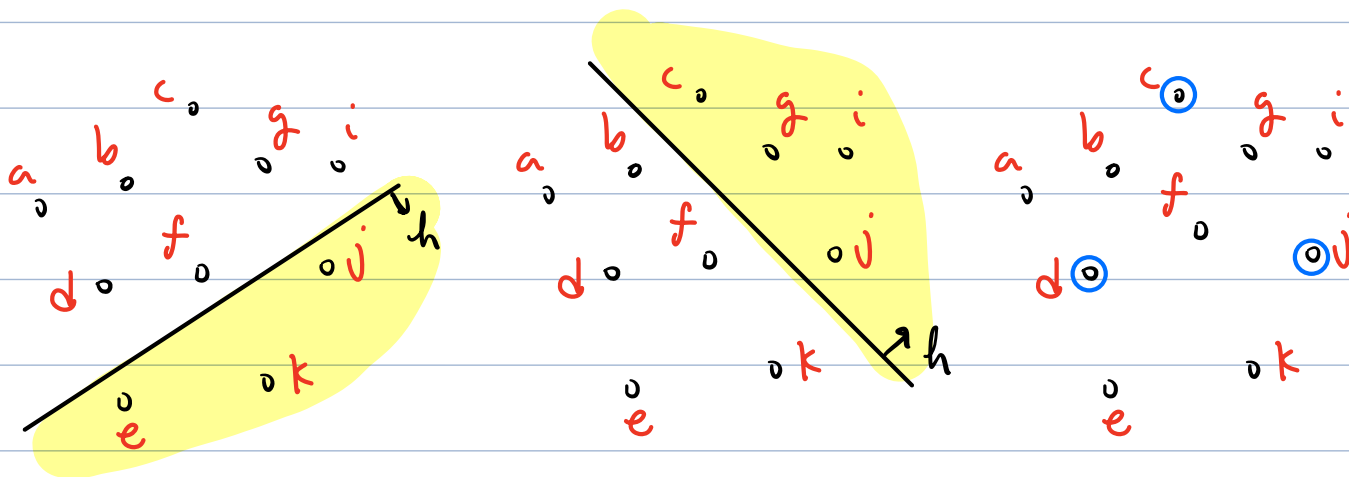
$\rightarrow O(n^2)$ vertices

Can maintain counts in $O(1)$ time
 $\Rightarrow O(n^2 \log n)$ time + $O(n)$ space

$O(n^2)$ by topol plane sweep

Computing k -sets:

Given a set $P = \{p_1, \dots, p_n\}$ in \mathbb{R}^2 and integer k , $1 \leq k \leq n-1$, a k -set is a k -element subset of P of the form $P \cap h$, for some halfplane h .



$\{e, k, j\}$

is a 3-set

$\{c, g, i, j\}$

is a 4-set

$\{c, d, j\}$

is **not** a 3-set

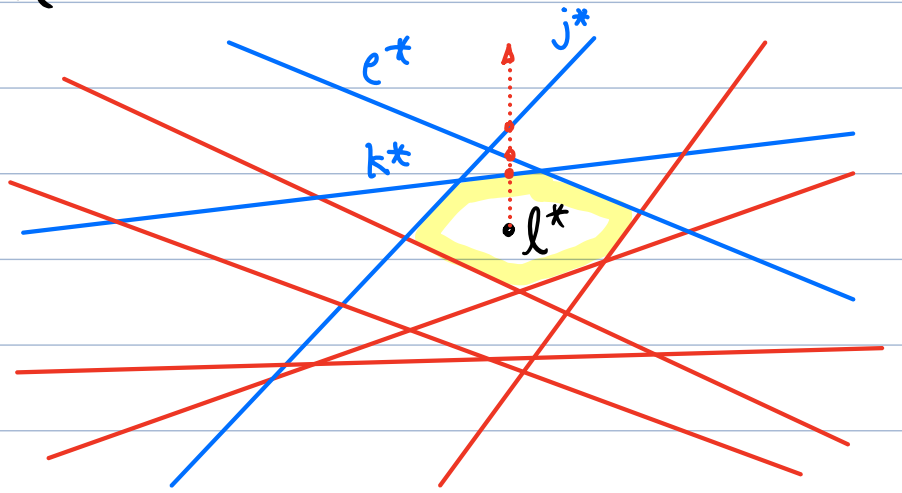
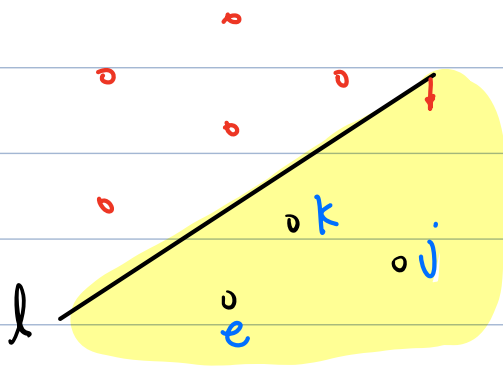
Problem: Given P and k , enumerate all k -sets of P .

How many? Naive $\leq \binom{n}{k} = \mathcal{O}(n^k)$

Better $\leq \binom{n}{2}$ (see below)

Best theoretic bounds: $\mathcal{O}(n \log k)$, $\mathcal{O}(nk^{1/2})$

Dual equivalent?



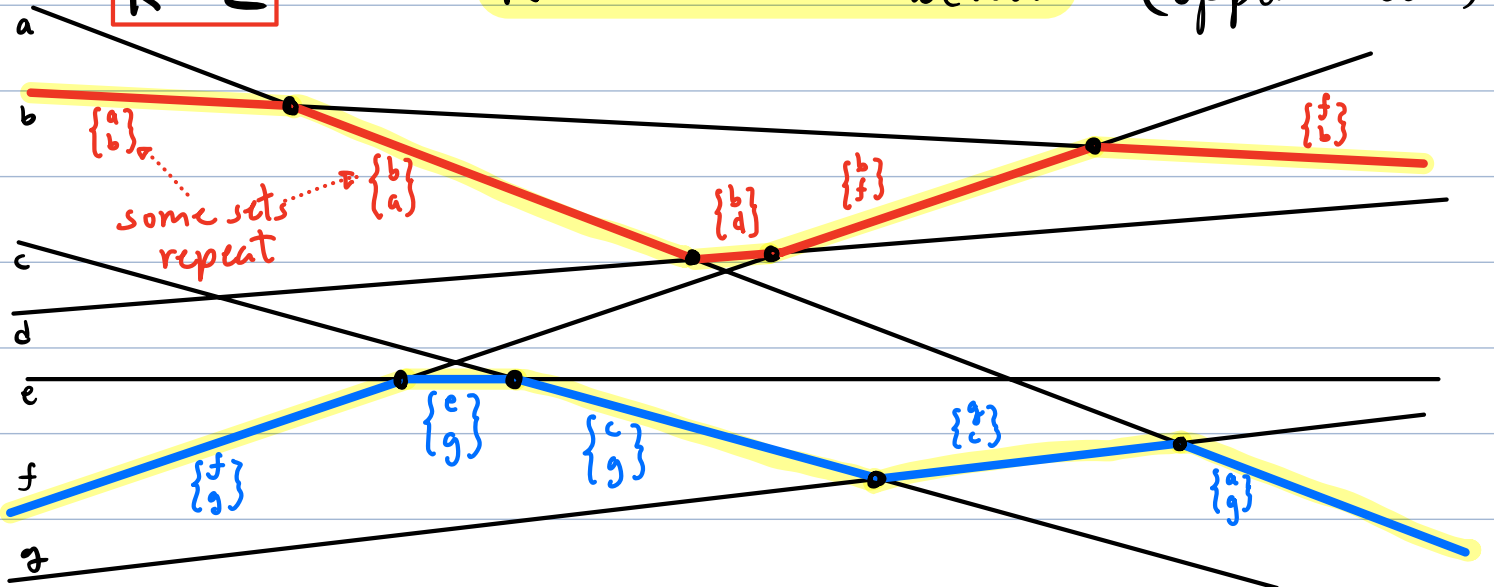
By order reversal:

k -pts of P lie below $l \iff k$ -lines of P^* pass above l^*

Approach:

- Traverse the arrangement $A(P^*)$
- Identify all edges with k lines on or above (lower k -set)
- k " " " below (upper k -set)

$n=7$
 $k=2$



Level: Given an arrangement of n lines $A(L)$, for $1 \leq k \leq n$, define level k , L_k , to be set of pts in $A(L)$ with

$\leq k-1$ lines (strictly) above

$\leq n-k$ lines (strictly) below

In above figure, we have shown L_2 and L_6

Obs: By applying plane sweep through $A(L)$, we can construct all levels in time $O(n^2)$

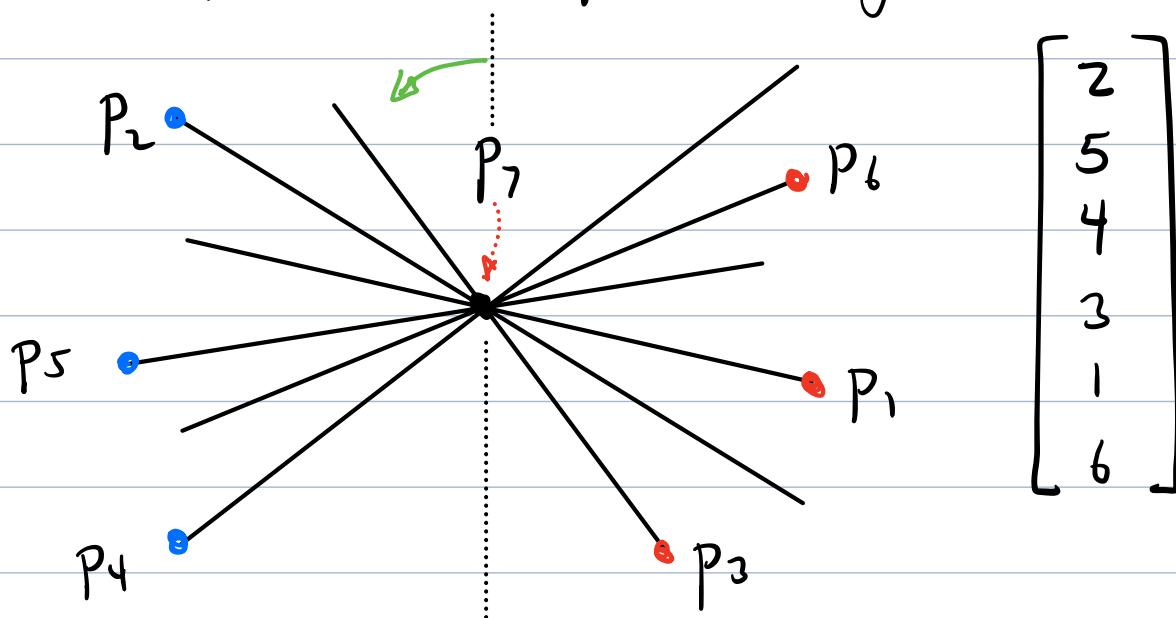
\Rightarrow Can identify all k -sets of P in time $O(n^2)$ by sweeping $A(P^*)$ + extracting levels $L_k + L_{n-k+1}$

Note: To actually list the sets adds additional k factor, total $O(k \cdot n^2)$

Avoid duplicates? Exercise

Sorting angular sequences:

Given a set $P = \{p_1, \dots, p_n\}$ in \mathbb{R}^2 ,
for each p_i , sort the remaining $n-1$ pts around p_i in angular order.



Naive: $O(n(n \log n)) = O(n^2 \log n)$
Sort angles for each point

Better: $O(n^2)$ using arrangements.

[see lect. notes for details]

Summary:

- Arrangements of lines in \mathbb{R}^2
- Complexity = $O(n^2)$
- Constructable in $O(n^2)$ time
(incremental, deterministic)
- Zone Theorem: Total zone complexity is $O(n)$. [$O(n^{d-1})$ in \mathbb{R}^d]
- Topological plane sweep
 $O(n^2)$ time ; $O(n)$ space
- Applications:
 - Narrowest k-corridor
 - Halfplane discrepancy
 - k-sets
 - Sorting angular sequences