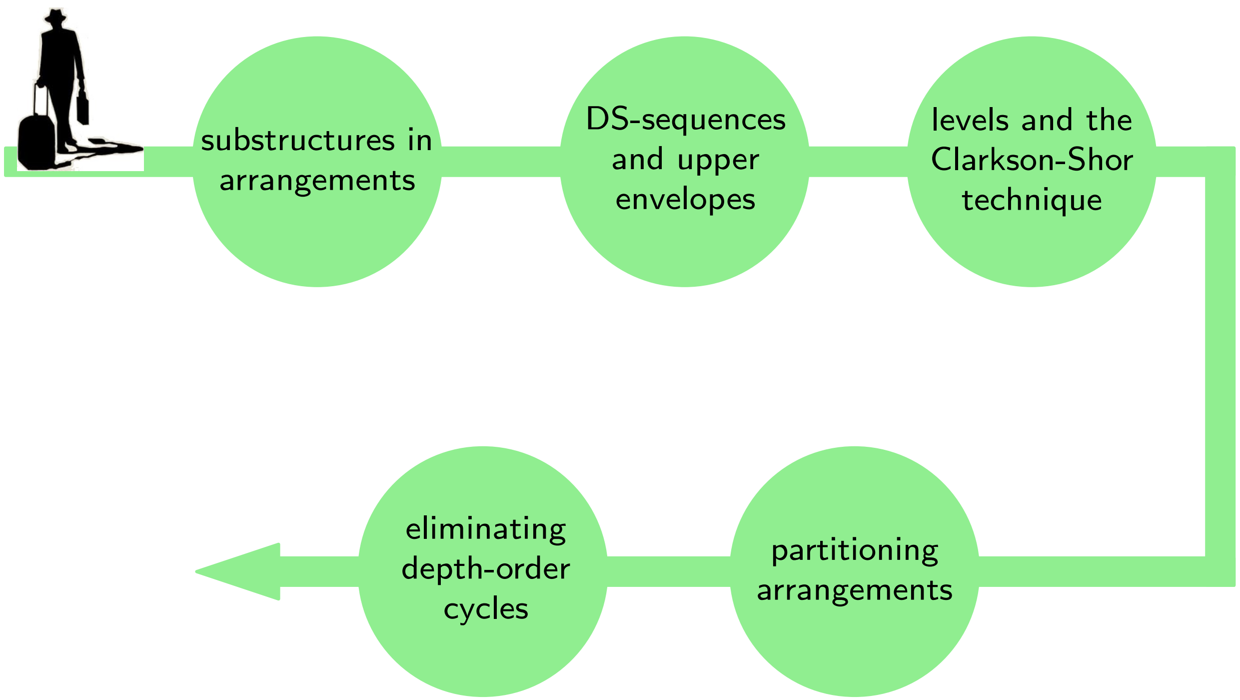
Fundamental Techniques in Computational Geometry:

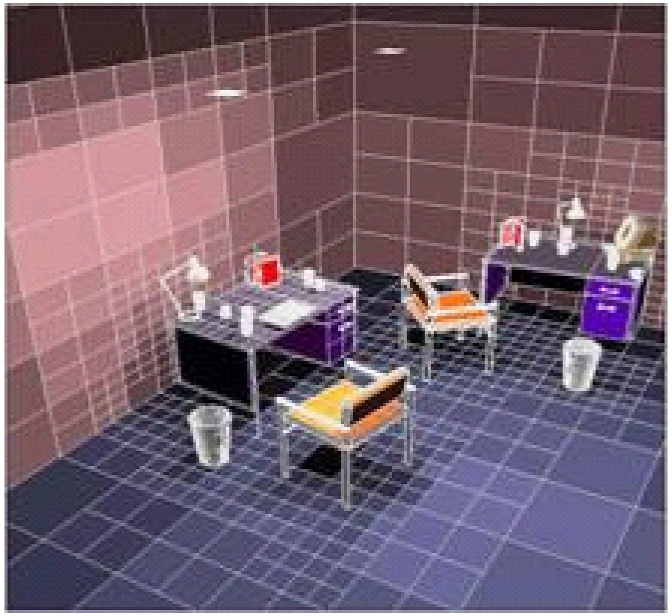
ARRANGEMENTS, PARTITIONS, AND APPLICATIONS

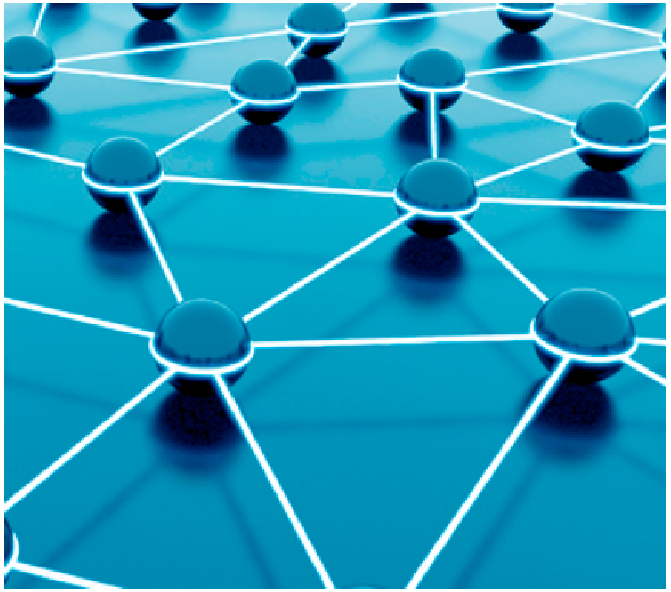


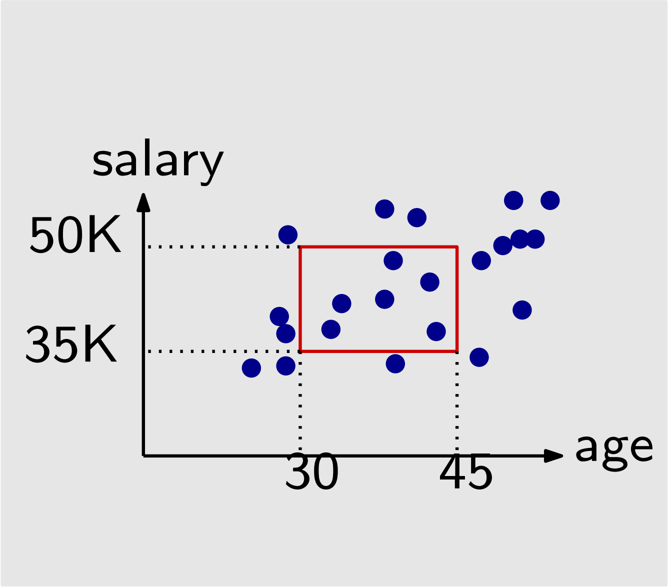
Mark de Berg (TU Eindhoven)

Geometry is everywhere . . .

- geographic information systems
- computer-aided design and manufacturing
- virtual reality
- robotics
- computational biology
- sensor networks
- databases
- and more . . .







Computational Geometry

area within algorithms research dealing with spatial data

- aim for provably correct solutions (no heuristics)
- theoretical analysis of running time, memory usage: $O(\cdots)$

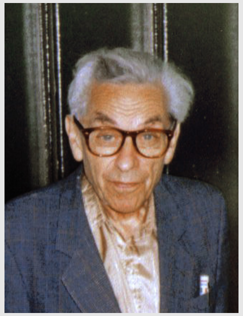
Computational Geometry

area within algorithms research dealing with spatial data

- aim for provably correct solutions (no heuristics)
- theoretical analysis of running time, memory usage: $O(\cdots)$

beautiful connections to discrete geometry

- combinatorial bounds needed to analyze algorithms
- computational-geometry techniques useful for combinatorial problems



Paul Erdős

Computational Geometry: Tools of the Trade

Algorithmic design techniques and tools

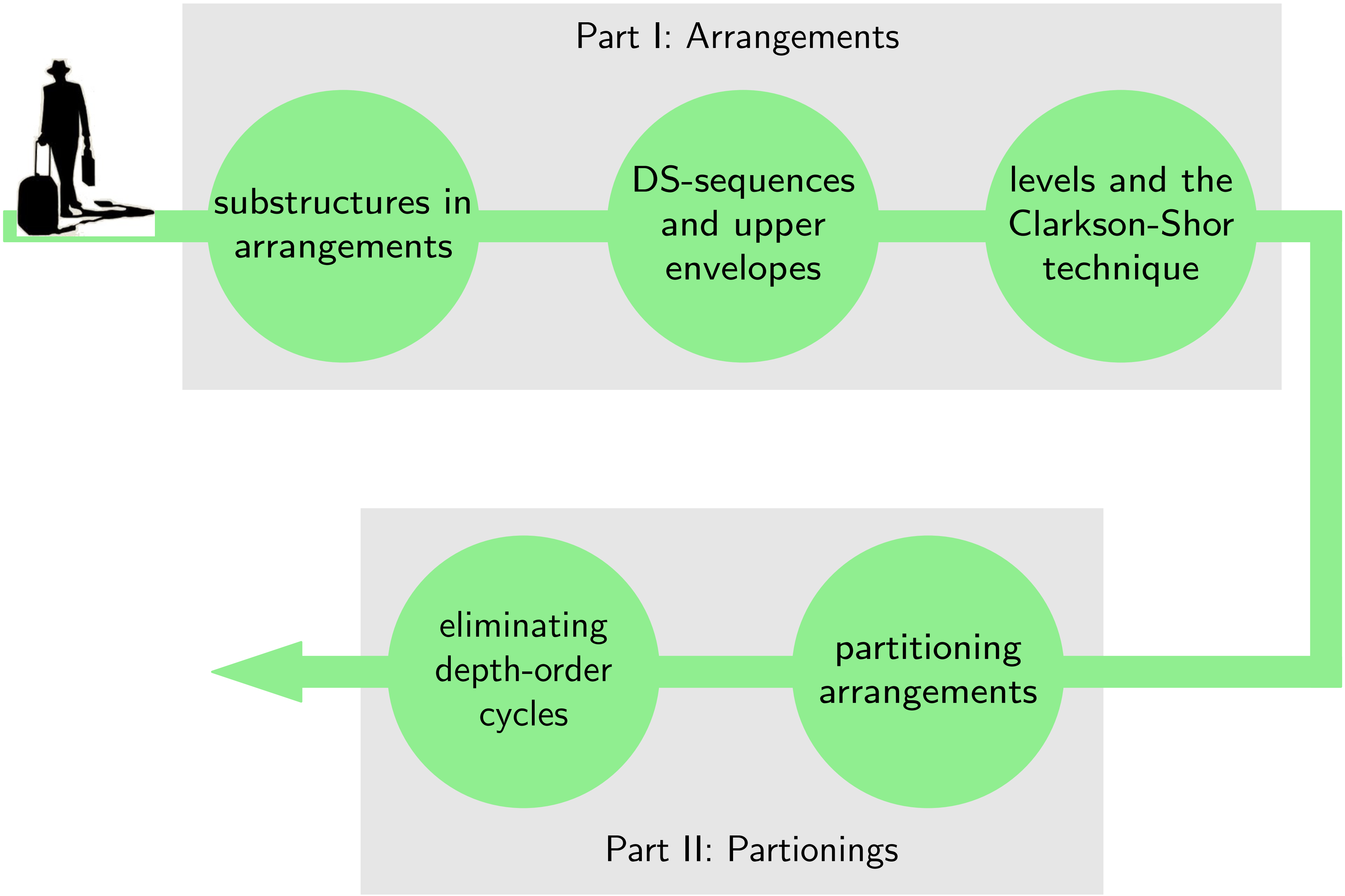
- plane sweep
- geometric divide-and-conquer
- randomized incremental construction
- parametric search
- (multi-level) geometric data structures
- ...

Geometric structures and concepts

- Voronoi diagrams and Delaunay triangulations
- arrangements
- cuttings, simplicial partitions, polynomial partitions
- coresets
- ...

today

Lecture Overview



substructures in arrangements

DS-sequences and upper envelopes

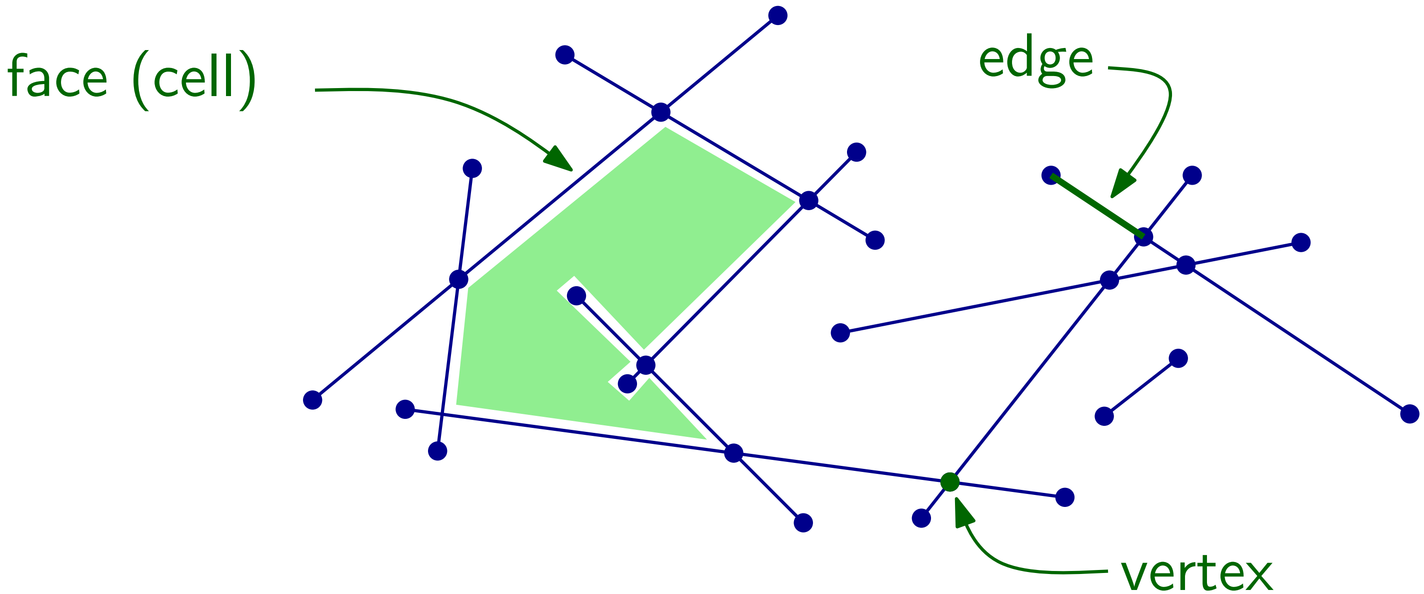
levels and the Clarkson-Shor technique

partitioning arrangements

eliminating depth-order cycles

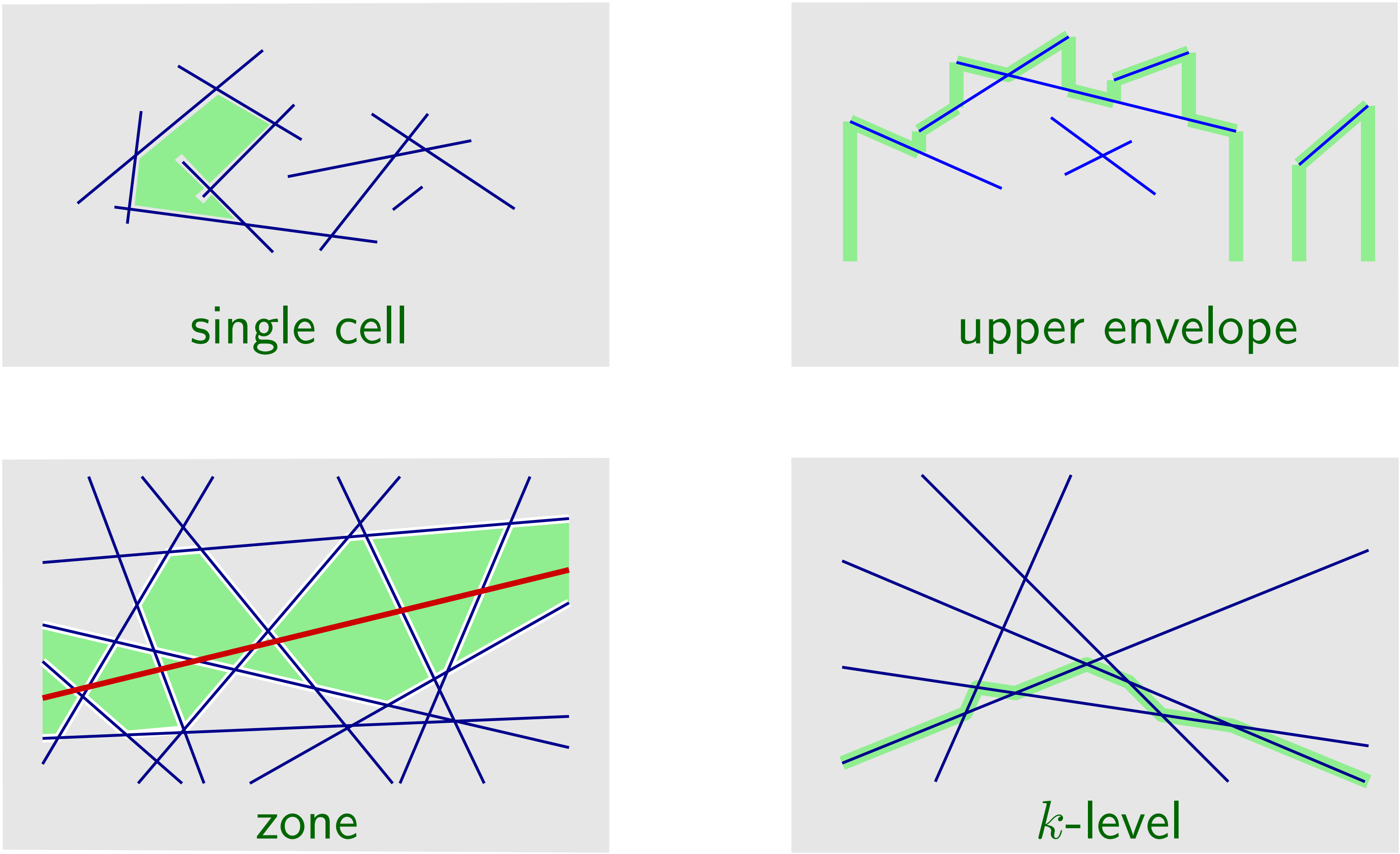
S: set of n lines / segments / curves / etc in $\mathbb{R}^2$

$\mathcal{A}(S)$ = arrangement induced by S
= partitioning of $\mathbb{R}^2$ into faces, edges, and vertices induced by S

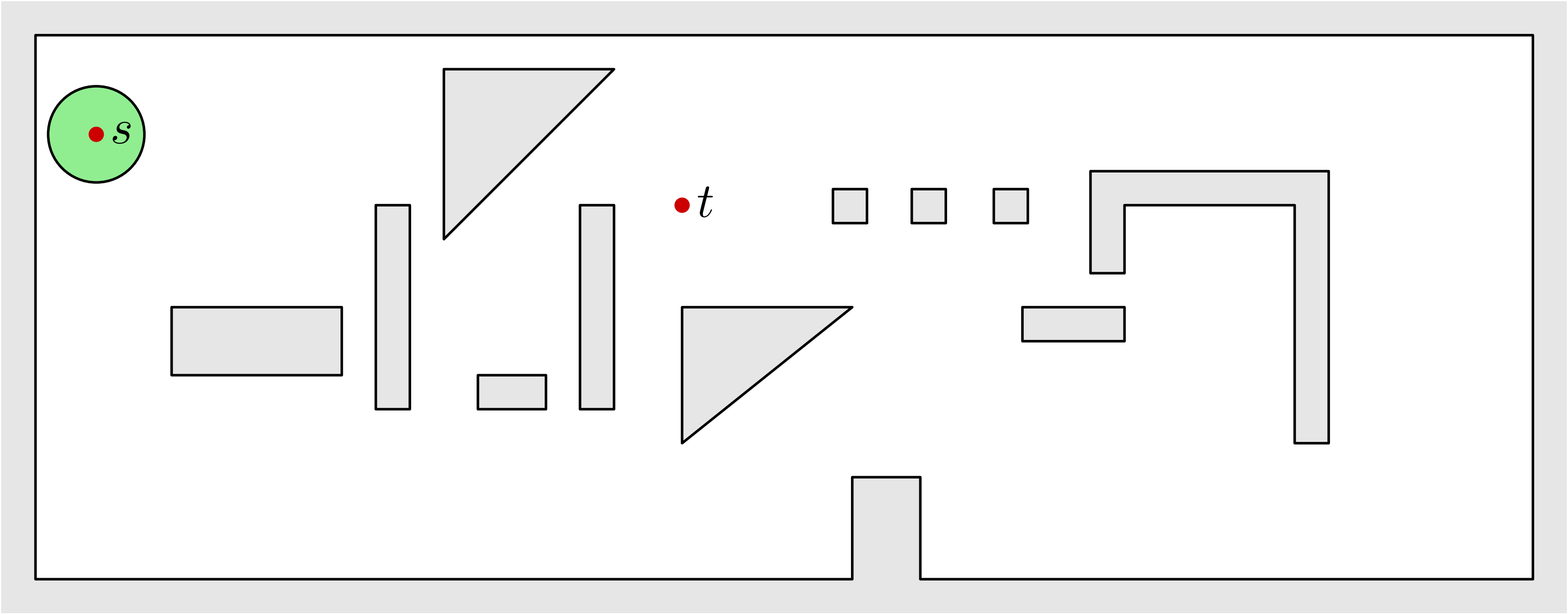


combinatorial complexity of $\mathcal{A}(S)$ = total number of vertices, edges, faces

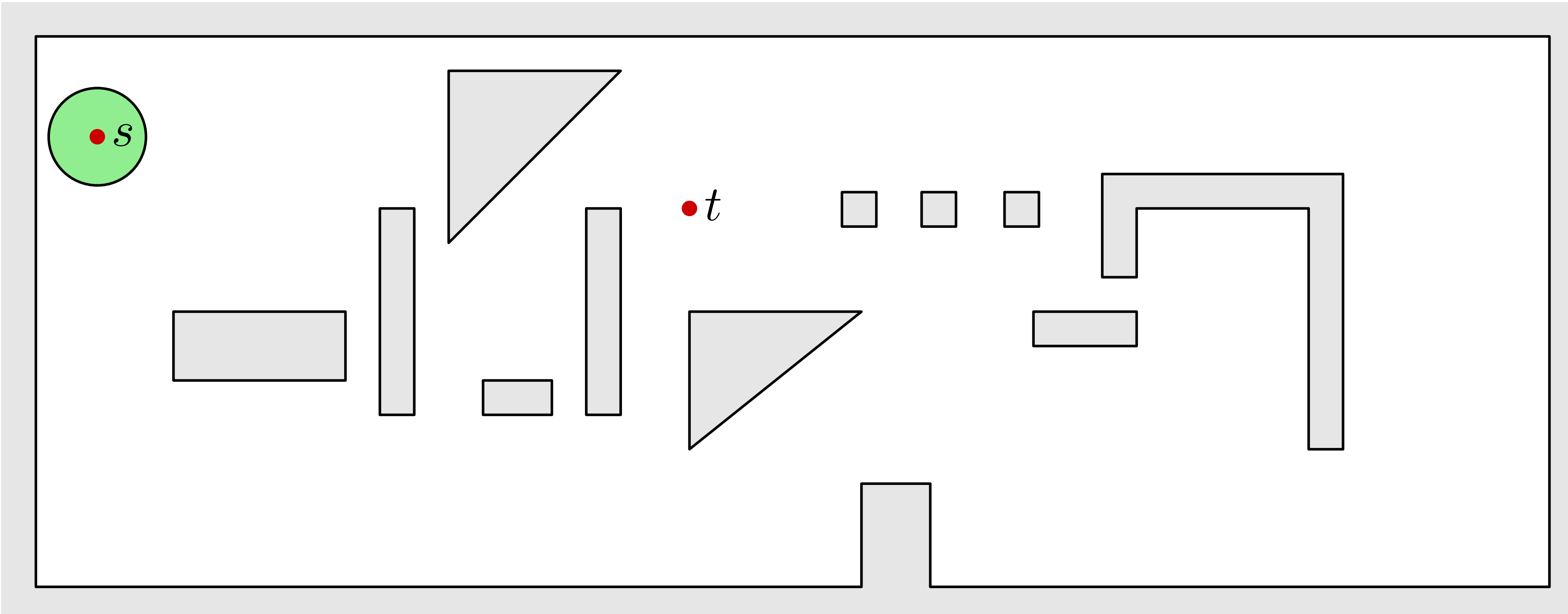
Substructures in Arrangements



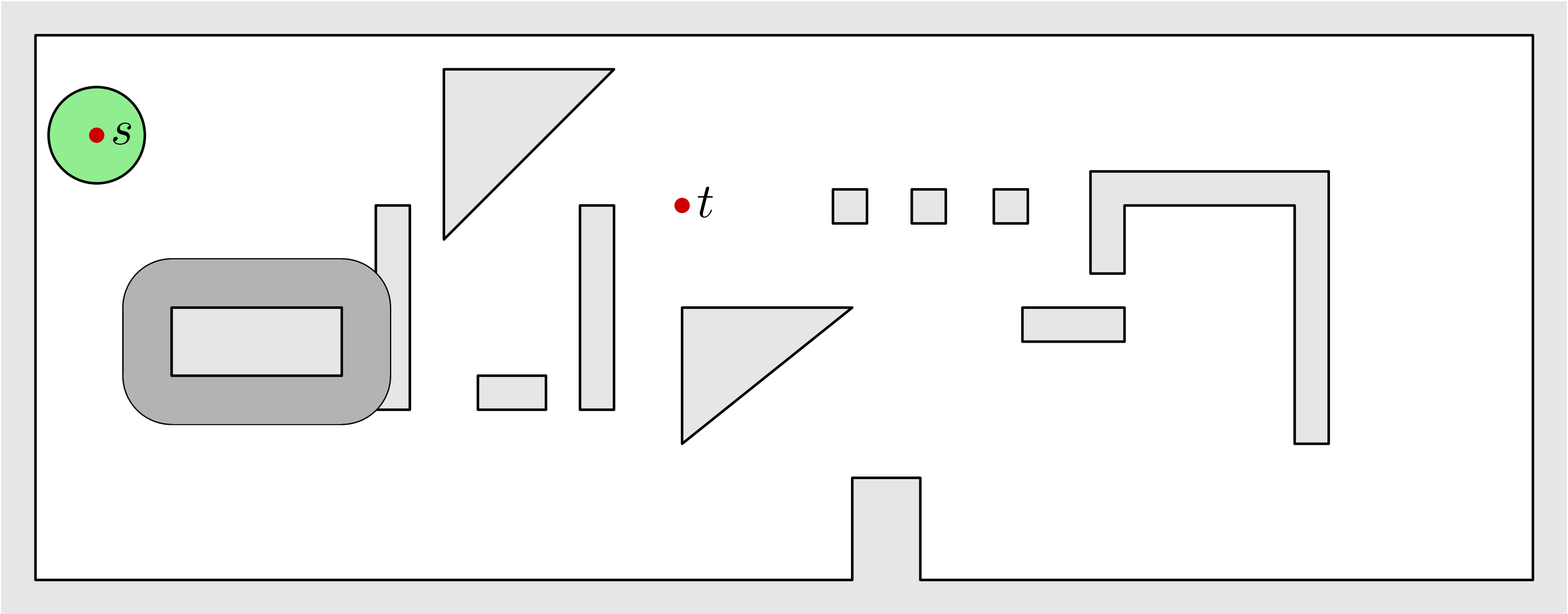
Many geometric problems can be phrased in terms of (substructures in) arrangements by viewing them in an appropriate parametric space.



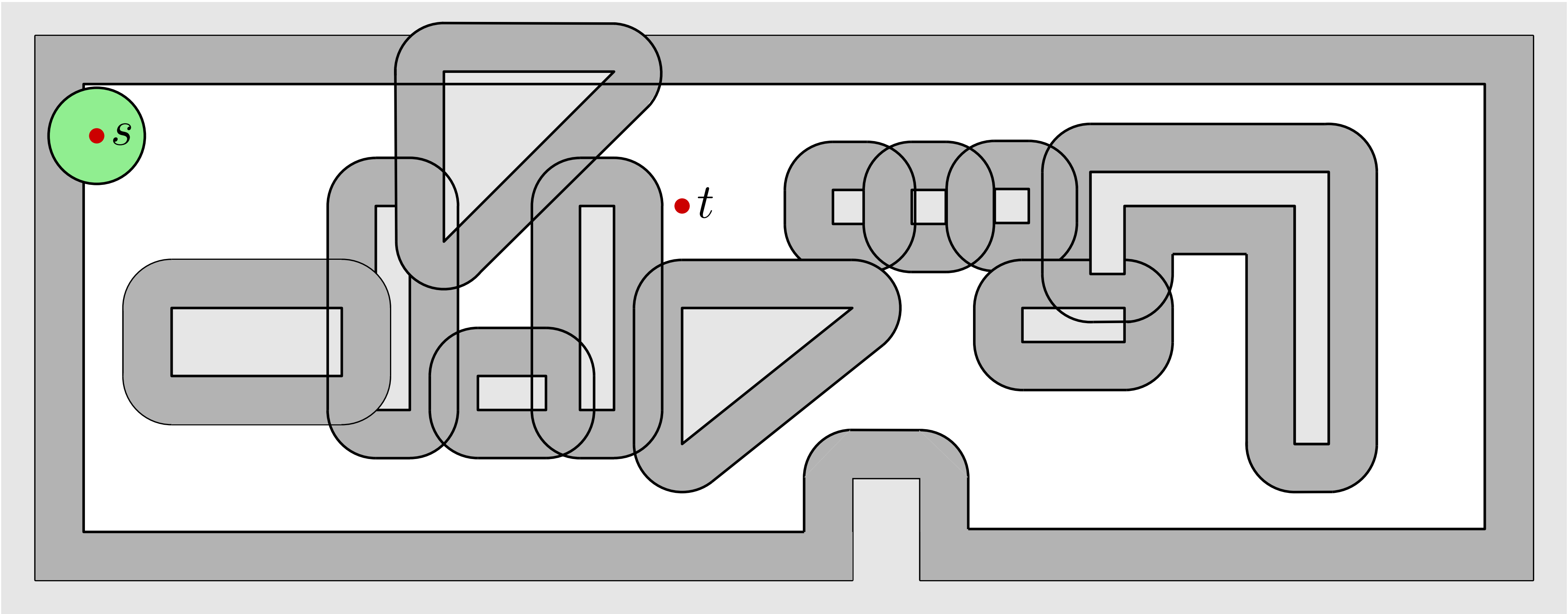
s
t



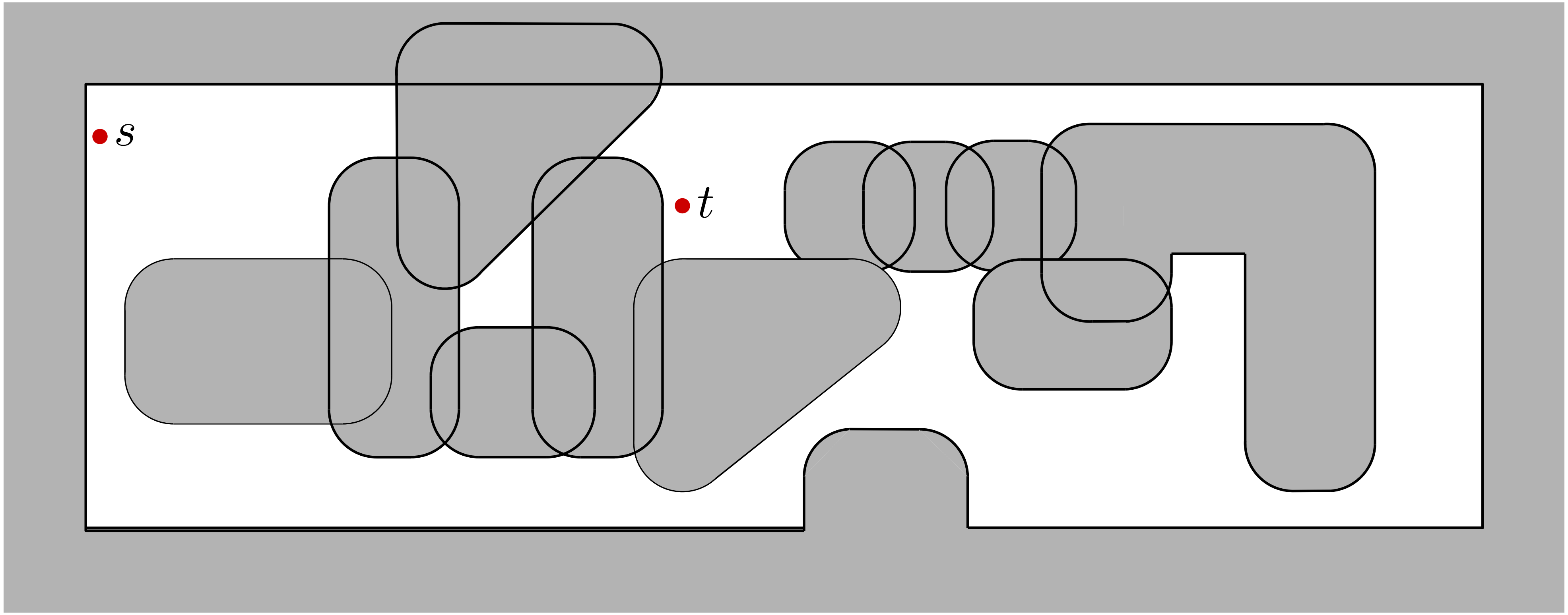
1. Transform problem to motion-planning problem for a point-shaped robot



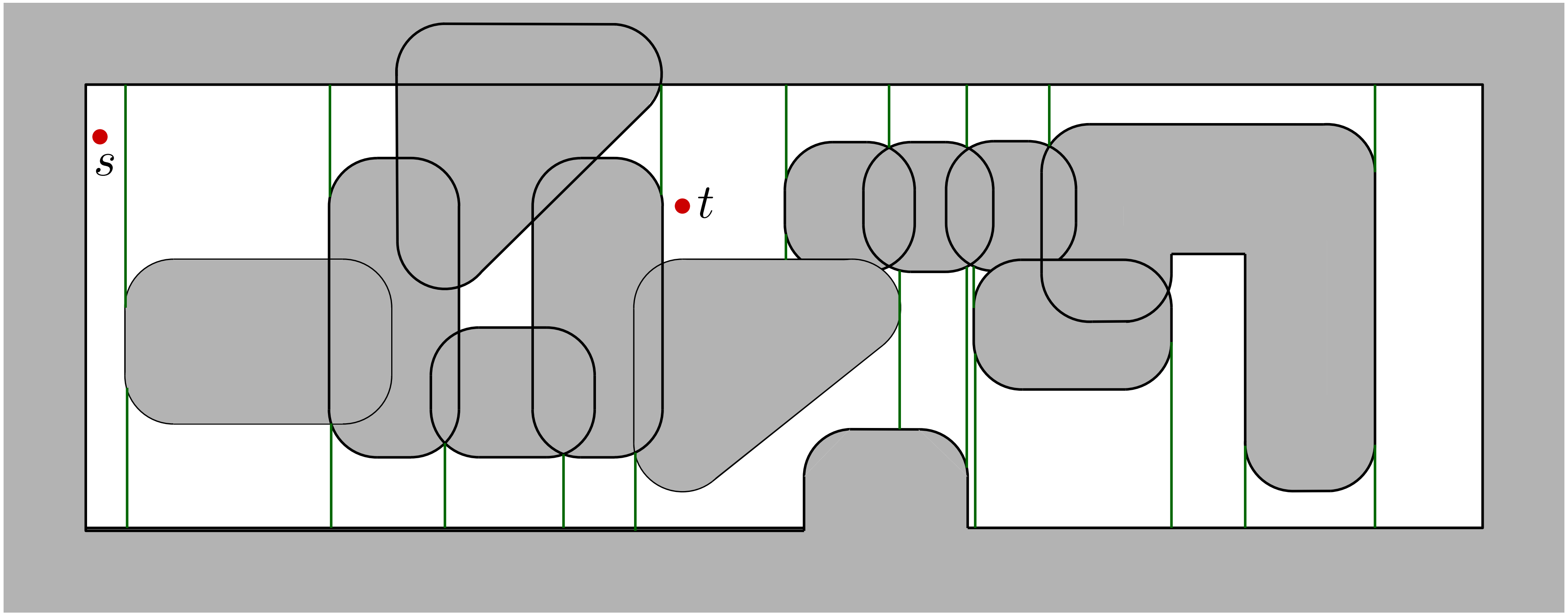
1. Transform problem to motion-planning problem for a point-shaped robot



1. Transform problem to motion-planning problem for a point-shaped robot

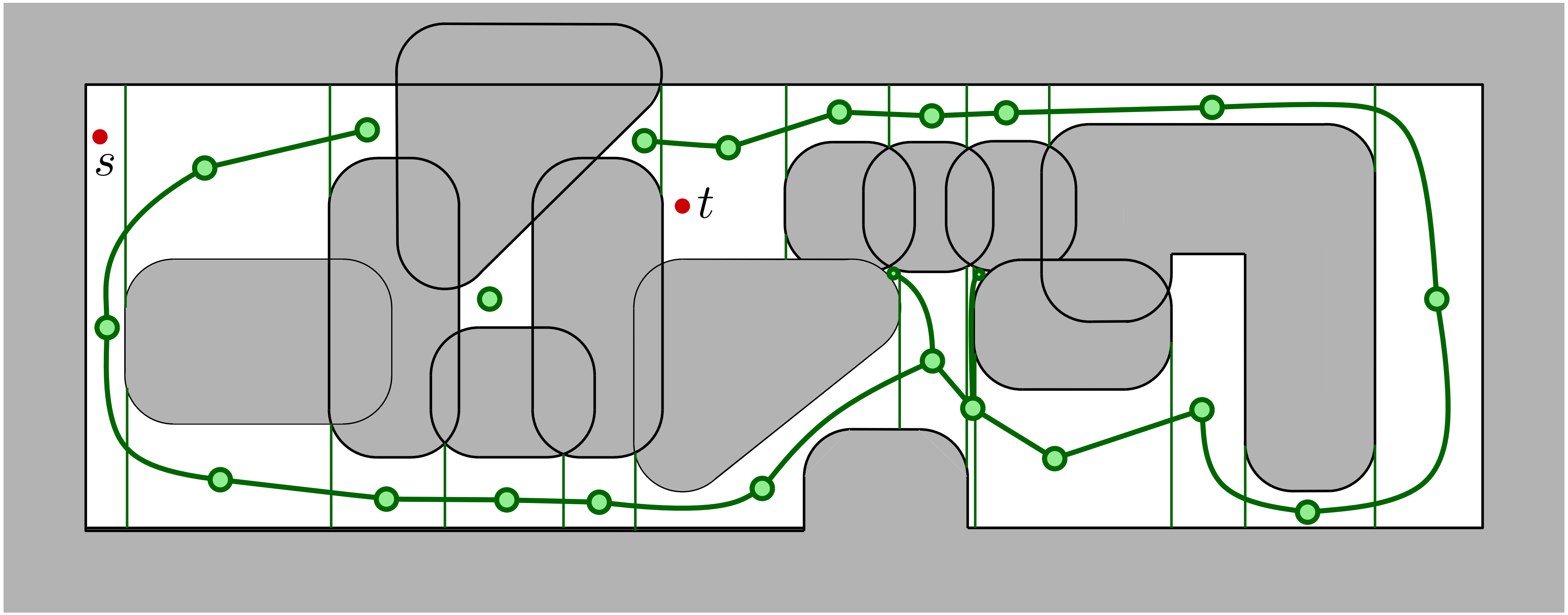


1. Transform problem to motion-planning problem for a point-shaped robot by expanding each obstacle. (Expanded obstacles can intersect!)

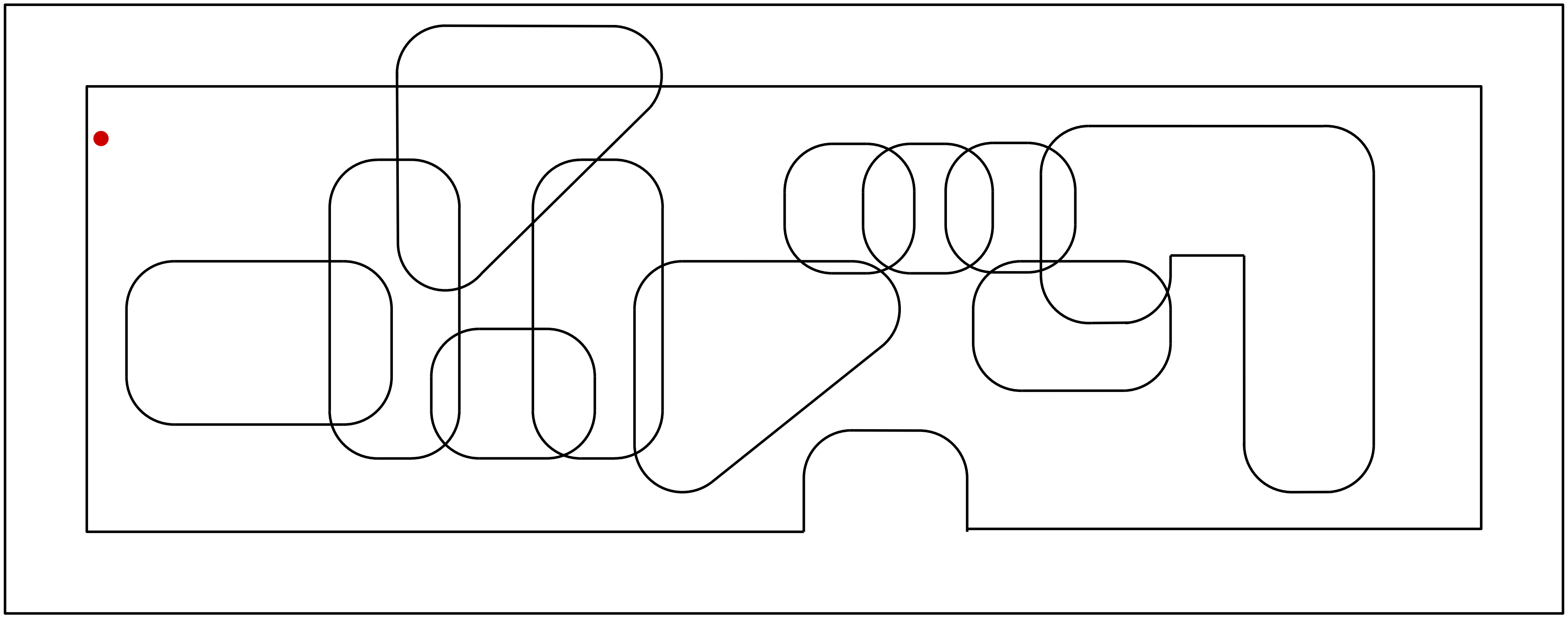


1. Transform problem to motion-planning problem for a point-shaped robot by expanding each obstacle. (Expanded obstacles can intersect!)
2. Decompose free space into "quadrilaterals"

Robot Motion Planning



1. Transform problem to motion-planning problem for a point-shaped robot by expanding each obstacle. (Expanded obstacles can intersect!)
2. Decompose free space into "quadrilaterals"
3. Construct motion graph $\mathcal{G}$ and compute path from s to t in $\mathcal{G}$



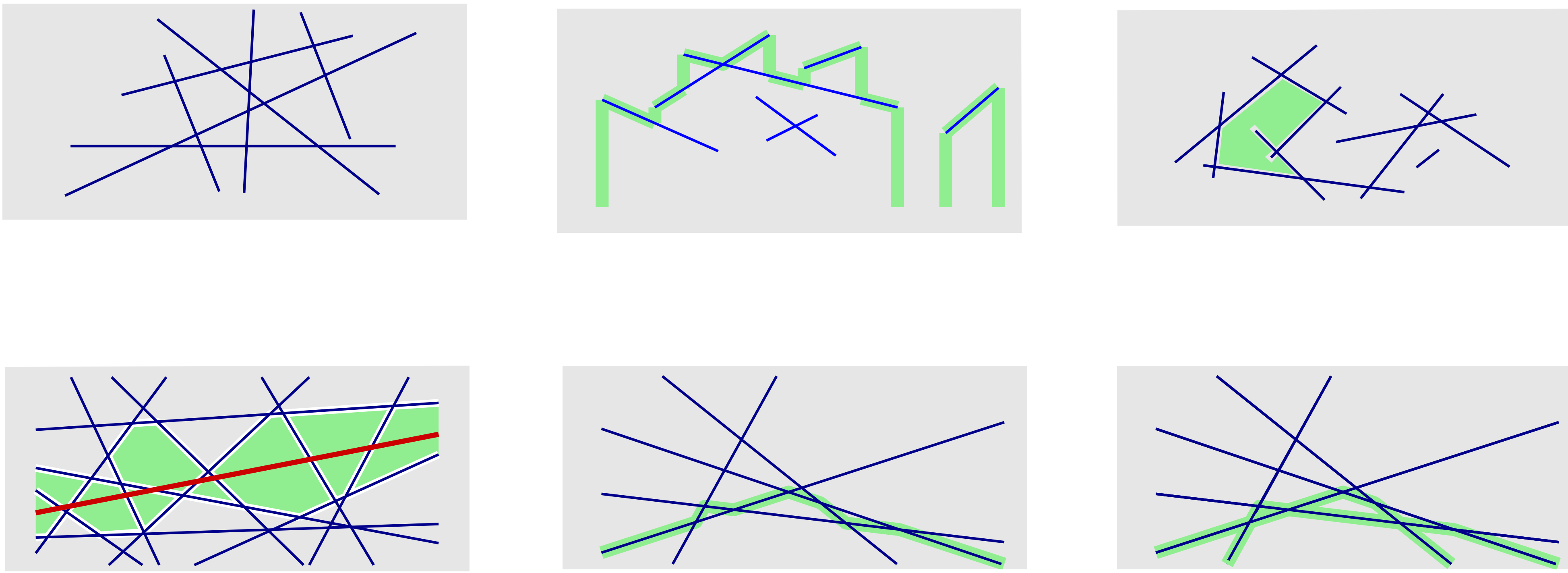
reachable region of the robot

$=$

single cell in arrangement induced by a set S of n curves in $\mathbb{R}^2$

for other types of robots: in $\mathbb{R}^d$, where $d = \#$(degrees of freedom)

The Complexity of (Substructures in) Arrangements



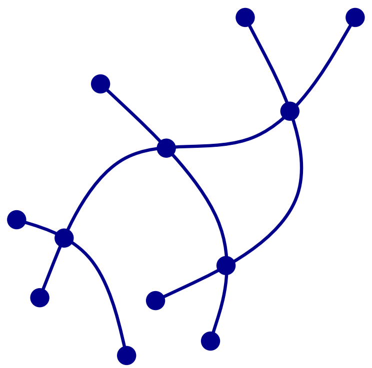
$\mathcal{A}(S)$:= arrangement of set S of n line/segments/curves in $\mathbb{R}^2$

(or: hyperplanes/$(d-1)$-simplices/surface patches in $\mathbb{R}^d$)

what is the worst-case complexity of these substructures in$\mathcal{A}(S)$?

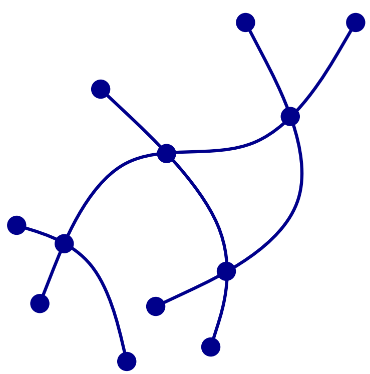
The Complexity of Arrangements

Theorem. Let S be a set of n simple curves such that any two curves intersect at most s times, where S is a fixed constant. Then the complexity of the full arrangement $\mathcal{A}(S)$ is $O(n^2)$.



Theorem. Let S be a set of n simple curves such that any two curves intersect at most s times, where S is a fixed constant. Then the complexity of the full arrangement $\mathcal{A}(S)$ is $O(n^2)$.

Proof.

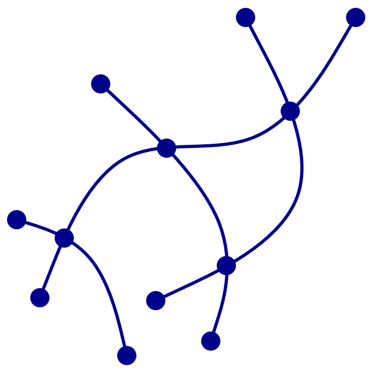


Assume curves are finite.

- number of vertices
- number of edges
- number of faces

Theorem. Let S be a set of n simple curves such that any two curves intersect at most s times, where S is a fixed constant. Then the complexity of the full arrangement $\mathcal{A}(S)$ is $O(n^2)$.

Proof.

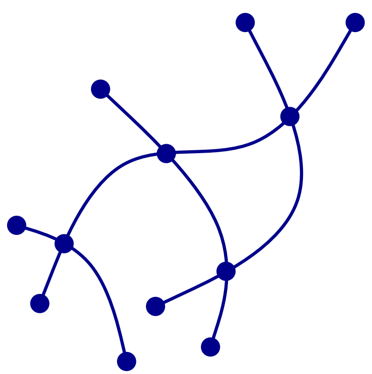


Assume curves are finite.

- number of vertices $\quad |V| \leqslant 2n + s \cdot \binom{n}{2} = O(n^2)$
- number of edges
- number of faces

Theorem. Let S be a set of n simple curves such that any two curves intersect at most s times, where S is a fixed constant. Then the complexity of the full arrangement $\mathcal{A}(S)$ is $O(n^2)$.

Proof.



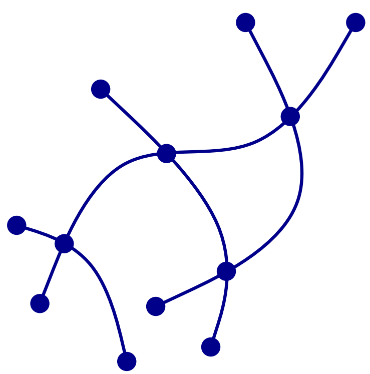
Assume curves are finite.

- number of vertices $|V| \leqslant 2n + s \cdot \binom{n}{2} = O(n^2)$
- number of edges $|E| \leqslant n \cdot (s(n-1)+1) = O(n^2)$
- number of faces

The Complexity of Arrangements

Theorem. Let S be a set of n simple curves such that any two curves intersect at most s times, where S is a fixed constant. Then the complexity of the full arrangement $\mathcal{A}(S)$ is $O(n^2)$.

Proof.



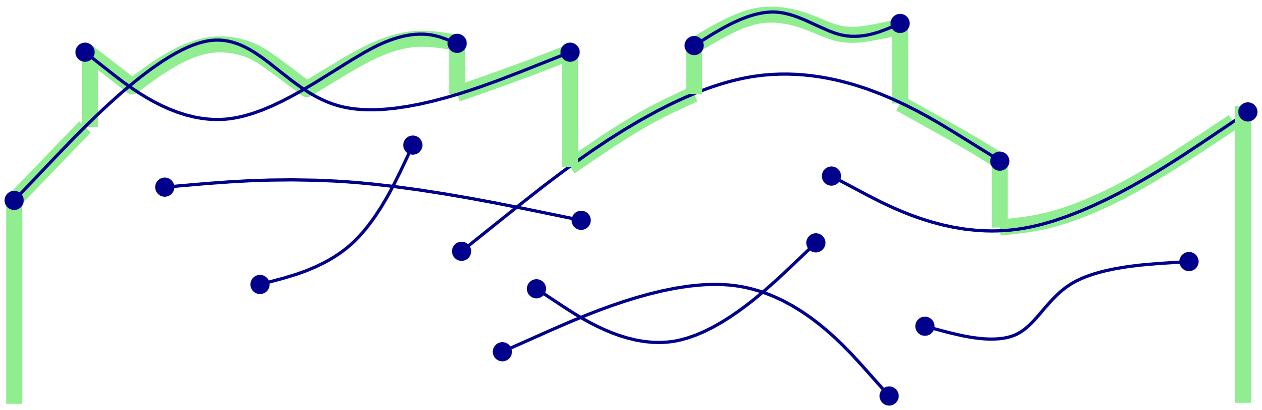
Assume curves are finite.

- number of vertices $|V| \leqslant 2n + s \cdot \binom{n}{2} = O(n^2)$
- number of edges $|E| \leqslant n \cdot (s(n-1)+1) = O(n^2)$
- number of faces

Euler's formula:

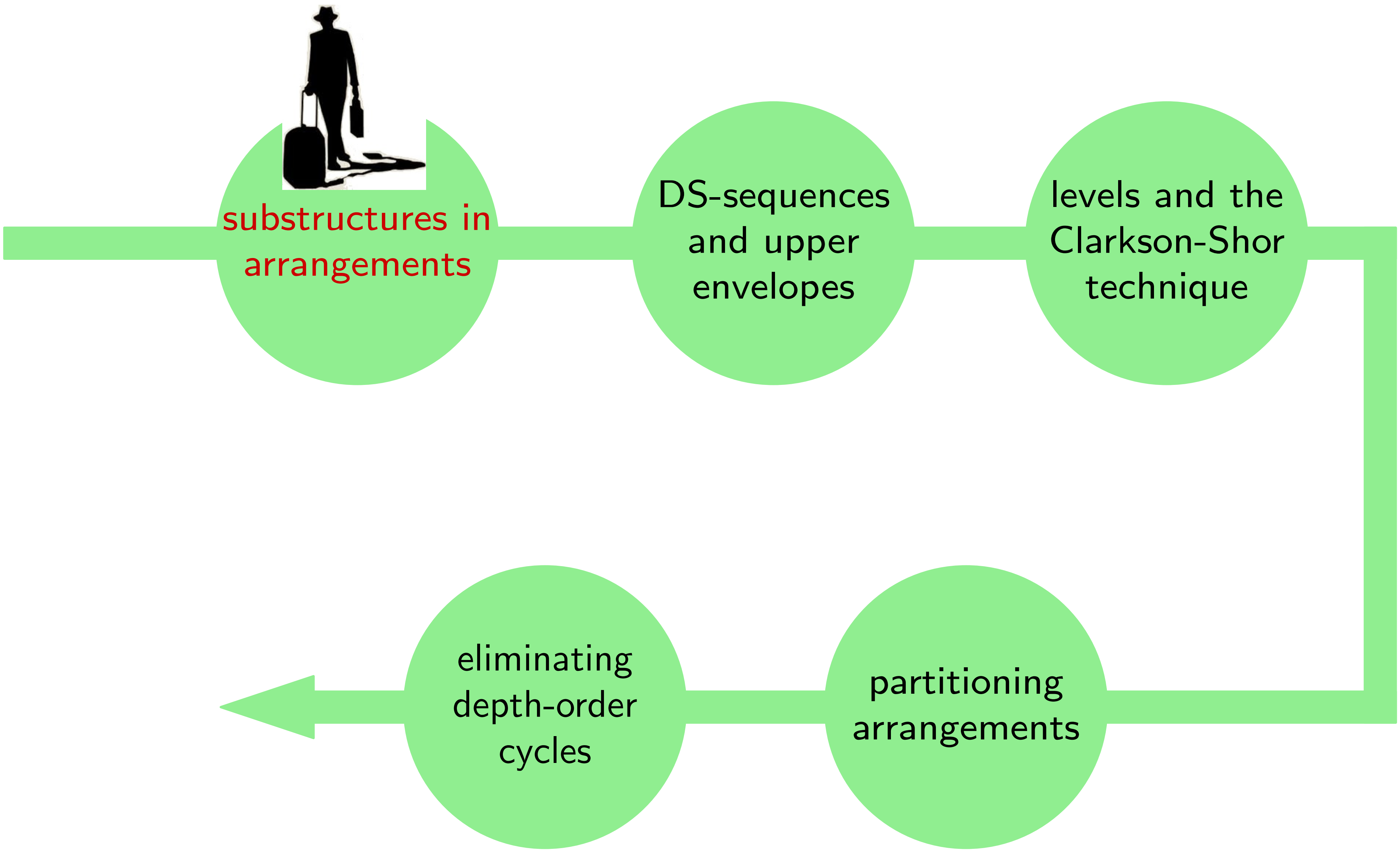
$$|V| - |E| + |F| = 2$$

∎

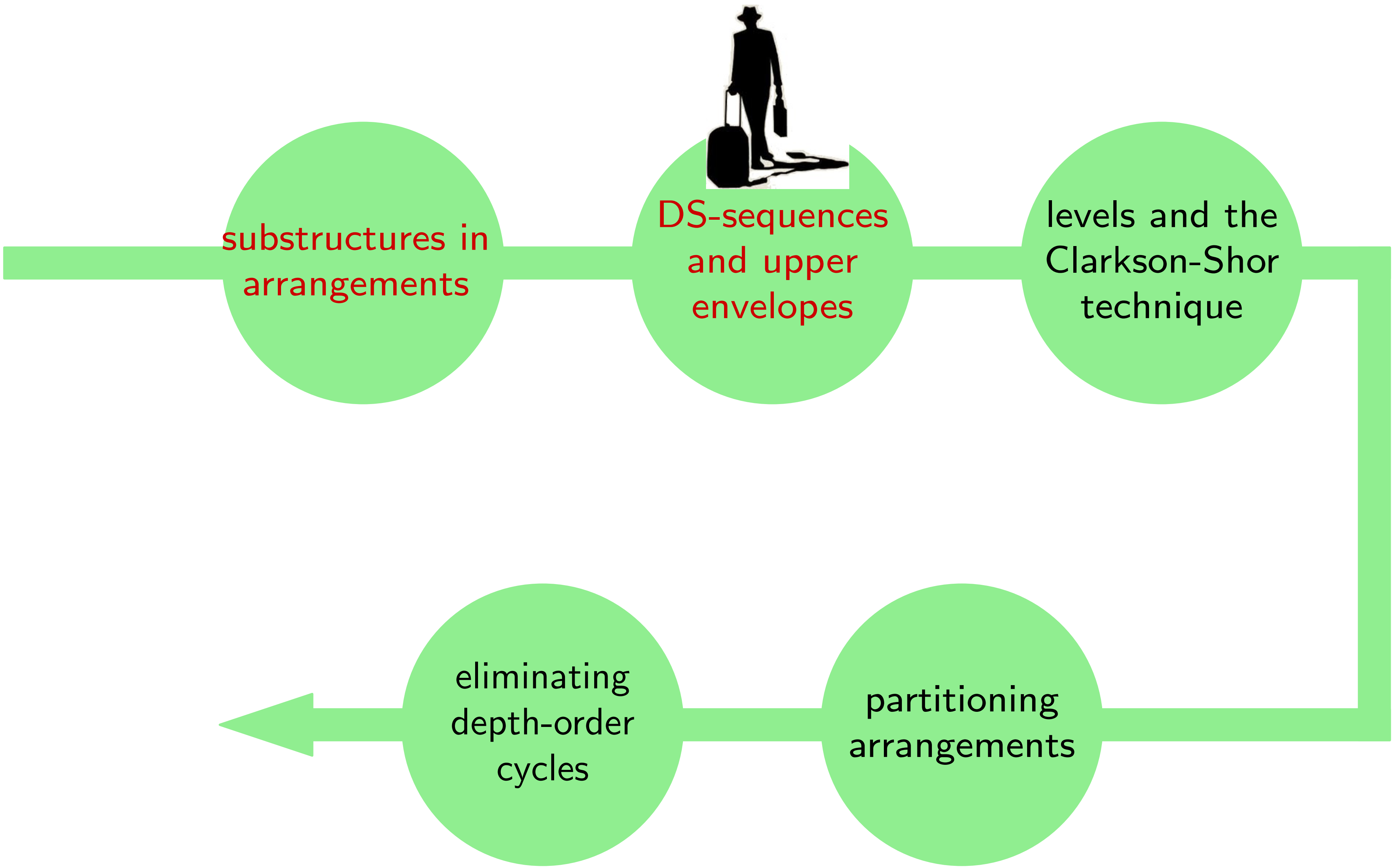


analysis using Davenport-Schinzel sequences

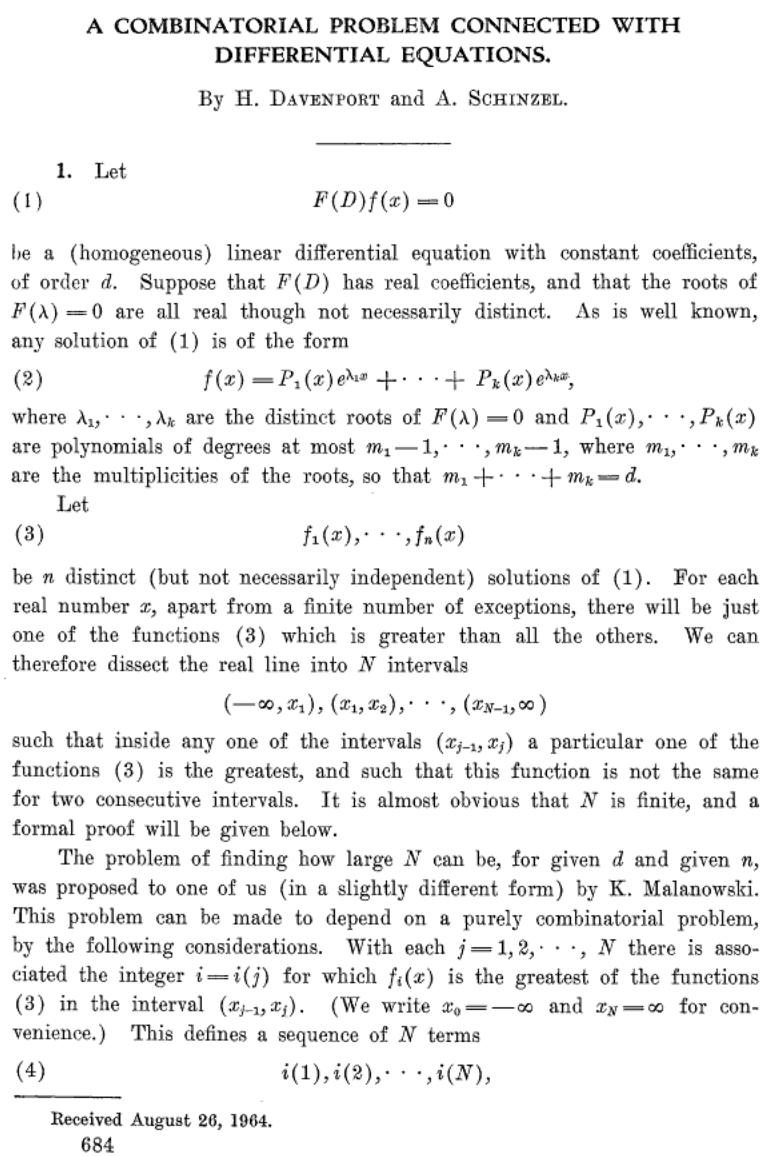
Lecture Overview



Lecture Overview



Davenport-Schinzel sequences



A COMBINATORIAL PROBLEM CONNECTED WITH DIFFERENTIAL EQUATIONS.

By H. Davenport and A. Schinzel.

1. Let

(1) $$F(D)f(x)=0$$

be a (homogeneous) linear differential equation with constant coefficients, of order d. Suppose that $F(D)$ has real coefficients, and that the roots of $F(\lambda)=0$ are all real though not necessarily distinct. As is well known, any solution of (1) is of the form

(2) $$f(x)=P_1(x)e^{\lambda_1 x}+\cdots+P_k(x)e^{\lambda_k x},$$

where $\lambda_1,\cdots,\lambda_k$ are the distinct roots of $F(\lambda)=0$ and $P_1(x),\cdots,P_k(x)$ are polynomials of degrees at most $m_1-1,\cdots,m_k-1$, where $m_1,\cdots,m_k$ are the multiplicities of the roots, so that $m_1+\cdots+m_k=d$.

Let

(3) $$f_1(x),\cdots,f_n(x)$$

be n distinct (but not necessarily independent) solutions of (1). For each real number x, apart from a finite number of exceptions, there will be just one of the functions (3) which is greater than all the others. We can therefore dissect the real line into N intervals

$$(-\infty,x_1),(x_1,x_2),\cdots,(x_{N-1},\infty)$$

such that inside any one of the intervals (x_{j-1},x_j) a particular one of the functions (3) is the greatest, and such that this function is not the same for two consecutive intervals. It is almost obvious that N is finite, and a formal proof will be given below.

The problem of finding how large N can be, for given d and given n, was proposed to one of us (in a slightly different form) by K. Malanowski. This problem can be made to depend on a purely combinatorial problem, by the following considerations. With each $j=1,2,\cdots,N$ there is associated the integer $i=i(j)$ for which $f_i(x)$ is the greatest of the functions (3) in the interval (x_{j-1},x_j). (We write $x_0=-\infty$ and $x_N=\infty$ for convenience.) This defines a sequence of N terms

(4) $$i(1),i(2),\cdots,i(N),$$

Received August 26, 1964.

American Journal of Mathematics 87:684–694 (1965)

Harold Davenport (1907–1965)

Andrzej Schinzel (1937–2021)

A combinatorial problem

Consider a sequence over the alphabet $\{1,\ldots,n\}$ such that

- $\ldots\ i\ i\ \ldots$ does not appear
- $\underbrace{\ldots i \ldots j \ldots i \ldots j \ldots}_{s+2 \text{ times}}$ does not appear

How long can such a sequence be?

Davenport-Schinzel sequences

Davenport-Schinzel sequence of order s (over alphabet of size n) is sequence that does not contain the following:

- $\ldots\ i\ i\ \ldots$ no two consecutive symbols are the same
- $\ldots\underbrace{i\ldots j\ldots i\ldots j}_{s+2\text{ times}}\ldots$ no alternating subsequence of length $s+2$

Example $(n = 9,\ s = 2)$

- 6, 4, 5, 6, 1, 2, 2, 7, 3
- 2, 5, 1, 2, 7, 8, 7, 1, 3, 4
- 3, 6, 4, 2, 5, 1, 5, 9, 8, 9, 7

Davenport-Schinzel sequence of order s (over alphabet of size n) is sequence that does not contain the following:

- $\ldots\ i\ i\ \ldots$ — no two consecutive symbols are the same
- $\ldots\underbrace{i \ldots j \ldots i \ldots j}_{s+2 \text{ times}}\ldots$ — no alternating subsequence of length $s+2$

Example $(n = 9,\ s = 2)$

- 6, 4, 5, 6, 1, 2, 2, 7, 3 ✗
- 2, 5, 1, 2, 7, 8, 7, 1, 3, 4
- 3, 6, 4, 2, 5, 1, 5, 9, 8, 9, 7

Davenport-Schinzel sequences

Davenport-Schinzel sequence of order s (over alphabet of size n) is sequence that does not contain the following:

- $\ldots\ i\ i\ \ldots$ — no two consecutive symbols are the same
- $\ldots\underbrace{i\ldots j\ldots i\ldots j}_{s+2\text{ times}}\ldots$ — no alternating subsequence of length $s+2$

Example $(n = 9,\ s = 2)$

- 6, 4, 5, 6, 1, 2, 2, 7, 3 ✕
- 2, 5, 1, 2, 7, 8, 7, 1, 3, 4 ✕
- 3, 6, 4, 2, 5, 1, 5, 9, 8, 9, 7

Davenport-Schinzel sequence of order s (over alphabet of size n) is sequence that does not contain the following:

- $\ldots\ i\ i\ \ldots$ — no two consecutive symbols are the same
- $\ldots\underbrace{i\ldots j\ldots i\ldots j}_{s+2\text{ times}}\ldots$ — no alternating subsequence of length $s+2$

Example $(n=9,\ s=2)$

- 6, 4, 5, 6, 1, 2, 2, 7, 3 ✗
- 2, 5, 1, 2, 7, 8, 7, 1, 3, 4 ✗
- 3, 6, 4, 2, 5, 1, 5, 9, 8, 9, 7 V

Davenport-Schinzel sequences

Davenport-Schinzel sequence of order s (over alphabet of size n) is sequence that does not contain the following:

- $\ldots\ i\ i\ \ldots$ — no two consecutive symbols are the same
- $\ldots\underbrace{i\ldots j\ldots i\ldots j}_{s+2 \text{ times}}\ldots$ — no alternating subsequence of length $s+2$

Example $(n = 9,\ s = 2)$

- 6, 4, 5, 6, 1, 2, 2, 7, 3 ✗
- 2, 5, 1, 2, 7, 8, 7, 1, 3, 4 ✗
- 3, 6, 4, 2, 5, 1, 5, 9, 8, 9, 7 V

Exercise: Determine the maximal possible length of a DS-sequence of order s as a function of n, for $s = 1$, $s = 2$, $s = 3$, $\ldots$

Davenport-Schinzel sequences

Davenport-Schinzel sequence of order s (over alphabet of size n) is sequence that does not contain the following:

- $\dots\ i\ i\ \dots$ — no two consecutive symbols are the same
- $\dots\underbrace{i \dots j \dots i \dots j}_{s+2 \text{ times}}\dots$ — no alternating subsequence of length $s+2$

Davenport-Schinzel sequences

Davenport-Schinzel sequence of order s (over alphabet of size n) is sequence that does not contain the following:

- $\ldots\ i\ i\ \ldots$ — no two consecutive symbols are the same
- $\ldots\underbrace{i\ldots j\ldots i\ldots j}_{s+2\text{ times}}\ldots$ — no alternating subsequence of length $s+2$

$\mathrm{DS}_s(n)$:= maximum length of DS-sequence of order s on n symbols

- $s = 1$:

- $s = 2$:

Davenport-Schinzel sequence of order s (over alphabet of size n) is sequence that does not contain the following:

- $\ldots\ i\ i\ \ldots$ — no two consecutive symbols are the same
- $\ldots\underbrace{i\ldots j\ldots i\ldots j}_{s+2 \text{ times}}\ldots$ — no alternating subsequence of length $s+2$

$\mathrm{DS}_s(n)$:= maximum length of DS-sequence of order s on n symbols

- $s = 1$: $\left.\begin{array}{l}\text{possible sequence: } 1, 2, 3, \ldots, n \\ \text{no symbol can appear twice}\end{array}\right\} \implies \quad \mathrm{DS}_1(n) = n$

- $s = 2$:

Davenport-Schinzel sequences

Davenport-Schinzel sequence of order s (over alphabet of size n) is sequence that does not contain the following:

- $\ldots\ i\ i\ \ldots$ — no two consecutive symbols are the same
- $\ldots \underbrace{i \ldots j \ldots i \ldots j}_{s+2 \text{ times}} \ldots$ — no alternating subsequence of length $s+2$

$\mathrm{DS}_s(n) :=$ maximum length of DS-sequence of order s on n symbols

- $s = 1$: $\left.\begin{array}{l}\text{possible sequence: } 1, 2, 3, \ldots, n \\ \text{no symbol can appear twice}\end{array}\right\} \Longrightarrow \quad \mathrm{DS}_1(n) = n$

- $s = 2$: possible sequence $1, 2, \ldots, n-1, n, n-1, \ldots, 2, 1$

  $\Longrightarrow \quad \mathrm{DS}_2(n) \geqslant 2n - 1$

  Proof by induction, remove symbol whose first occurrence is last, plus at most one adjacent symbol:

  $\mathrm{DS}_2(n) \leqslant \mathrm{DS}(n-1) + 2 \Longrightarrow \quad \mathrm{DS}_2(n) \leqslant 2n - 1$

Theorem. $\mathrm{DS}_s(n)$ is near-linear for any constant s. In particular,

- $\mathrm{DS}_1(n) = n$
- $\mathrm{DS}_2(n) = 2n - 1$
- $\mathrm{DS}_3(n) = \Theta(n\alpha(n))$
- $\mathrm{DS}_s(n) = o(n \log^* n)$ for any fixed constant s

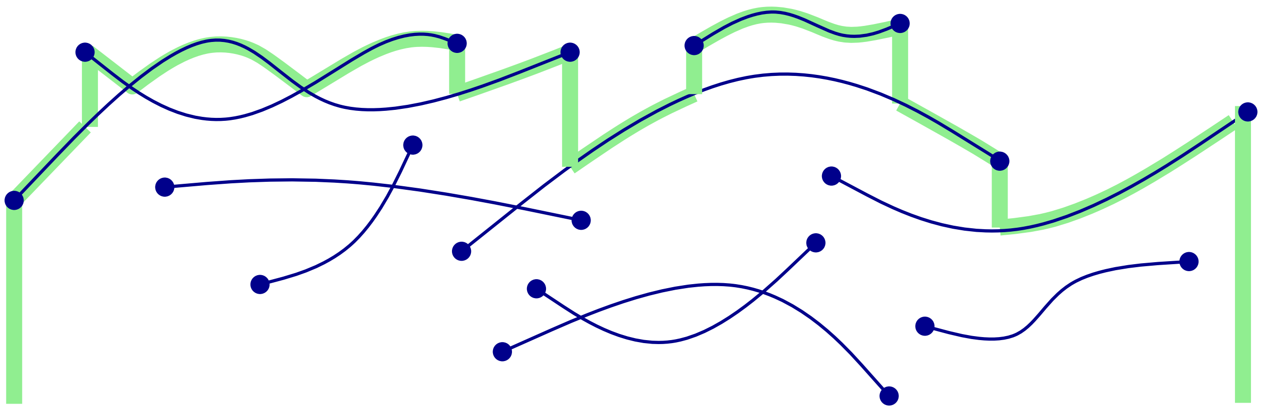
where $\alpha(n)$ is the inverse Ackermann function

$\alpha(n)$ grows slower than super-super-super-super-super-slowly ...

$\alpha(n)$ is inverse of Ackermann function $A(n)$, where $A(n) = A_n(n)$ with:

$$\begin{array}{rcll} A_1(n) & = & 2n & \text{for } n \geqslant 1 \\ A_k(1) & = & 2 & \text{for } k \geqslant 1 \\ A_k(n) & = & A_{k-1}(A_k(n-1)) & \text{for } k \geqslant 2 \text{ and } n \geqslant 2 \end{array}$$

$A(1) = 2$, $A(2) = 4$, $A(3) = 16$, $A(4) =$ tower of 65536 2's

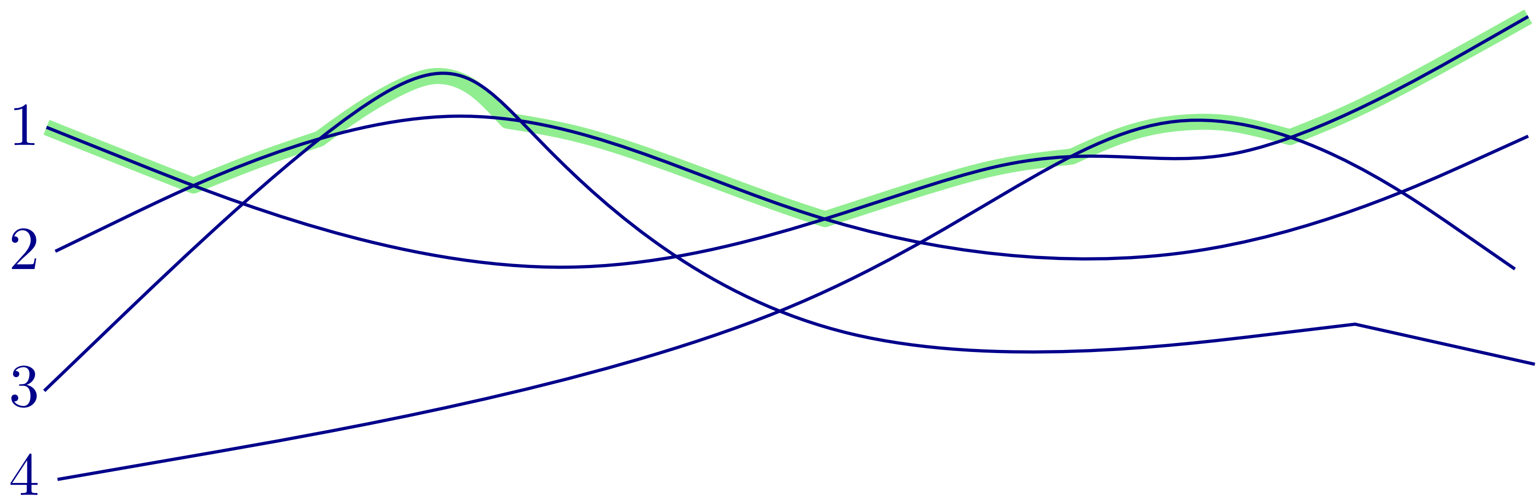


back to upper envelopes

Theorem. Let S be a set of n **infinite** x-monotone curves such that any two curves intersect at most s times. Then the maximum complexity of the upper envelope of S is $O(\mathrm{DS}_s(n))$.

for example: $O(n)$ for lines

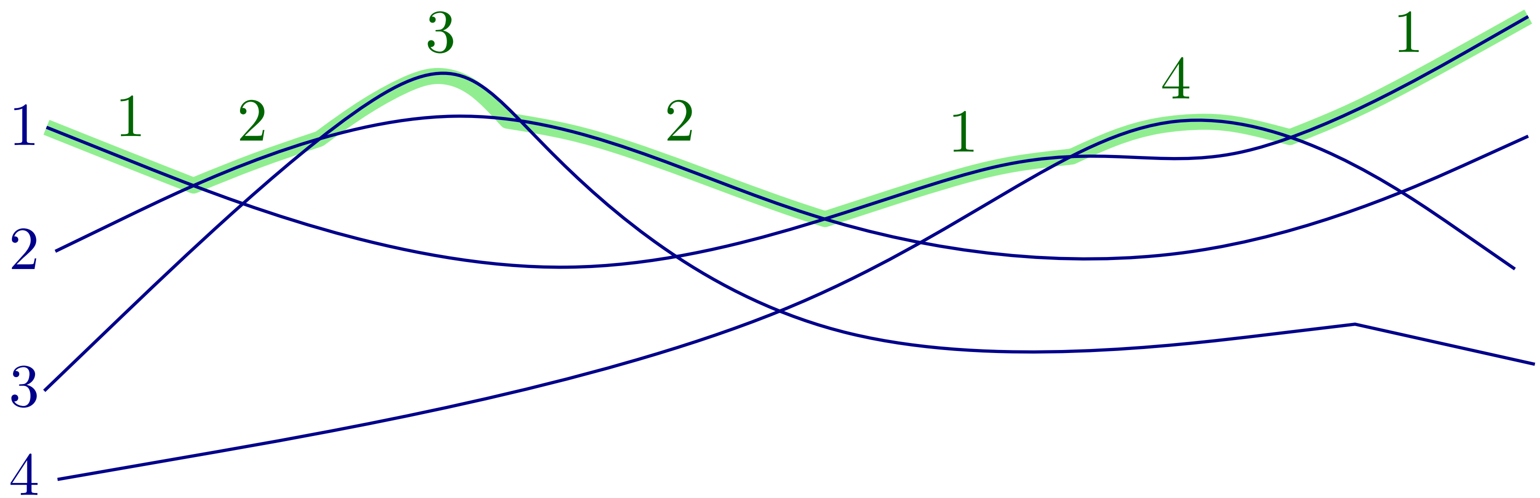
Proof.



Theorem. Let S be a set of n **infinite** x-monotone curves such that any two curves intersect at most s times. Then the maximum complexity of the upper envelope of S is $O(\mathrm{DS}_s(n))$.

for example: $O(n)$ for lines

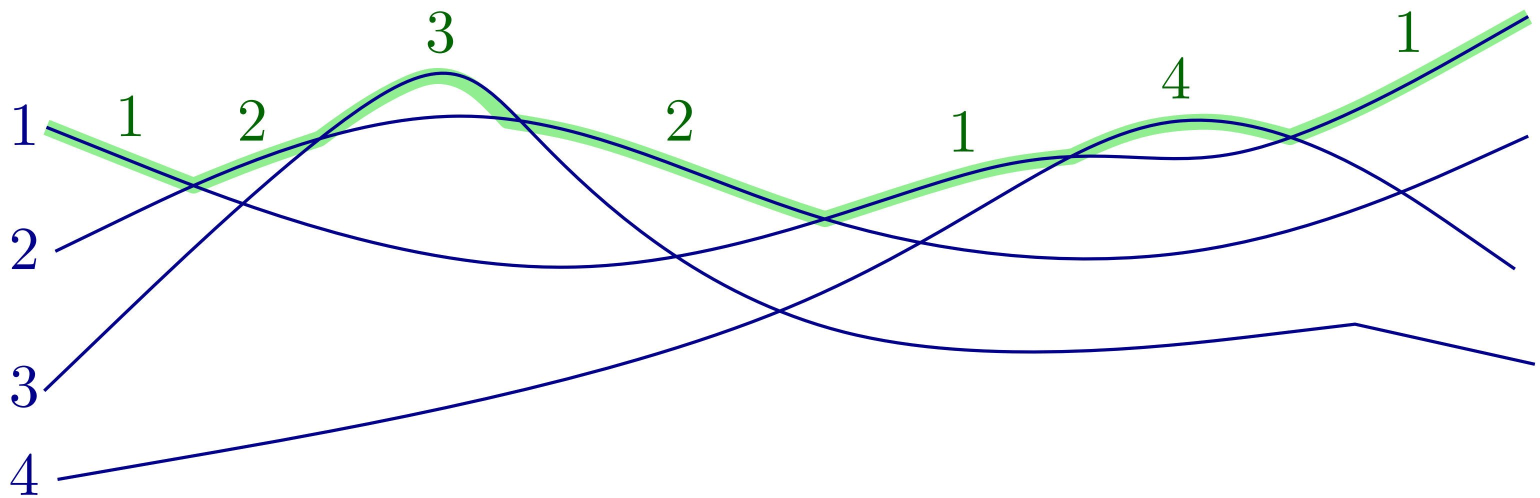
Proof.



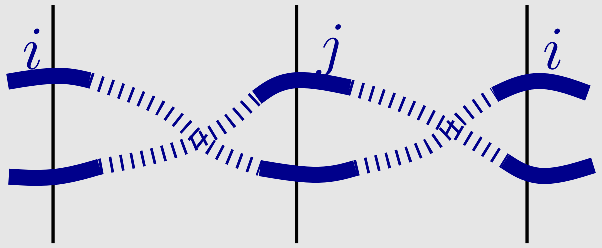
Theorem. Let S be a set of n **infinite** x-monotone curves such that any two curves intersect at most s times. Then the maximum complexity of the upper envelope of S is $O(\mathrm{DS}_s(n))$.

for example: $O(n)$ for lines

Proof.



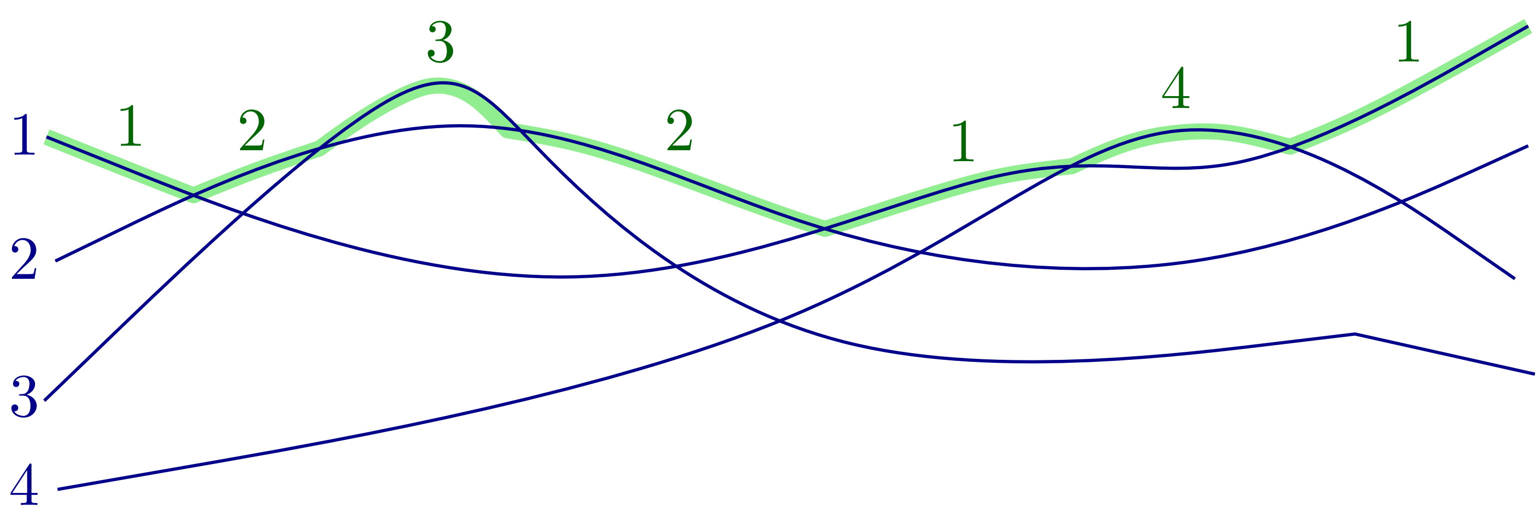
alternating sequence of length t implies $t - 1$ intersections



Theorem. Let S be a set of n **infinite** x-monotone curves such that any two curves intersect at most s times. Then the maximum complexity of the upper envelope of S is $O(\mathrm{DS}_s(n))$.

for example: $O(n)$ for lines

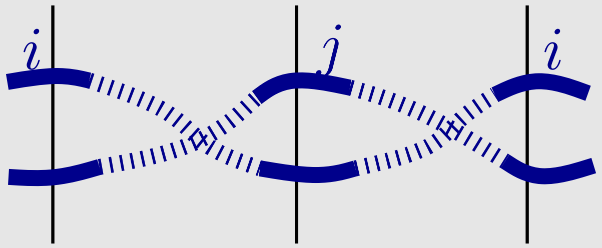
Proof.



we cannot have alternating sequence of length $s+2$

$\Longrightarrow$ DS(n, s)-sequence

alternating sequence of length t implies $t-1$ intersections



■

The Complexity of Upper Envelopes

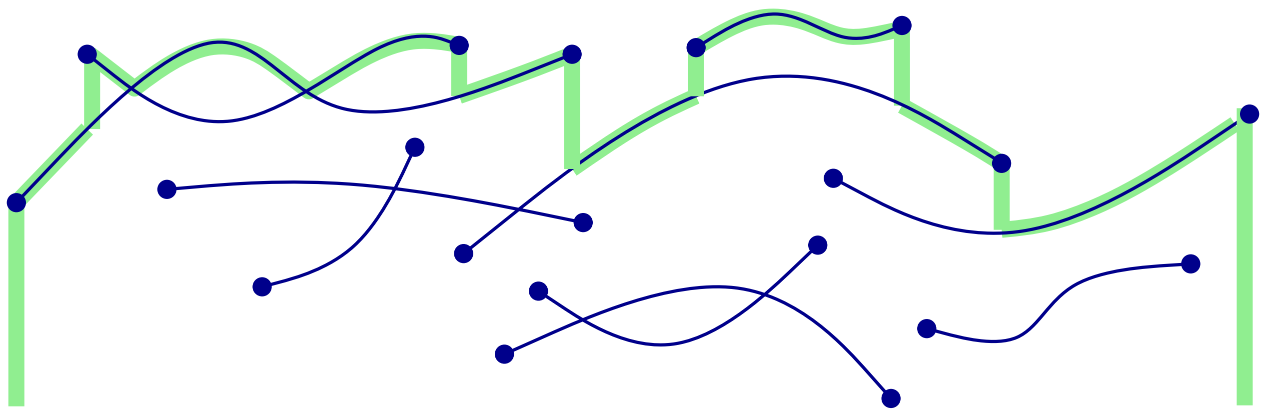
Theorem. Let S be a set of n ~~infinite~~ x-monotone curves such that any two curves intersect at most s times. Then the maximum complexity of the upper envelope of S is $O(\mathrm{DS}_{s+2}(n))$.

for example: $O(n\alpha(n))$ for line segments

Theorem. Let S be a set of n ~~infinite~~ x-monotone curves such that any two curves intersect at most s times. Then the maximum complexity of the upper envelope of S is $O(\mathrm{DS}_{s+2}(n))$.

for example: $O(n\alpha(n))$ for line segments

Proof.

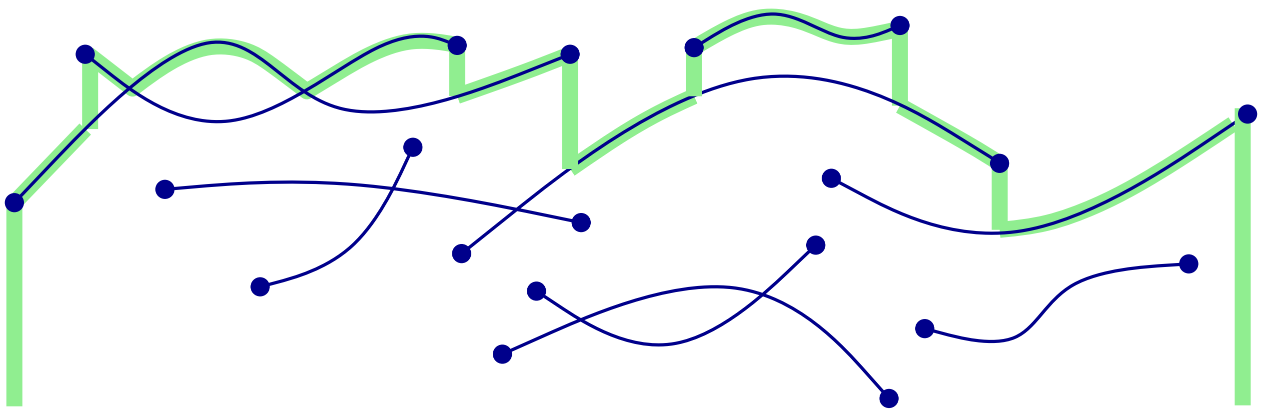


The Complexity of Upper Envelopes

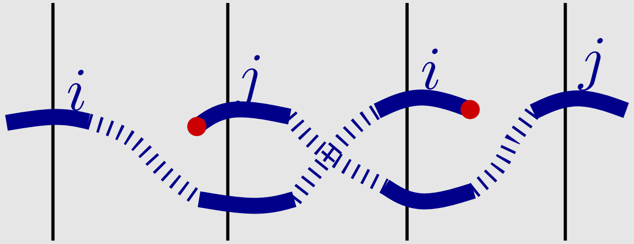
Theorem. Let S be a set of n ~~infinite~~ x-monotone curves such that any two curves intersect at most s times. Then the maximum complexity of the upper envelope of S is $O(\mathrm{DS}_{s+2}(n))$.

for example: $O(n\alpha(n))$ for line segments

Proof.



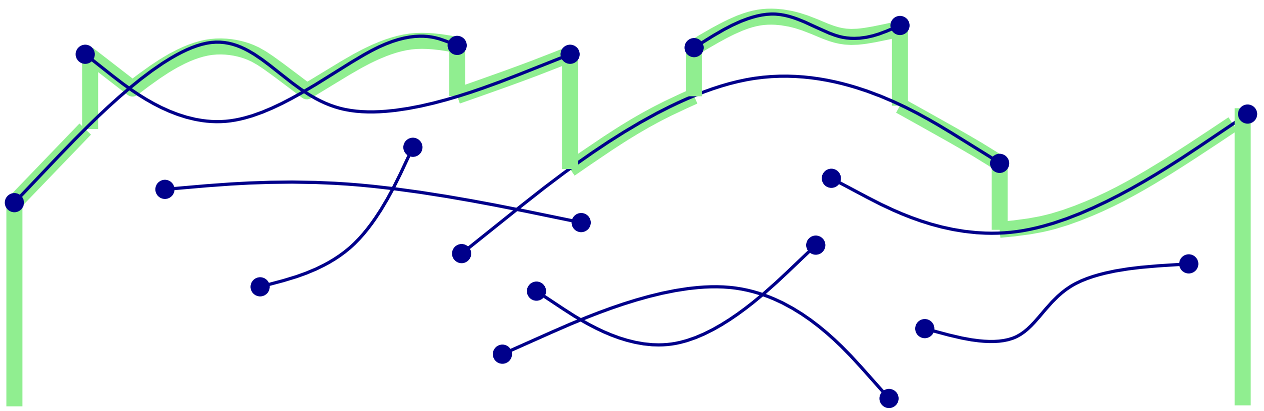
alternating sequence of length t implies $t-3$ intersections



Theorem. Let S be a set of n ~~infinite~~ x-monotone curves such that any two curves intersect at most s times. Then the maximum complexity of the upper envelope of S is $O(\mathrm{DS}_{s+2}(n))$.

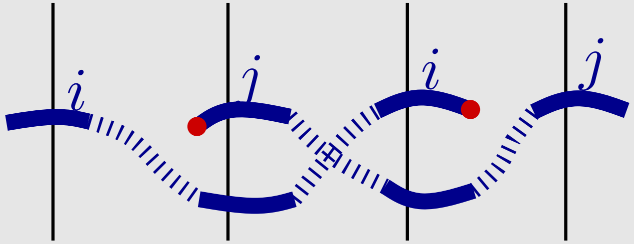
for example: $O(n\alpha(n))$ for line segments

Proof.



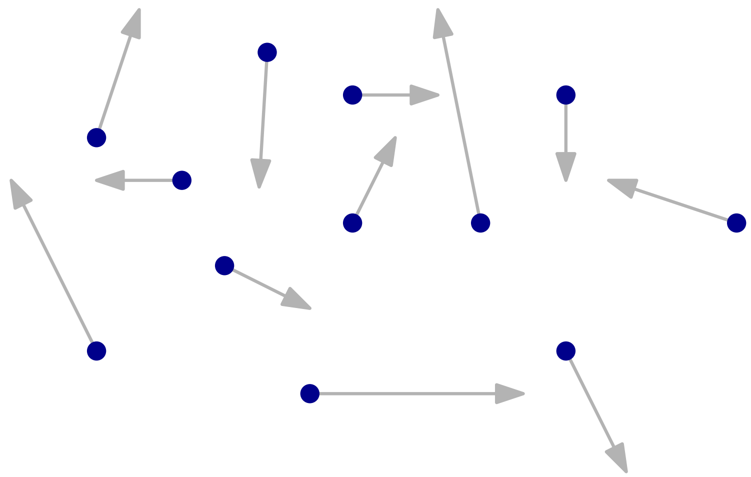
we cannot have alternating sequence of length $s+4$
$\Longrightarrow$ DS($n, s+2$)-sequence

alternating sequence of length t implies $t-3$ intersections



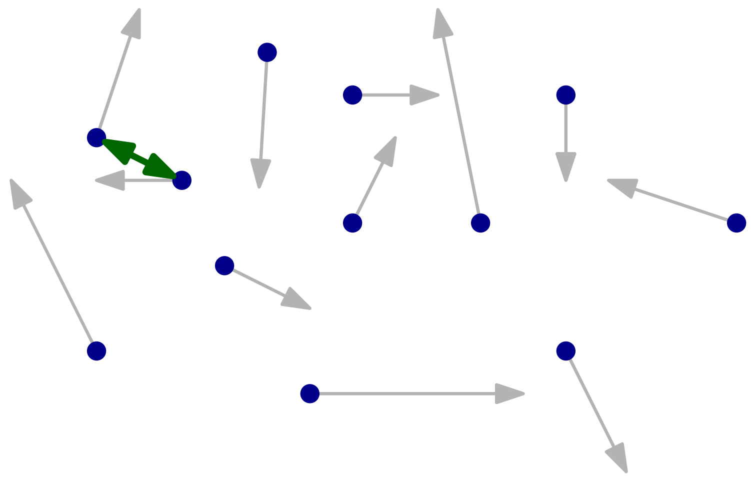
■

P: set of n points in $\mathbb{R}^2$ that move linearly (or: on polynomial trajectories)

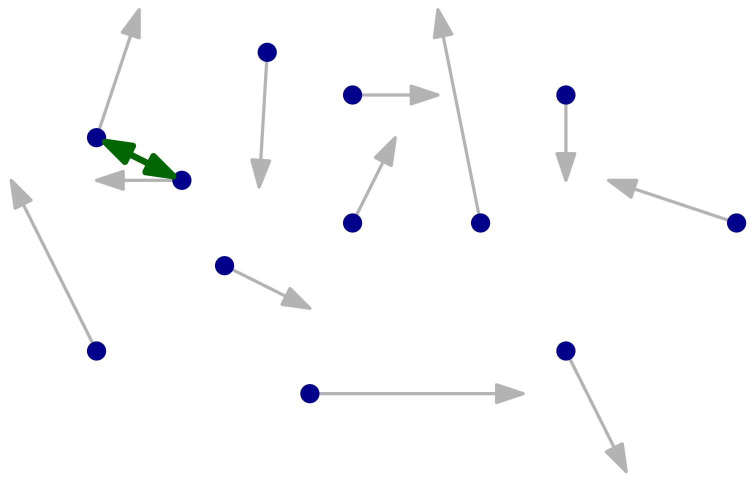


- How often can the closest pair change, in the worst case?
- How often can the convex hull change, in the worst case?
- How often can the Delaunay triangulation change, in the worst case?

How often can the closest pair change, in the worst case?

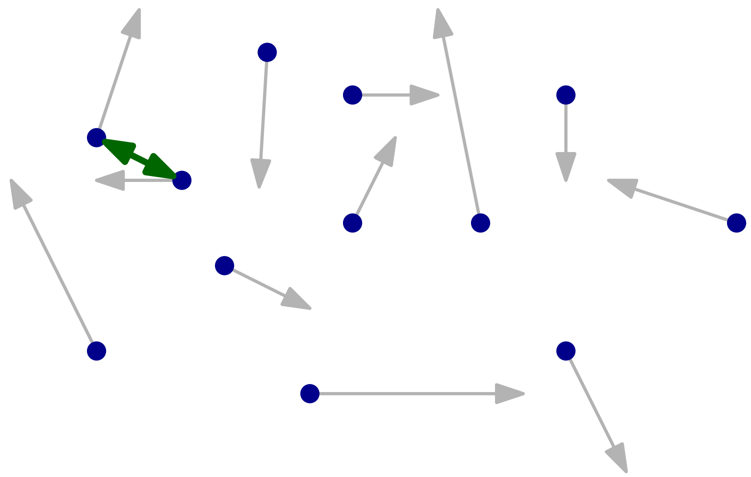


How often can the closest pair change, in the worst case?



Lower bound

How often can the closest pair change, in the worst case?

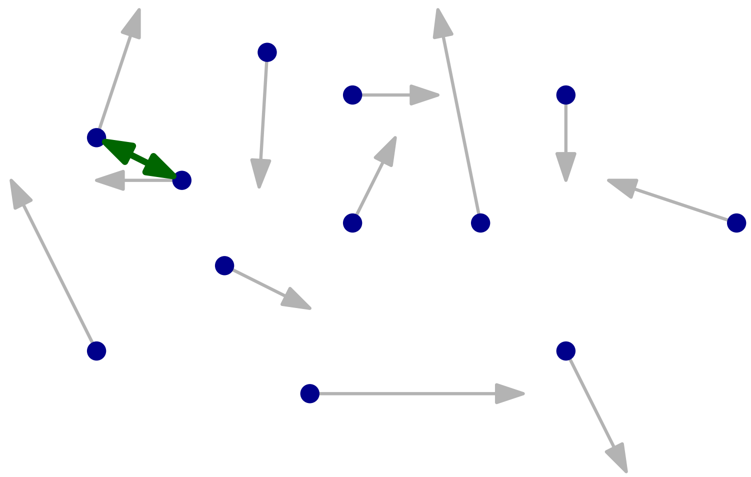


Lower bound

$\Omega(n^2)$ changes

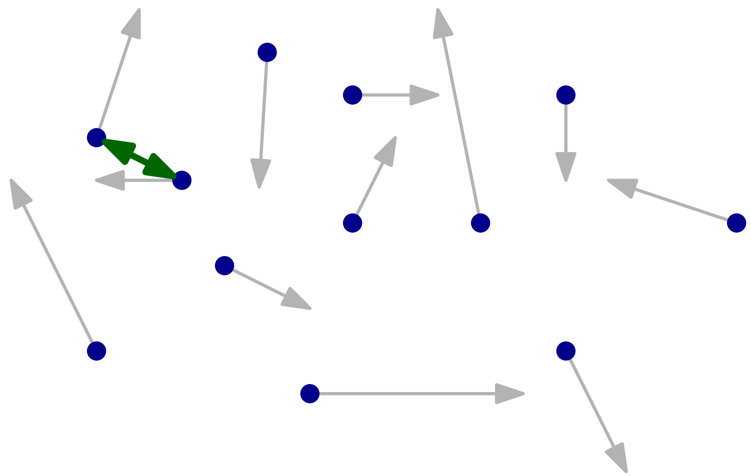
Upper Envelopes: Applications for Moving Ponits

How often can the closest pair change, in the worst case?



Upper bound

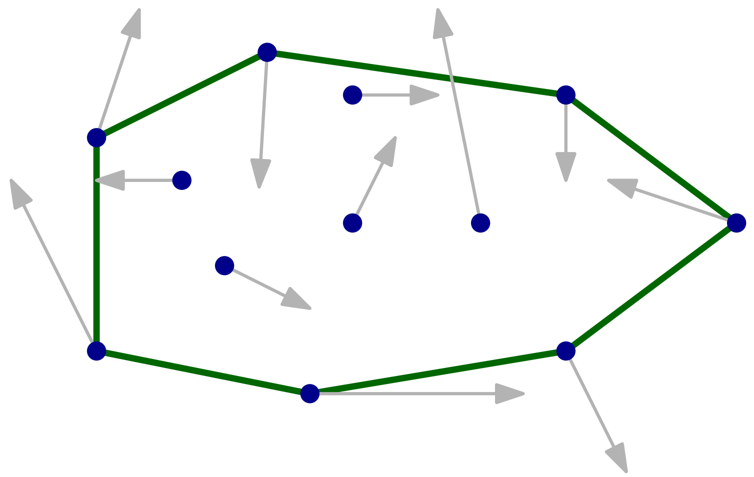
How often can the closest pair change, in the worst case?



Upper bound

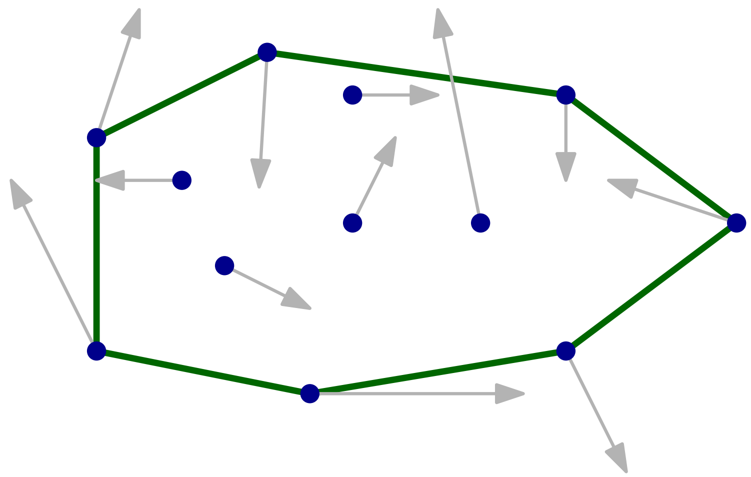
- for each pair p, q define $f_{pq}(t) :=$ distance between p and q at time t
- number of changes = complexity of lower envelope of n^2 functions
  $\approx O(n^2)$

How often can the convex hull change, in the worst case?

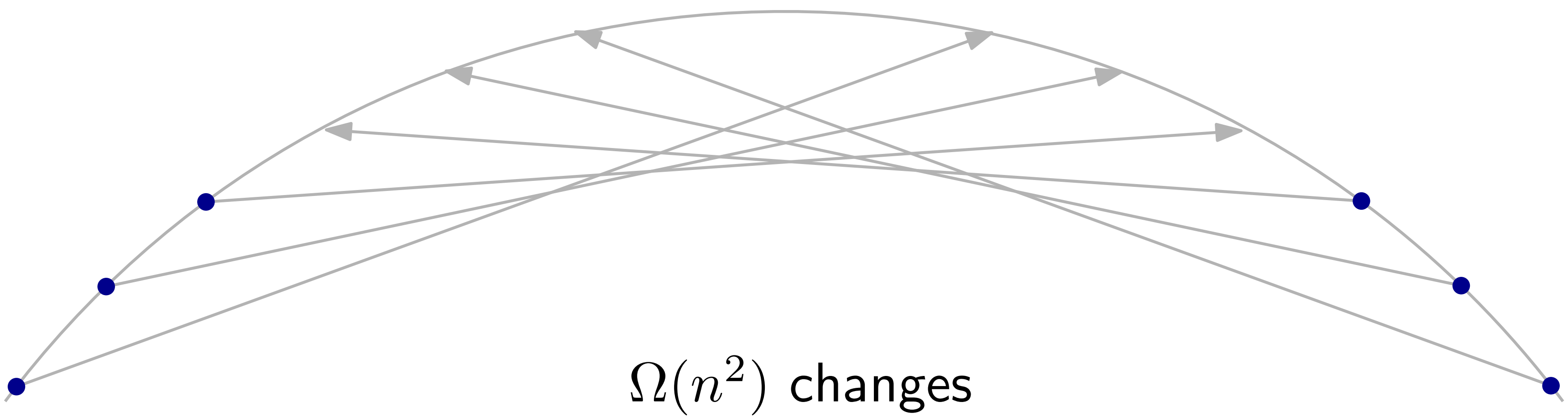


Lower bound

How often can the convex hull change, in the worst case?

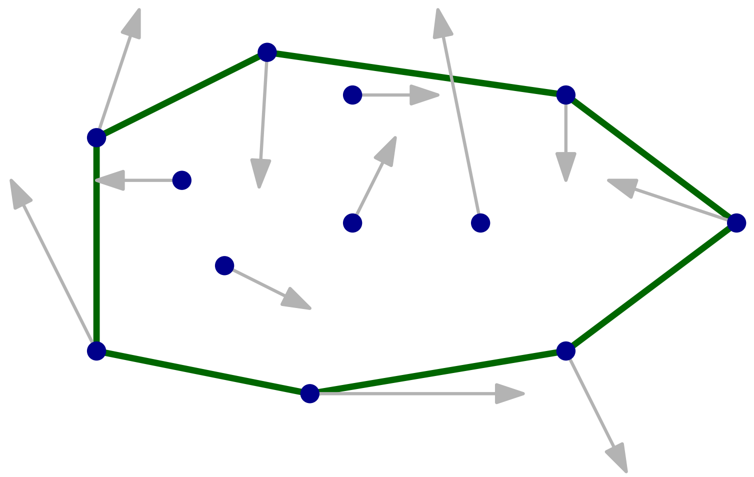


Lower bound



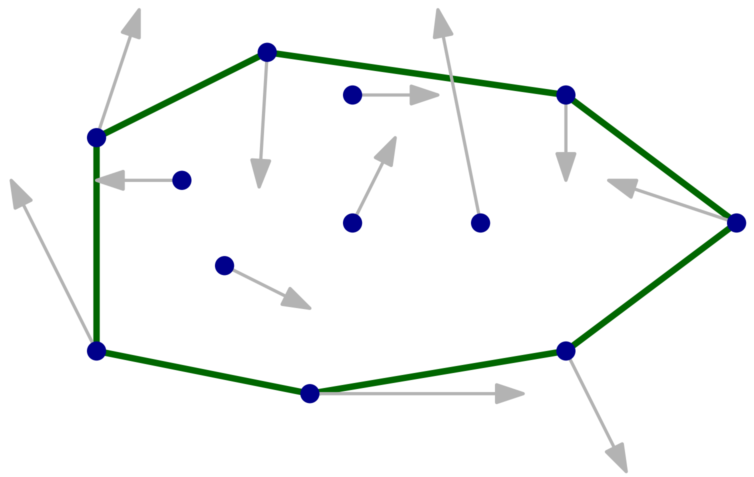
$\Omega(n^2)$ changes

How often can the convex hull change, in the worst case?



Trivial upper bound

How often can the convex hull change, in the worst case?



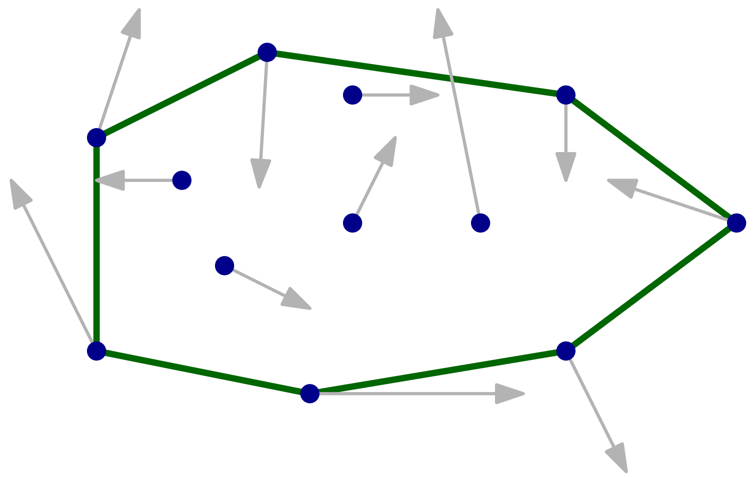
Trivial upper bound

convex hull changes $\Longrightarrow$ three points become collinear

$\Longrightarrow$ happens $O(1)$ times for each triple

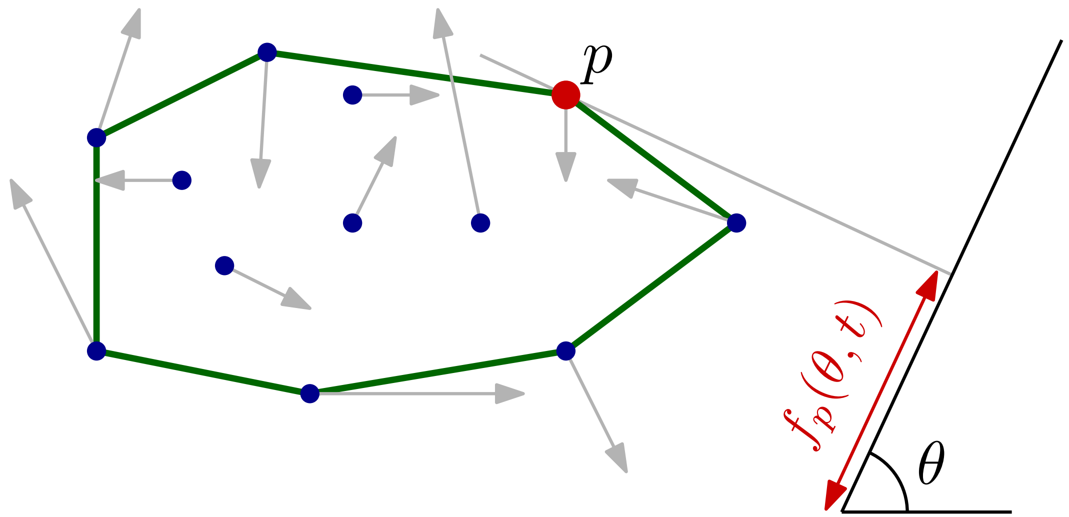
$\Longrightarrow$ $O(n^3)$ changes to convex hull

How often can the convex hull change, in the worst case?



A better bound using upper envelopes

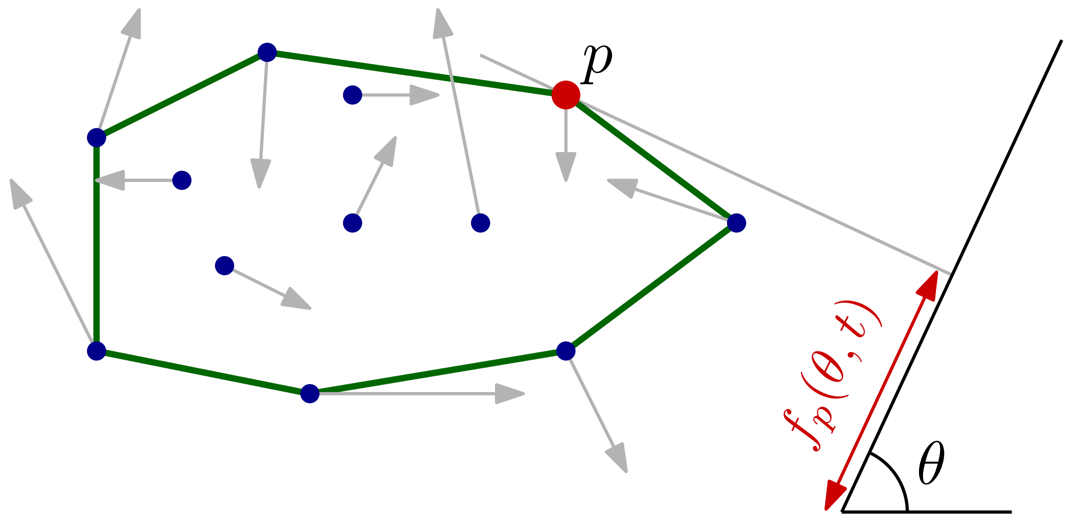
How often can the convex hull change, in the worst case?



A better bound using upper envelopes

- for each point p define function $f_p : [0, 2\pi) \times \mathbb{R}_{\geqslant 0} \to \mathbb{R}$

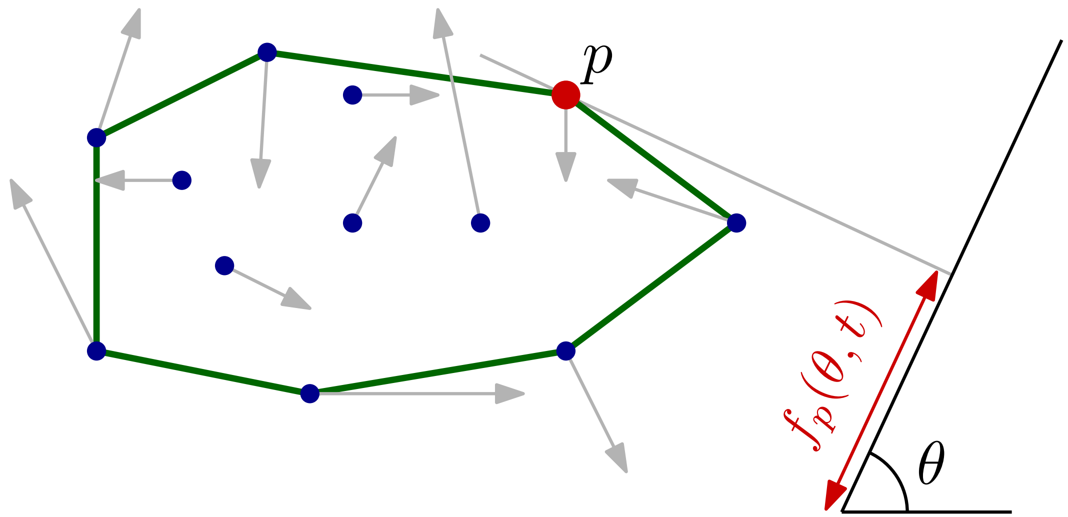
How often can the convex hull change, in the worst case?



A better bound using upper envelopes

- for each point p define function $f_p : [0, \pi) \times \mathbb{R}_{\geqslant 0} \rightarrow \mathbb{R}$
- p on convex hull at time t iff (there is a θ such that $f_p(\theta, t) \geqslant f_q(\theta, t)$ for all q at time t) or $(\ldots \leqslant \ldots)$

How often can the convex hull change, in the worst case?

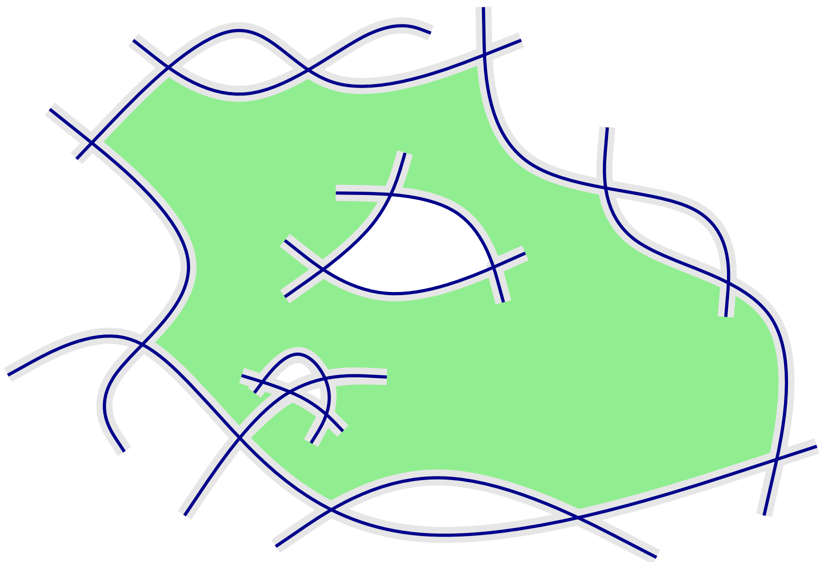


A better bound using upper envelopes

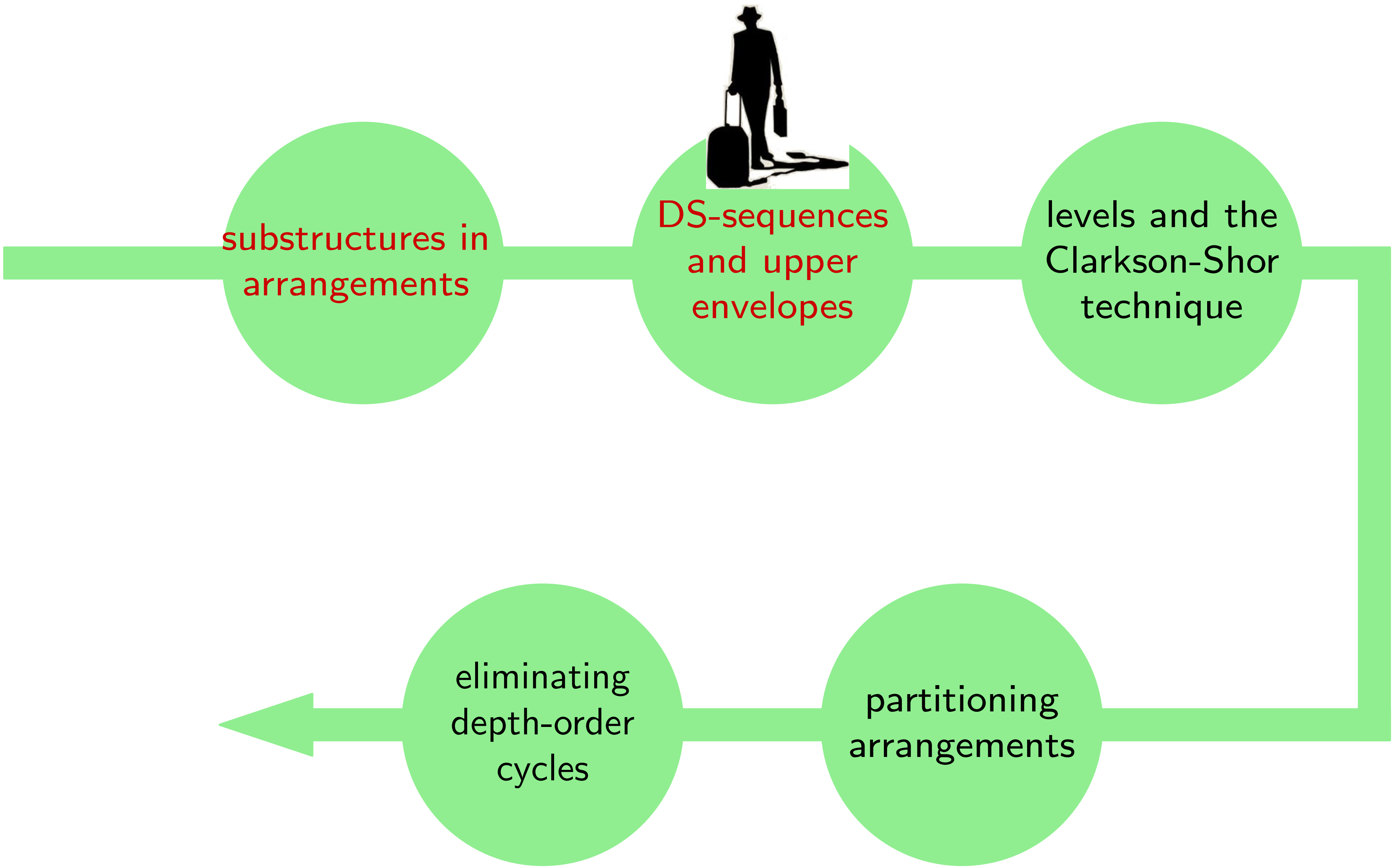
- for each point p define function $f_p : [0, \pi) \times \mathbb{R}_{\geqslant 0} \to \mathbb{R}$
- p on convex hull at time t iff (there is a θ such that $f_p(\theta, t) \geqslant f_q(\theta, t)$ for all q at time t) or $(\ldots \leqslant \ldots)$
- number of changes $= O($complexity of upper envelope of surfaces in $\mathbb{R}^3) = O(n^{2+\varepsilon})$

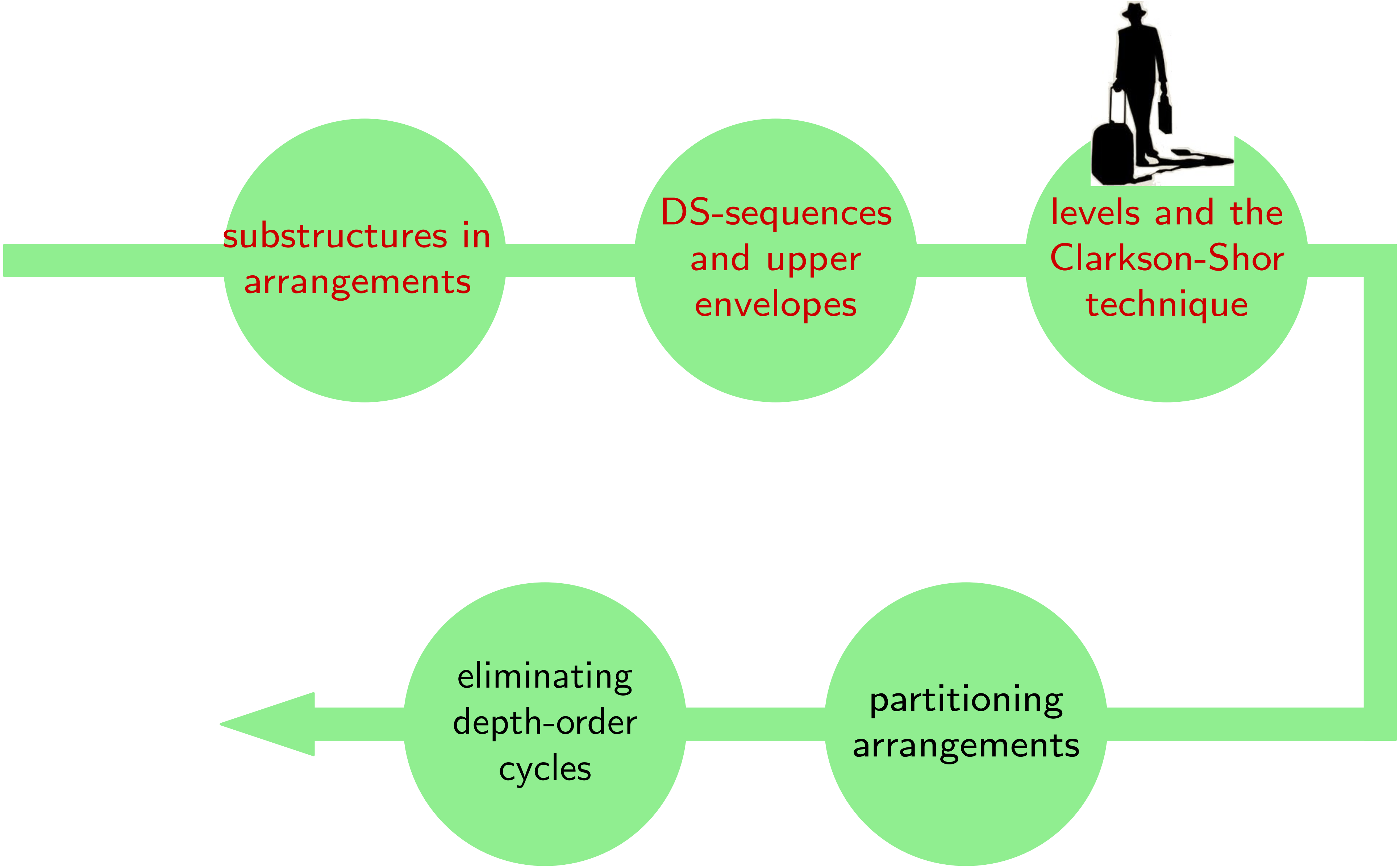
Theorem. Let S be a set of n curves in the plane such that any two curves intersect at most s times. Then the maximum complexity of a single cell of $\mathcal{A}(S)$ is $O(\mathrm{DS}_{s+2}(n))$.

for example: $O(n\alpha(n))$ for line segments

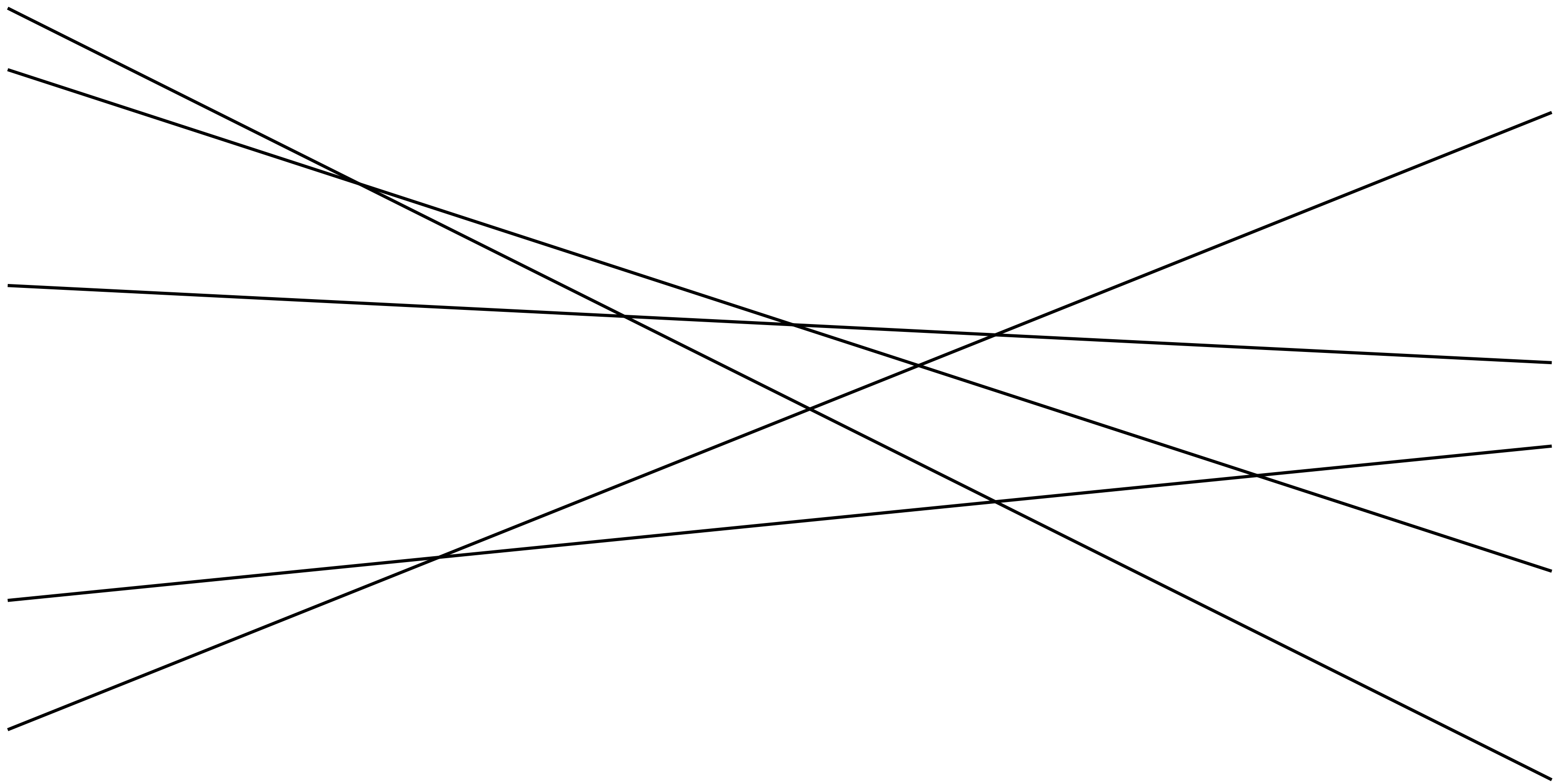


proof also uses Davenport-Schinzel sequences but is more complicated

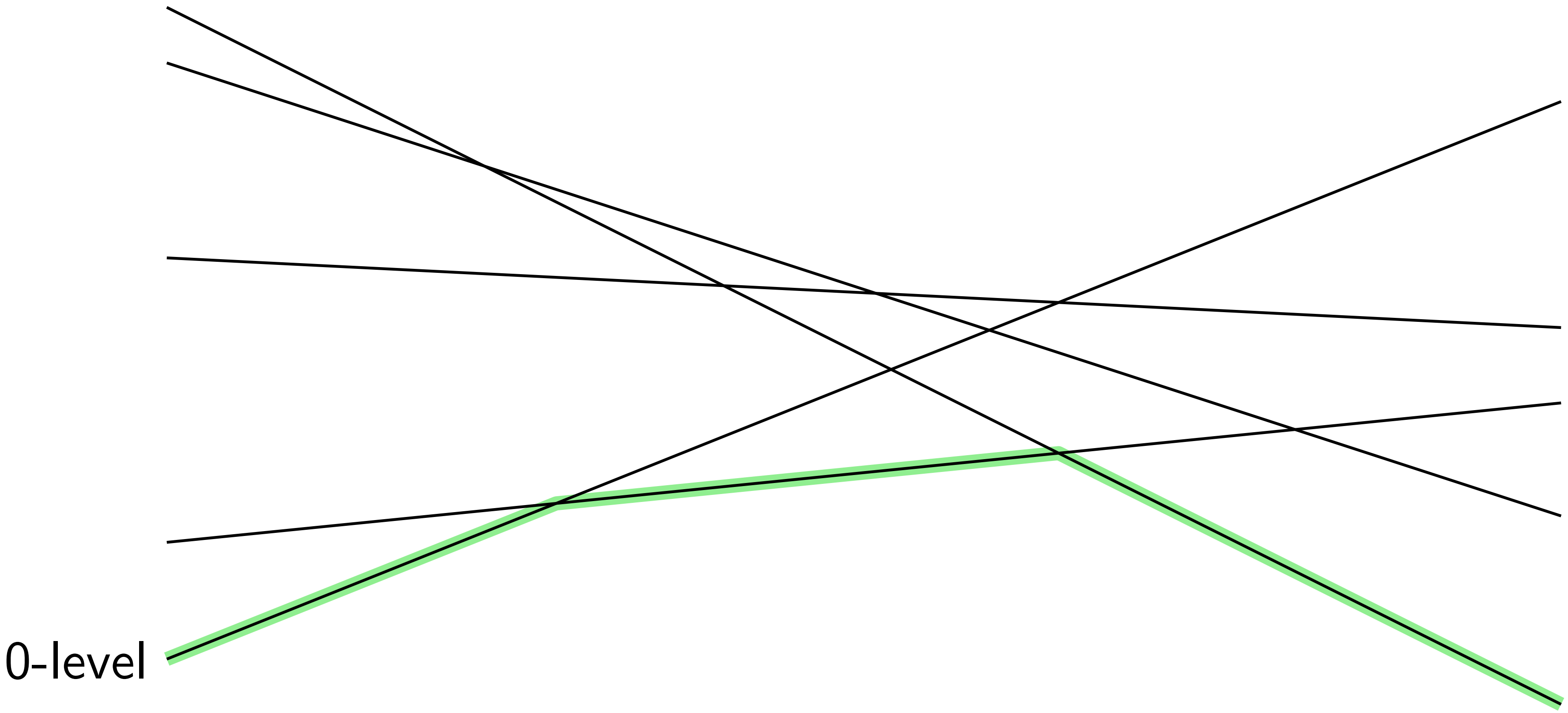

substructures in arrangements
DS-sequences and upper envelopes
levels and the Clarkson-Shor technique
partitioning arrangements
eliminating depth-order cycles


substructures in arrangements
DS-sequences and upper envelopes
levels and the Clarkson-Shor technique
partitioning arrangements
eliminating depth-order cycles

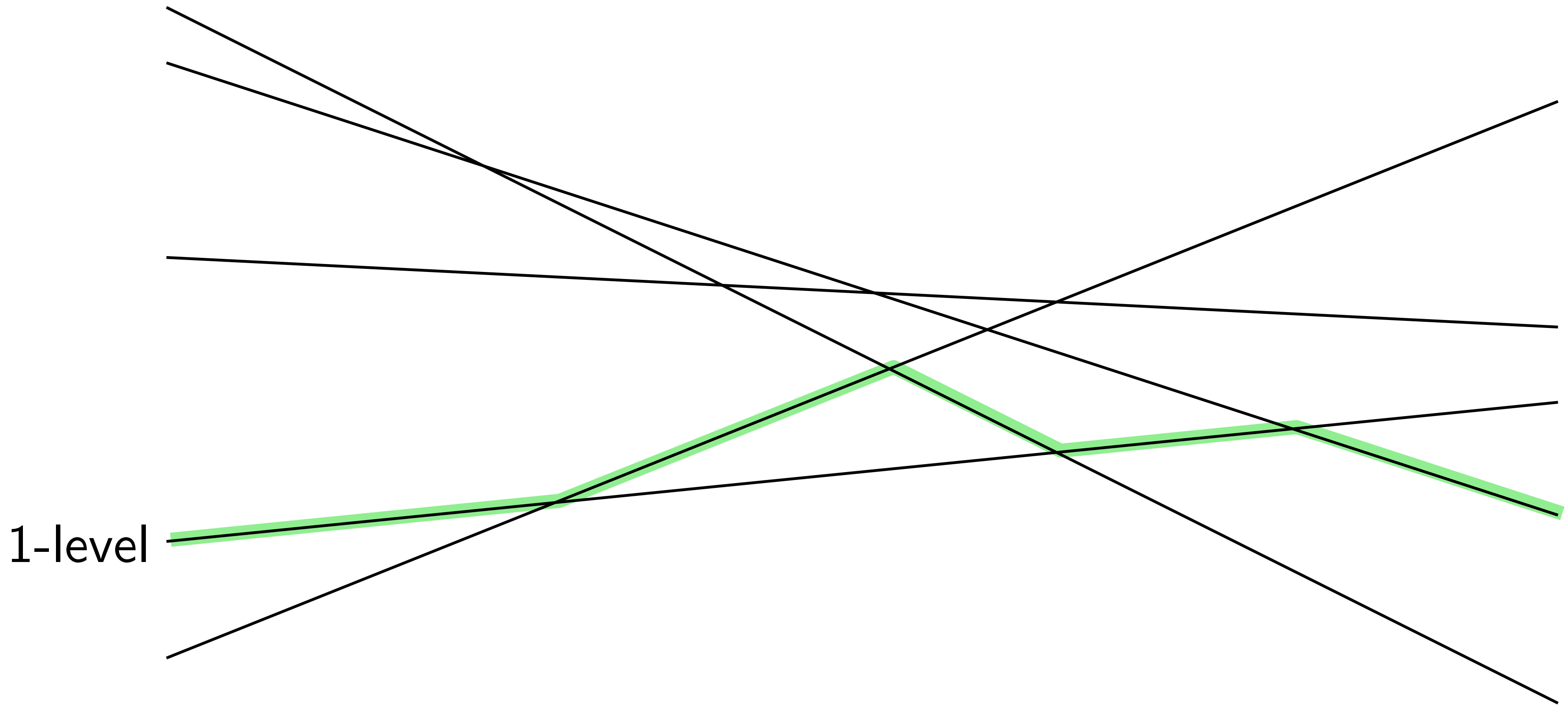
Levels in arrangements

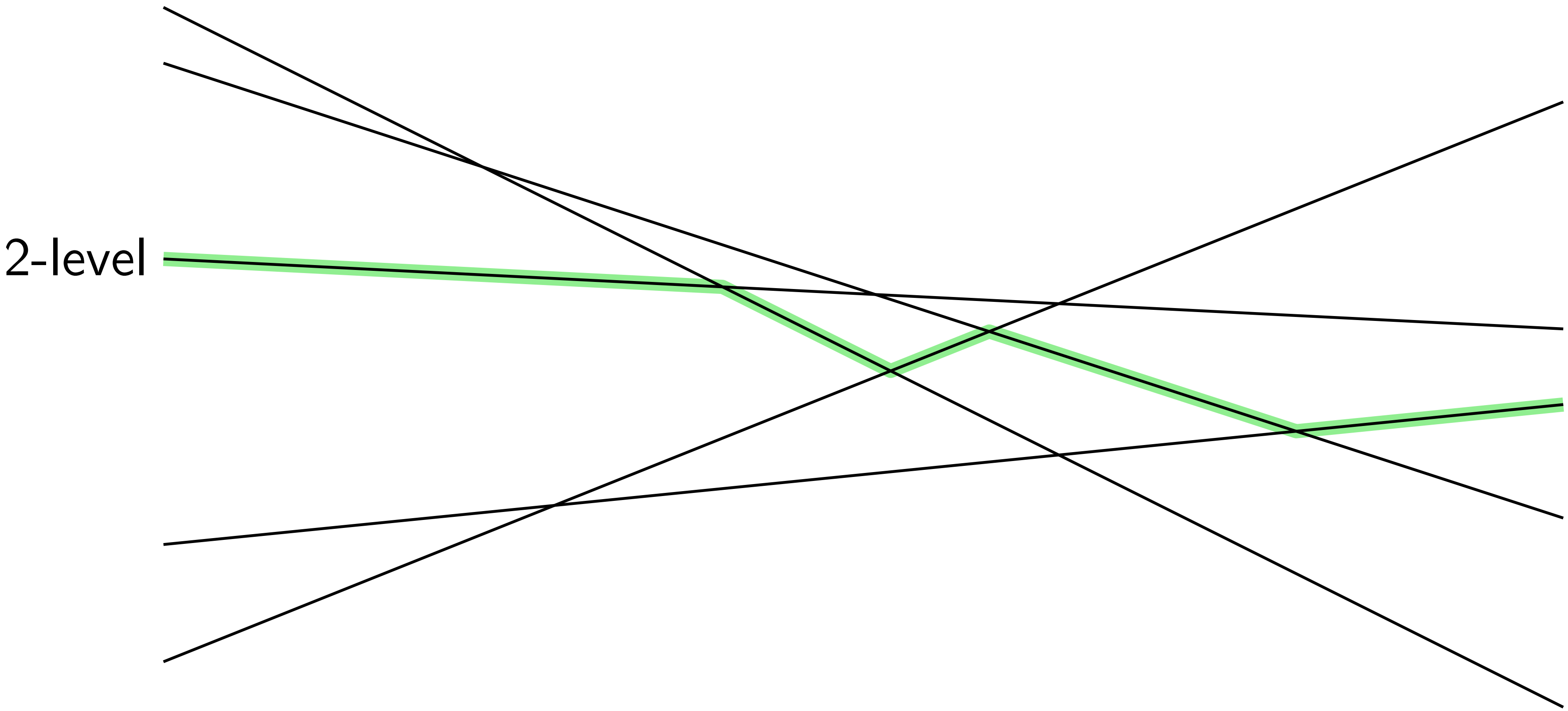


Levels in arrangements

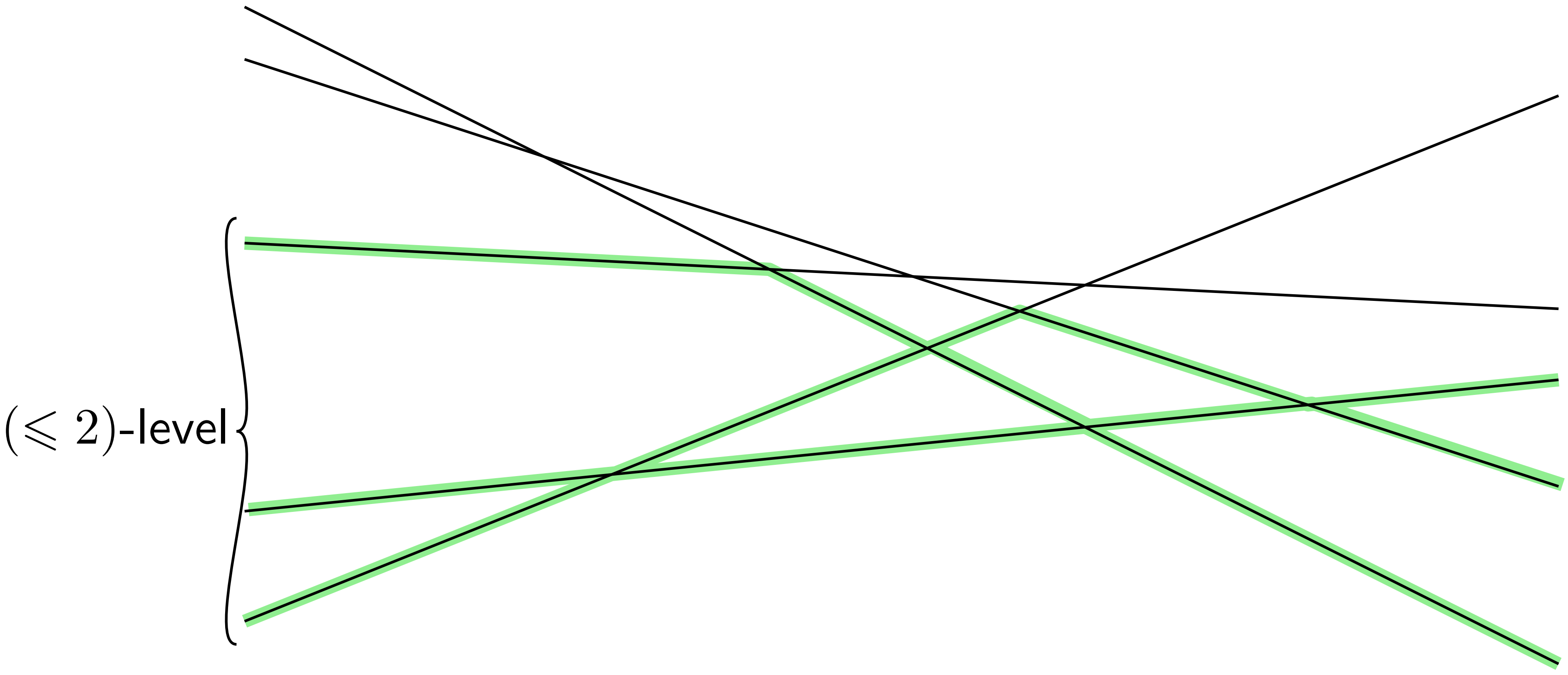


Levels in arrangements

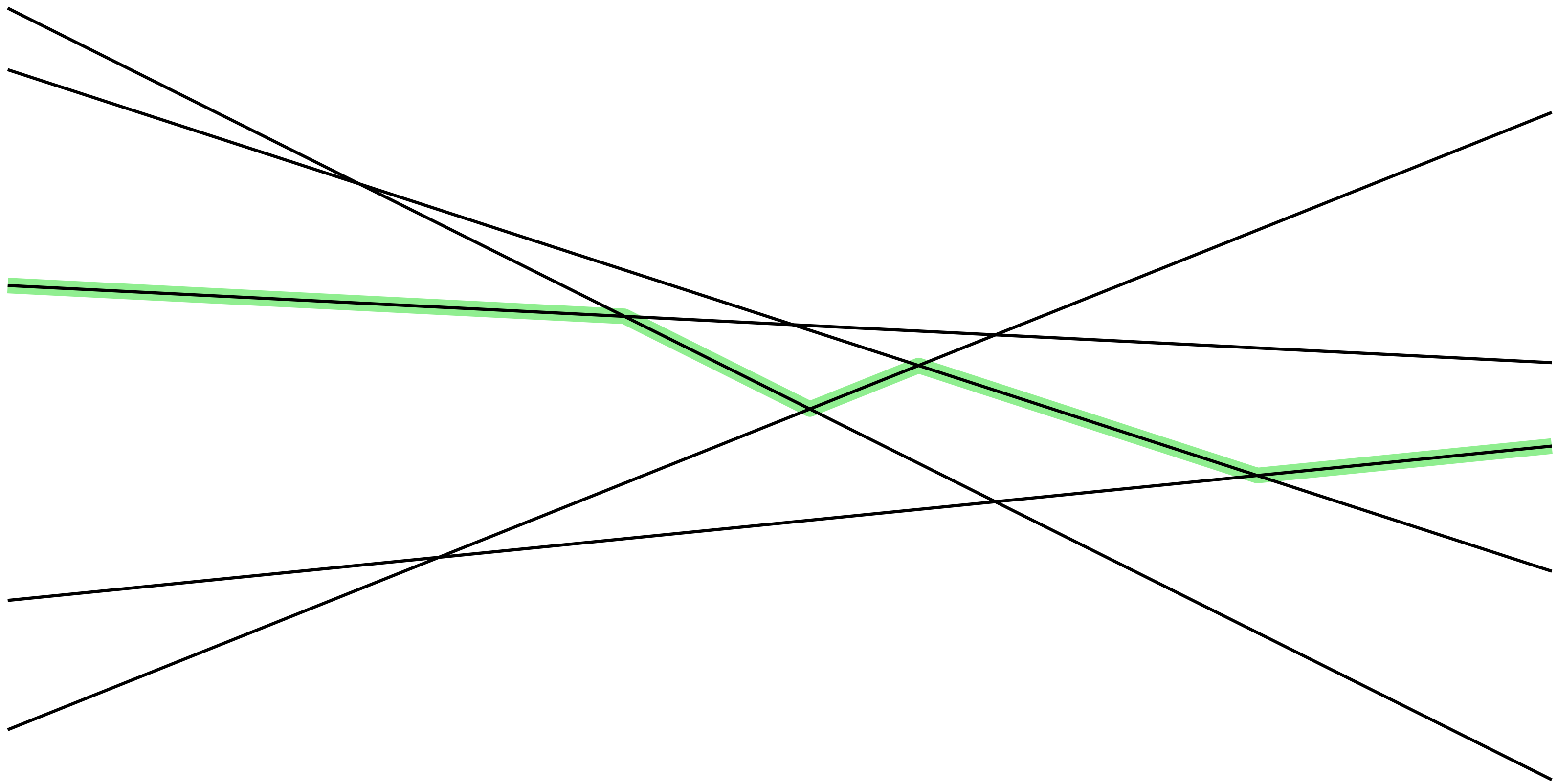



2-level

Levels in arrangements

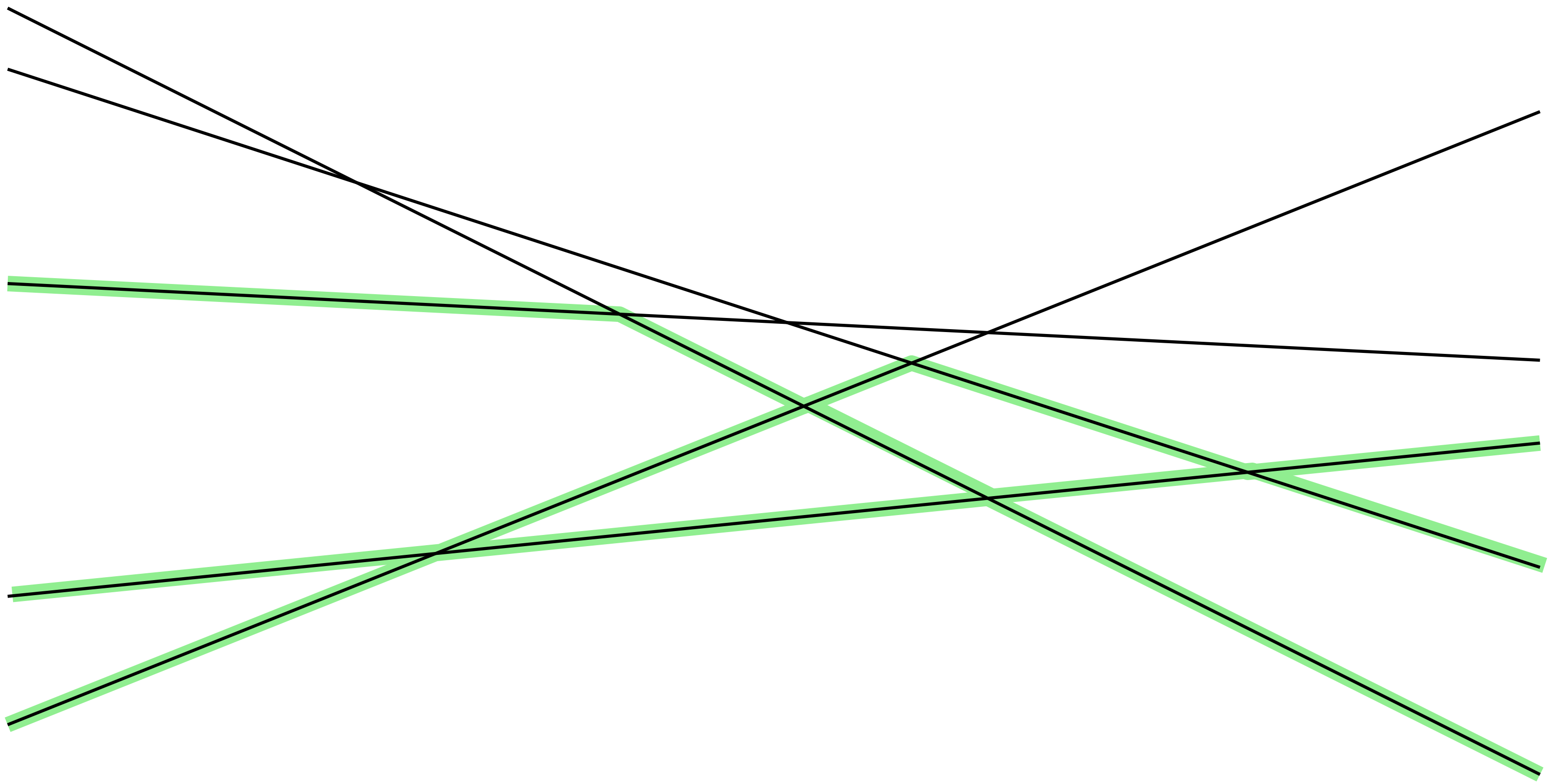


Levels in arrangements

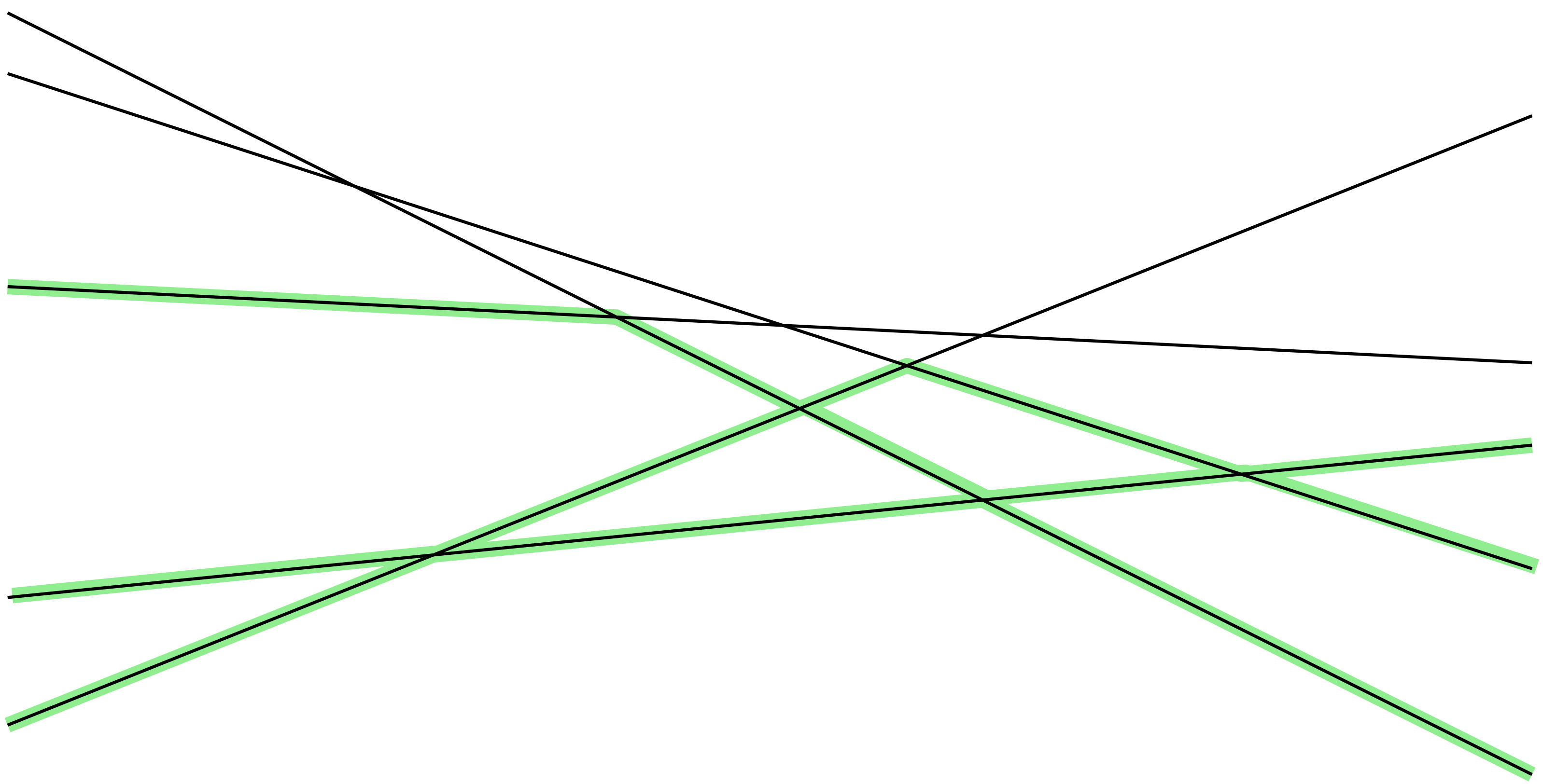


What is the max complexity of the k-level in an arrangement of n lines?

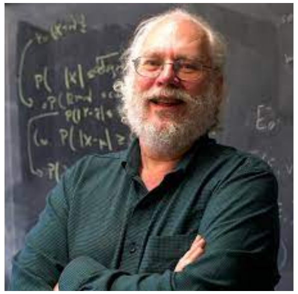
- 0-level $=$ lower envelope $\implies$ complexity $\leqslant n$
- $k \geqslant 1$: complexity is $n2^{\Omega(\sqrt{\log k})}$ and $O(nk^{1/3})$ major open problem



What is the max complexity of the $(\leqslant k)$-level in an arrangement of n lines?



What is the max complexity of the $(\leqslant k)$-level in an arrangement of n lines?



Clarkson-Shor '89: $\Theta(nk)$

- in $\mathbb{R}^d$: $\Theta(n^{\lfloor d/2 \rfloor} k^{\lceil d/2 \rceil})$
- bound for $d = 2$ was already known

Theorem. The max complexity of the $(\leqslant k)$-level in an arrangement induced by a set L of n lines in the plane is $O(nk)$.

Theorem. The max complexity of the $(\leqslant k)$-level in an arrangement induced by a set L of n lines in the plane is $O(nk)$.

Proof.

Theorem. The max complexity of the $(\leqslant k)$-level in an arrangement induced by a set L of n lines in the plane is $O(nk)$.

Proof.

Take sample $R \subset L$ by picking each line $\ell \in L$ with probability $1/k$.

Theorem. The max complexity of the $(\leqslant k)$-level in an arrangement induced by a set L of n lines in the plane is $O(nk)$.

Proof.

Take sample $R \subset L$ by picking each line $\ell \in L$ with probability $1/k$.

$$\mathbb{E}\big[\text{complexity of 0-level of } R\big] \;\leqslant\; \mathbb{E}\big[|R|\big] \;=\; n/k$$

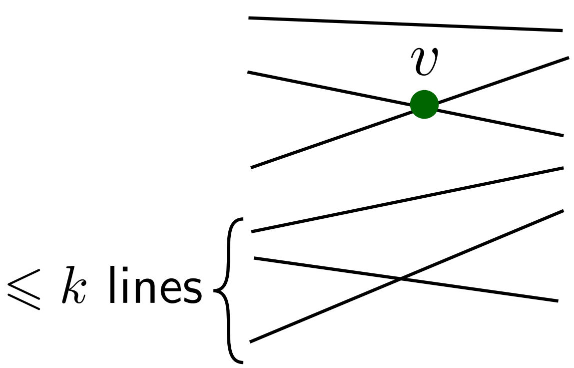
The Clarkson-Shor Technique: Application to $(\leqslant k)$-levels

Theorem. The max complexity of the $(\leqslant k)$-level in an arrangement induced by a set L of n lines in the plane is $O(nk)$.

Proof.

Take sample $R \subset L$ by picking each line $\ell \in L$ with probability $1/k$.

$$\mathbb{E}\big[\text{complexity of 0-level of } R\big] \leqslant \mathbb{E}\big[|R|\big] = n/k$$



vertex of k-level of L shows up on 0-level of R iff

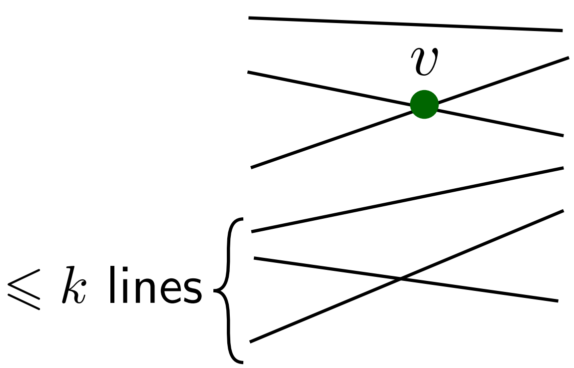
- both lines defining v are in R
- none of the at most k lines below v are in R

Theorem. The max complexity of the $(\leqslant k)$-level in an arrangement induced by a set L of n lines in the plane is $O(nk)$.

Proof.

Take sample $R \subset L$ by picking each line $\ell \in L$ with probability $1/k$.

$$\mathbb{E}\big[\text{complexity of 0-level of } R\big] \;\leqslant\; \mathbb{E}\big[|R|\big] \;=\; n/k$$



vertex of k-level of L shows up on 0-level of R iff

- both lines defining v are in R
- none of the at most k lines below v are in R

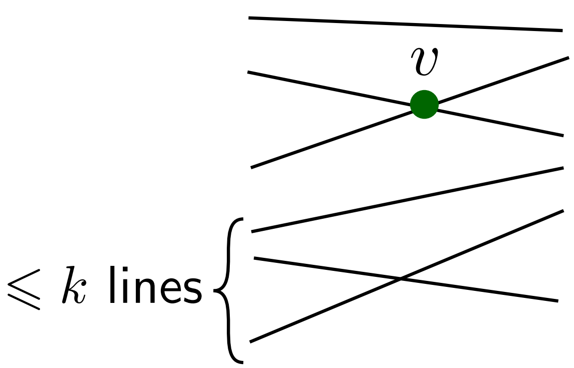
$$\text{prob} \geqslant \left(\tfrac{1}{k}\right)^2 \cdot \left(1 - \tfrac{1}{k}\right)^k \geqslant \left(\tfrac{1}{k}\right)^2 \cdot \tfrac{1}{e}$$

Theorem. The max complexity of the $(\leqslant k)$-level in an arrangement induced by a set L of n lines in the plane is $O(nk)$.

Proof.

Take sample $R \subset L$ by picking each line $\ell \in L$ with probability $1/k$.

$$\mathbb{E}\big[\text{complexity of 0-level of } R\big] \leqslant \mathbb{E}\big[|R|\big] = n/k$$



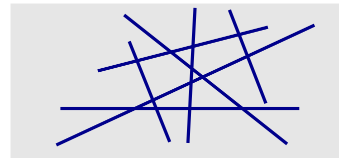
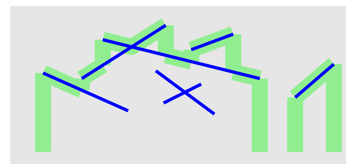
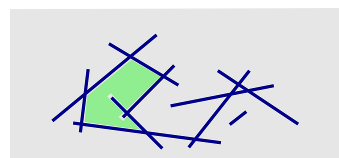
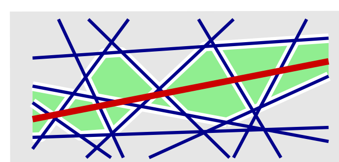
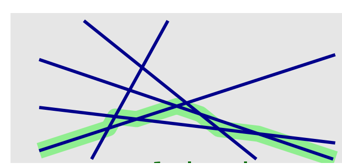
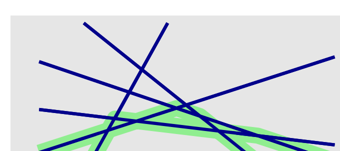
vertex of k-level of L shows up on 0-level of R iff

- both lines defining v are in R
- none of the at most k lines below v are in R

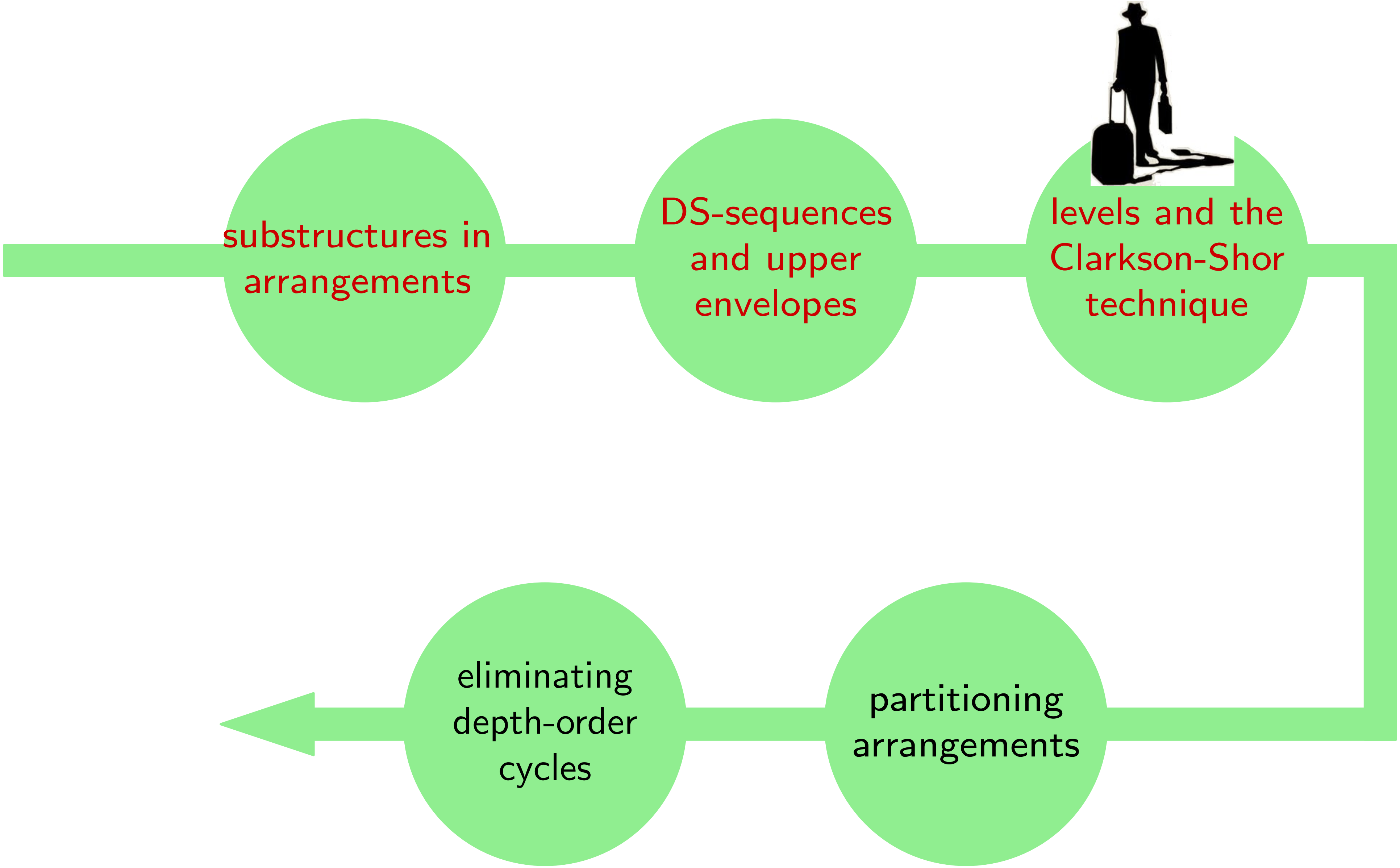
$$\text{prob} \geqslant \left(\tfrac{1}{k}\right)^2 \cdot \left(1 - \tfrac{1}{k}\right)^k \geqslant \left(\tfrac{1}{k}\right)^2 \cdot \tfrac{1}{e}$$

$$\mathbb{E}\big[\text{complexity of 0-level of } R\big] \geqslant (\text{complexity of } k\text{-level in } L) \cdot \left(\tfrac{1}{k}\right)^2 \cdot \tfrac{1}{e}$$

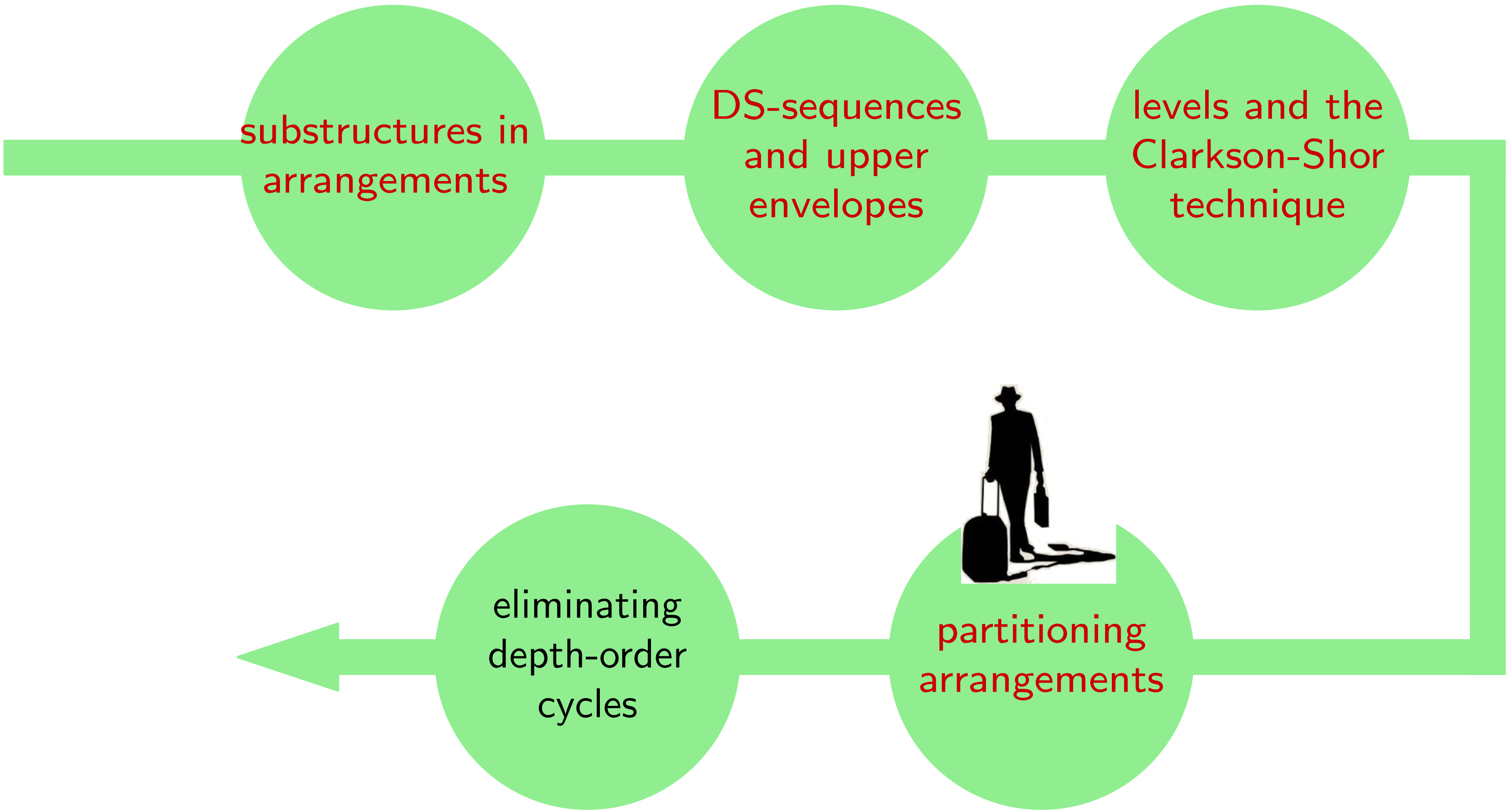
Overview of Complexity of Substructures in Arrangements in $\mathbb{R}^2$

full arrangement		$\Theta(n^2)$
upper envelope (segments)		$\Theta(n\alpha(n))$
single cell (segments)		$\Theta(n\alpha(n))$
zone (lines)		$\Theta(n)$
k-level (lines)		$O(nk^{1/3})$ $\quad n2^{\Omega(\sqrt{\log k})}$
($\leqslant k$)-level (lines)		$\Theta(nk)$

Lecture Overview



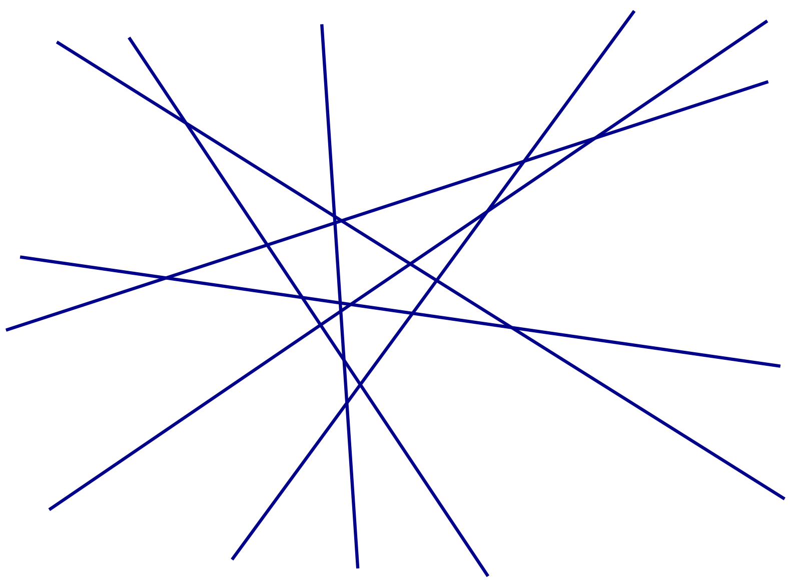
Lecture Overview



- divide-and-conquer: important algorithmic design technique
- for geometric problems: perform divide step by partitioning space

- divide-and-conquer: important algorithmic design technique
- for geometric problems: perform divide step by partitioning space

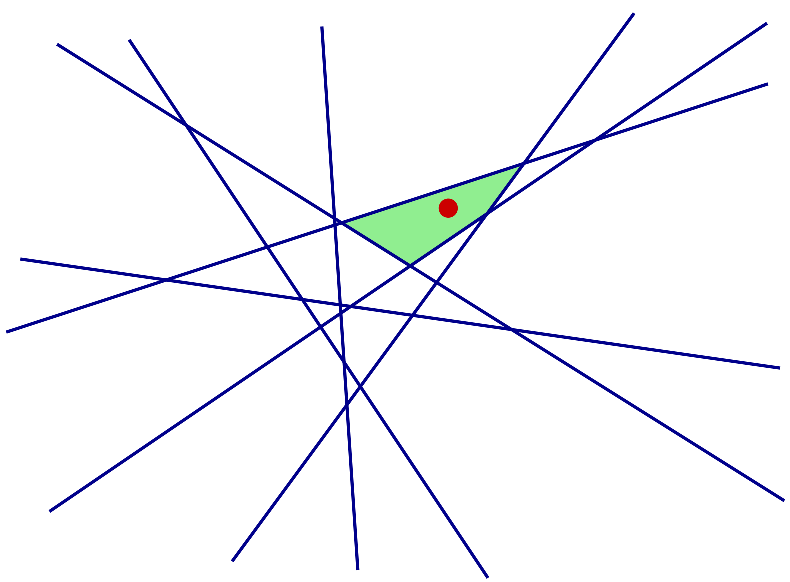
Example: point location in arrangements



Store lines in data structure such that we can find the cell containing a query point q in $O(\log n)$ time

- divide-and-conquer: important algorithmic design technique
- for geometric problems: perform divide step by partitioning space

Example: point location in arrangements

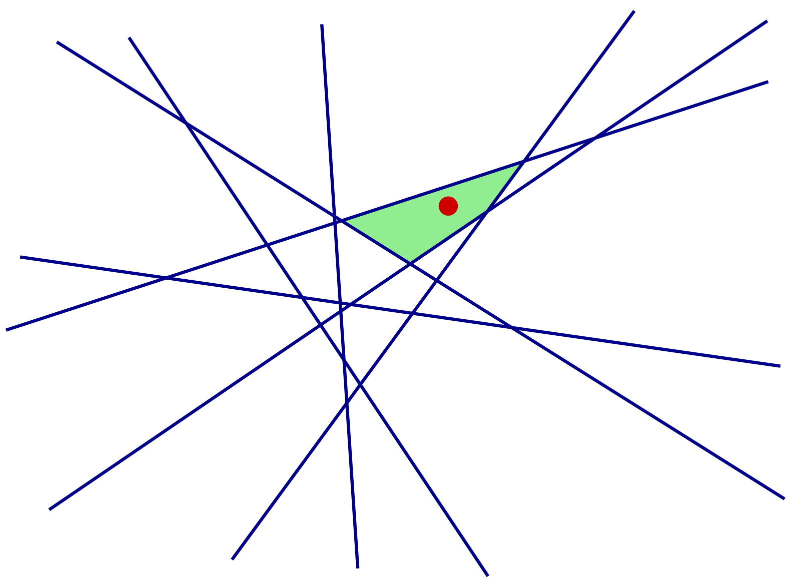


Store lines in data structure such that we can find the cell containing a query point q in $O(\log n)$ time

Point Location in Arrangements

- divide-and-conquer: important algorithmic design technique
- for geometric problems: perform divide step by partitioning space

Example: point location in arrangements



Store lines in data structure such that we can find the cell containing a query point q in $O(\log n)$ time

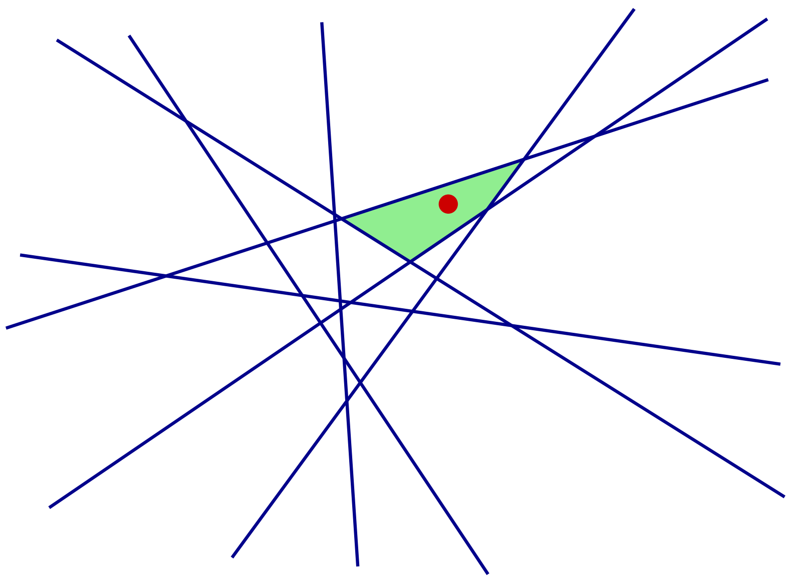
Idea

- Partition plane into small number of regions
- Find region containing query point q
- Recursively find cell containing q within region

Point Location in Arrangements

- divide-and-conquer: important algorithmic design technique
- for geometric problems: perform divide step by partitioning space

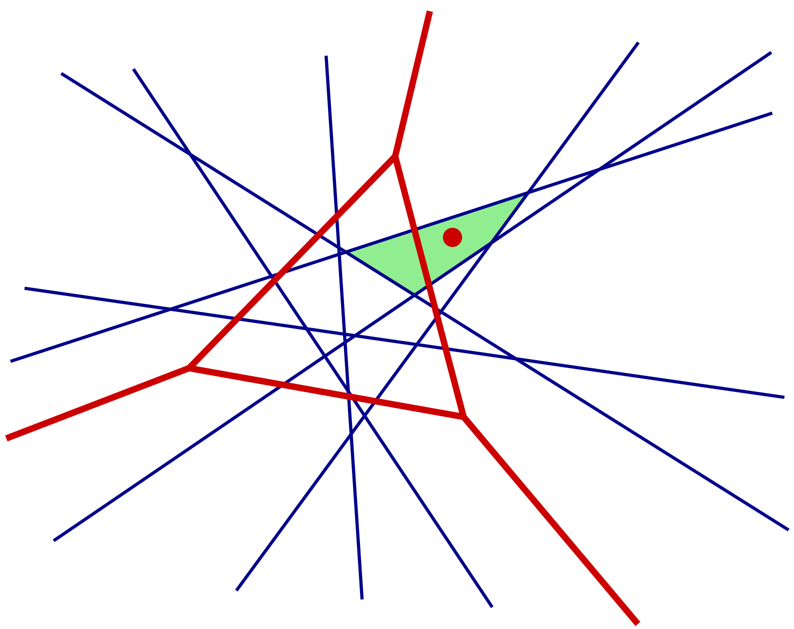
Example: point location in arrangements



Store lines in data structure such that we can find the cell containing a query point q in $O(\log n)$ time

- divide-and-conquer: important algorithmic design technique
- for geometric problems: perform divide step by partitioning space

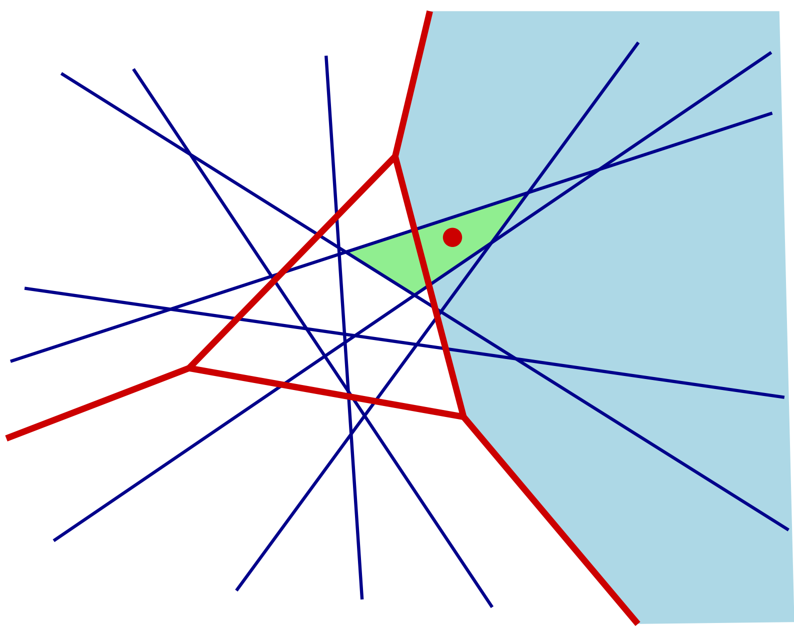
Example: point location in arrangements

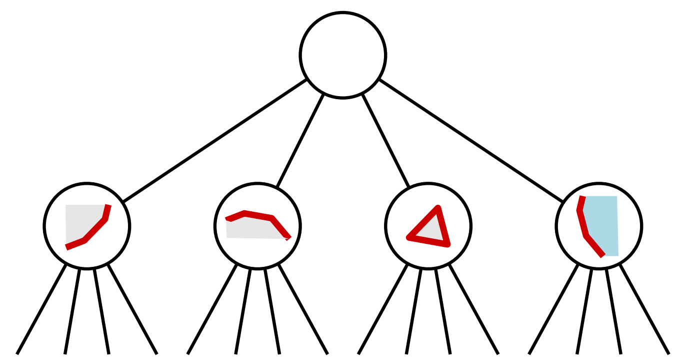


Store lines in data structure such that we can find the cell containing a query point q in $O(\log n)$ time

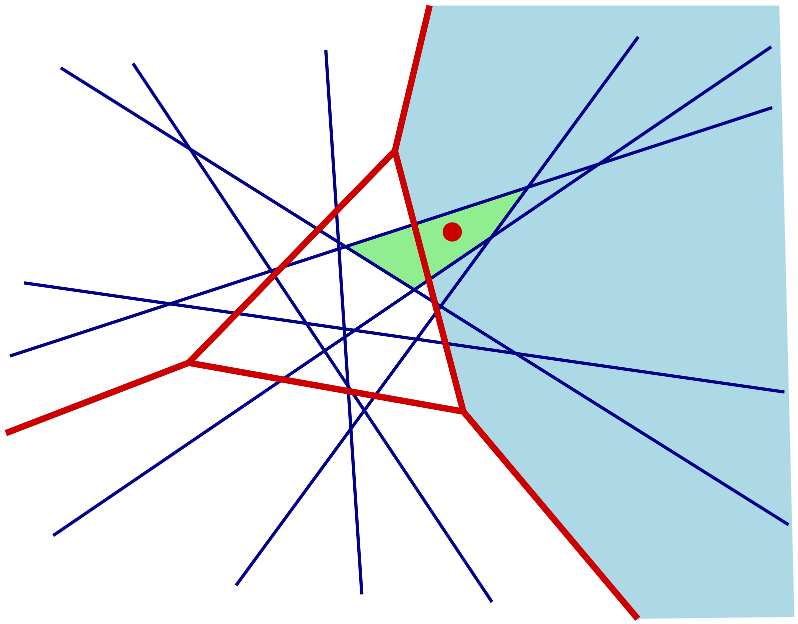
- divide-and-conquer: important algorithmic design technique
- for geometric problems: perform divide step by partitioning space

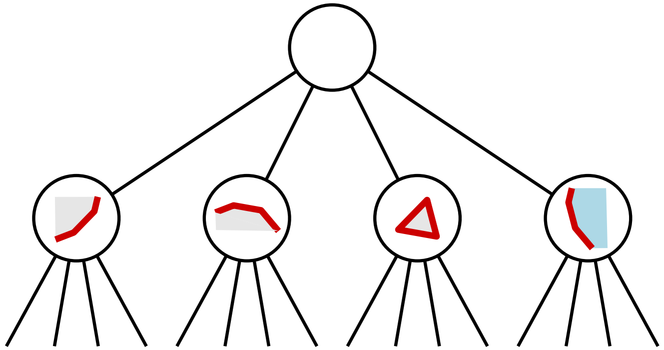
Example: point location in arrangements





Store lines in data structure such that we can find the cell containing a query point q in $O(\log n)$ time



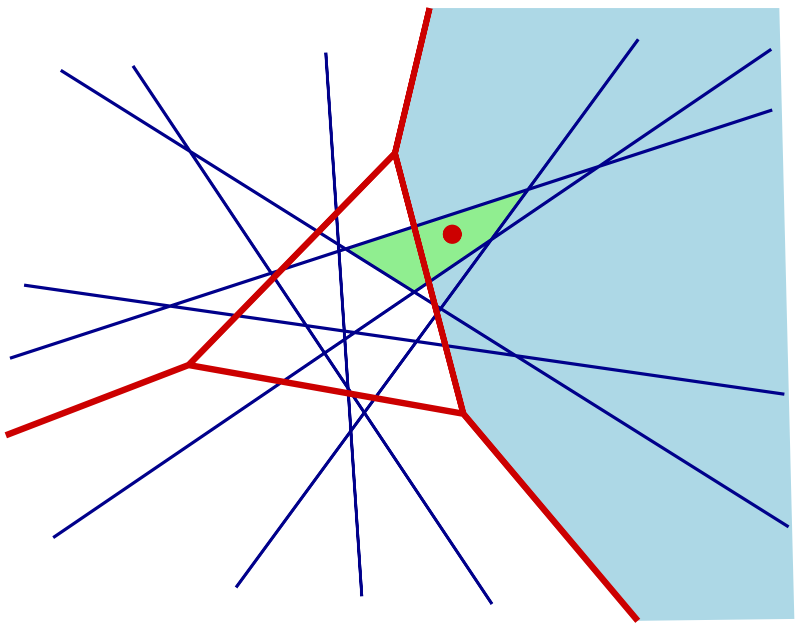


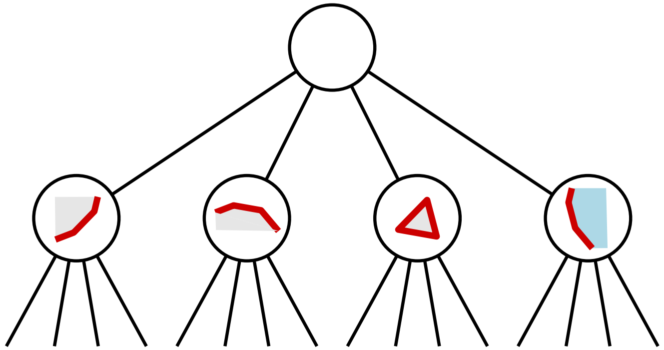
Analysis

Suppose each region intersects at most n/r lines, for some constant r

- query time:

- storage:

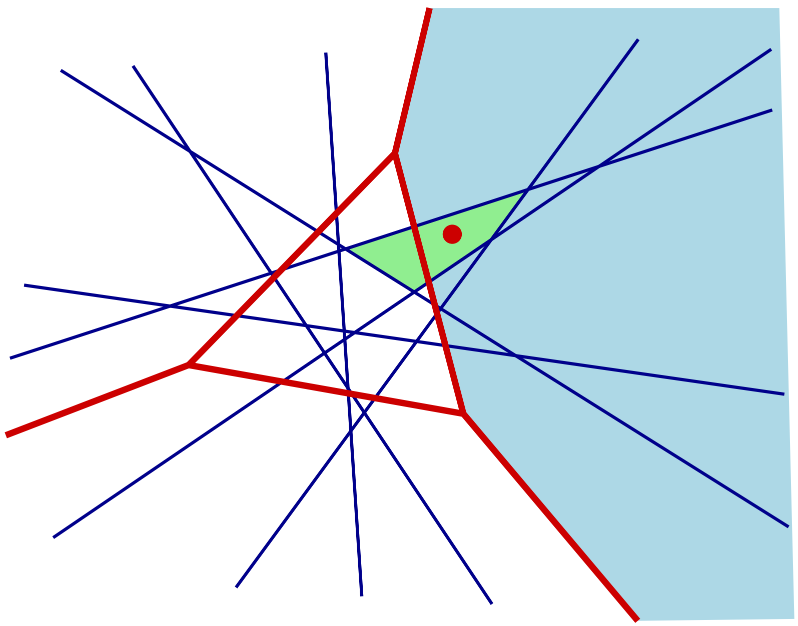


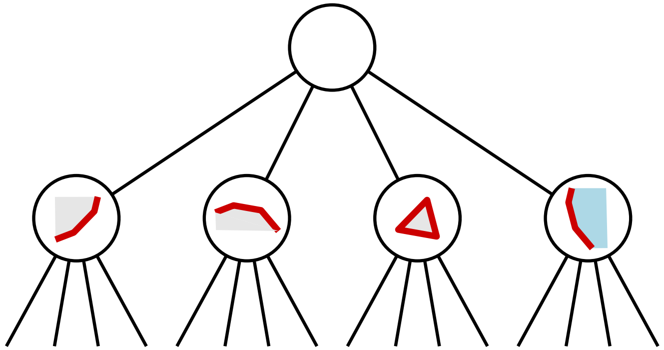


Analysis

Suppose each region intersects at most n/r lines, for some constant r

- query time: $Q(n) = O(r) + Q(n/r) \quad \Longrightarrow \quad Q(n) = O(\log n)$
- storage:

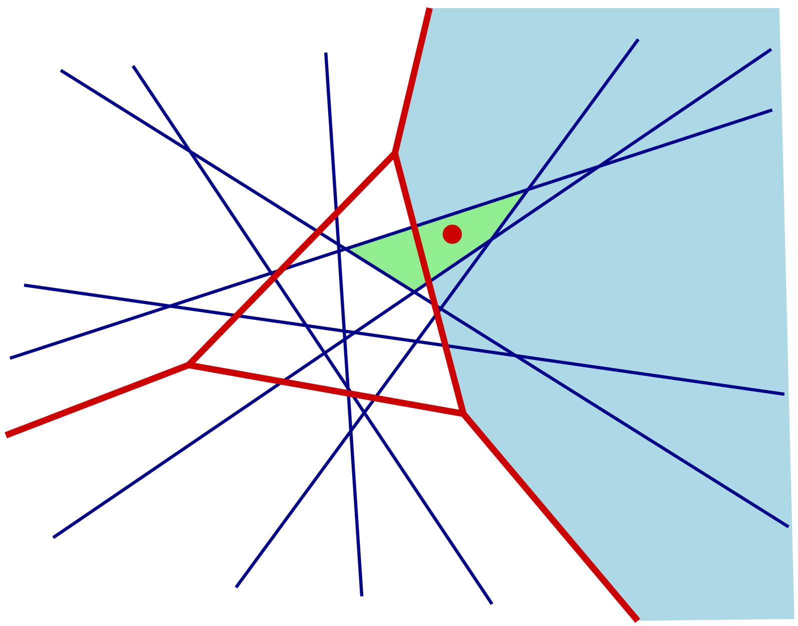


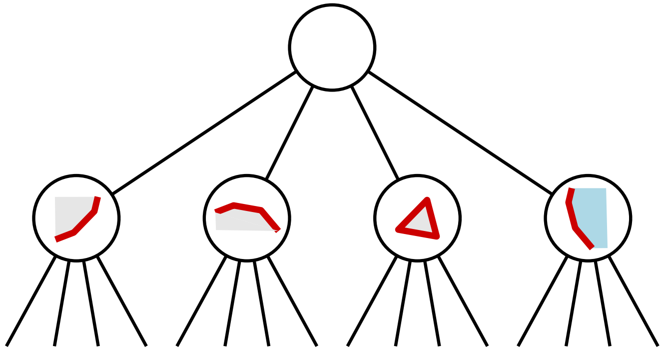


Analysis

Suppose each region intersects at most n/r lines, for some constant r

- query time: $Q(n) = O(r) + Q(n/r) \quad \Longrightarrow \quad Q(n) = O(\log n)$
- storage: $S(n) = (\text{number of regions}) \cdot S(n/r)$





Analysis

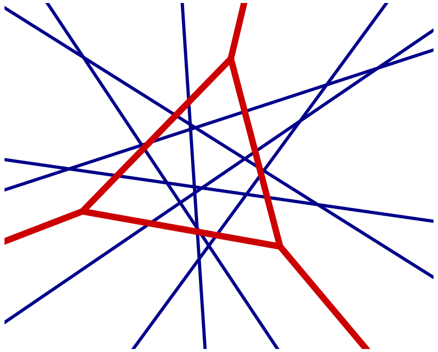
Suppose each region intersects at most n/r lines, for some constant r

- query time: $Q(n) = O(r) + Q(n/r) \implies Q(n) = O(\log n)$
- storage: $S(n) = (\text{number of regions}) \cdot S(n/r)$

  $(\text{number of regions}) = O(r^2) \implies S(n) = O(n^{2+\varepsilon})$

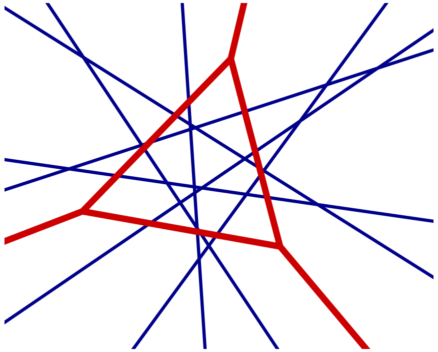
$(1/r)$-cutting for set L of n lines in $\mathbb{R}^2$

partitioning of $\mathbb{R}^2$ into (possibly unbounded) triangles Δ_i such that each Δ_i intersects only n/r lines



$(1/r)$-cutting for set L of n lines in $\mathbb{R}^2$

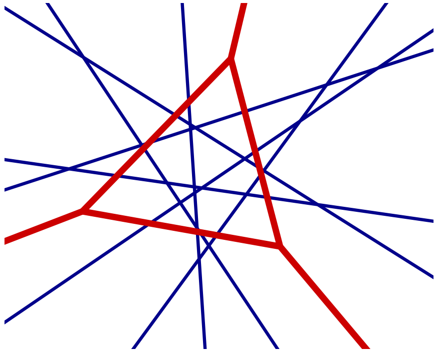
partitioning of $\mathbb{R}^2$ into (possibly unbounded) triangles Δ_i such that each Δ_i intersects only n/r lines



Theorem. For any set L of n lines in $\mathbb{R}^2$ and any r with $1 \leqslant r \leqslant n$ there is a $(1/r)$-cutting consisting of $O(r^2)$ triangles.

$(1/r)$-cutting for set L of n lines in $\mathbb{R}^2$

partitioning of $\mathbb{R}^2$ into (possibly unbounded) triangles Δ_i such that each Δ_i intersects only n/r lines



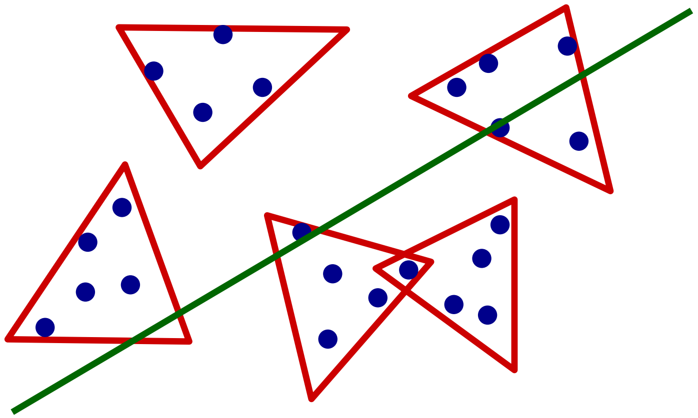
Theorem. For any set L of n lines in $\mathbb{R}^2$ and any r with $1 \leqslant r \leqslant n$ there is a $(1/r)$-cutting consisting of $O(r^2)$ triangles.

Theorem. For any set L of n hyperplanes in $\mathbb{R}^d$ and any r with $1 \leqslant r \leqslant n$ there is a $(1/r)$-cutting consisting of $O(r^d)$ simplices.

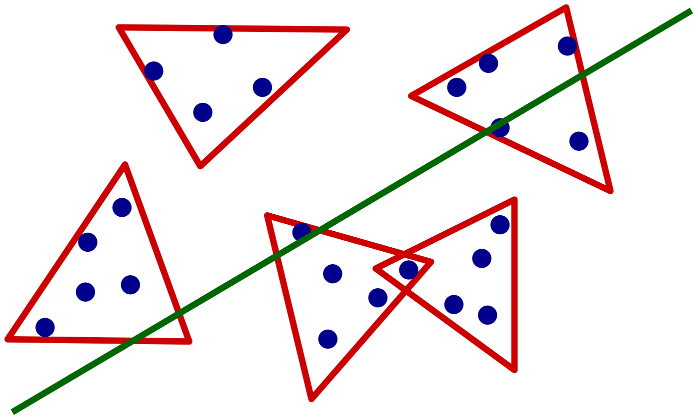
fine simplicial partition for set P of n points in $\mathbb{R}^2$

collection $\{(P_1, \Delta_1), \ldots, (P_r, \Delta_r)\}$ where

- $P = P_1 \cup \ldots \cup P_r$ and P_i's are disjoint
- Δ_i is triangle containing P_i
- $n/(2r) \leqslant |P_i| \leqslant 2n/r$



fine simplicial partition for set P of n points in $\mathbb{R}^2$



collection $\{(P_1, \Delta_1), \ldots, (P_r, \Delta_r)\}$ where

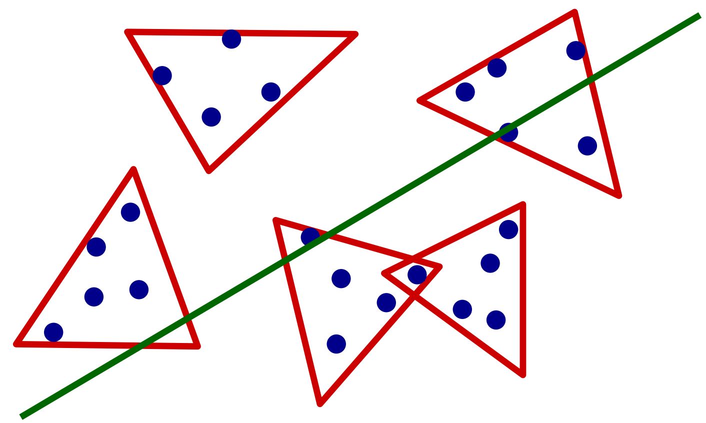
- $P = P_1 \cup \ldots \cup P_r$ and P_i's are disjoint
- Δ_i is triangle containing P_i
- $n/(2r) \leqslant |P_i| \leqslant 2n/r$

Theorem. For any set P of n points in $\mathbb{R}^2$ and any r with $1 \leqslant r \leqslant n$ there is a fine simplicial partition of $O(r)$ triangles such that any line crosses $O(\sqrt{r})$ triangles.

fine simplicial partition for set P of n points in $\mathbb{R}^2$

collection $\{(P_1, \Delta_1), \ldots, (P_r, \Delta_r)\}$ where

- $P = P_1 \cup \ldots \cup P_r$ and P_i's are disjoint
- Δ_i is triangle containing P_i
- $n/(2r) \leqslant |P_i| \leqslant 2n/r$



Theorem. For any set P of n points in $\mathbb{R}^2$ and any r with $1 \leqslant r \leqslant n$ there is a fine simplicial partition of $O(r)$ triangles such that any line crosses $O(\sqrt{r})$ triangles.

Theorem. For any set P of n points in $\mathbb{R}^d$ and any r with $1 \leqslant r \leqslant n$ there is a fine simplicial partition of $O(r)$ simplices such that any hyperplane crosses $O(r^{1-1/d})$ simplices.

Partitioning Arrangements: Basic Results

Cuttings (and simplicial partitions) form the basis of data structures for

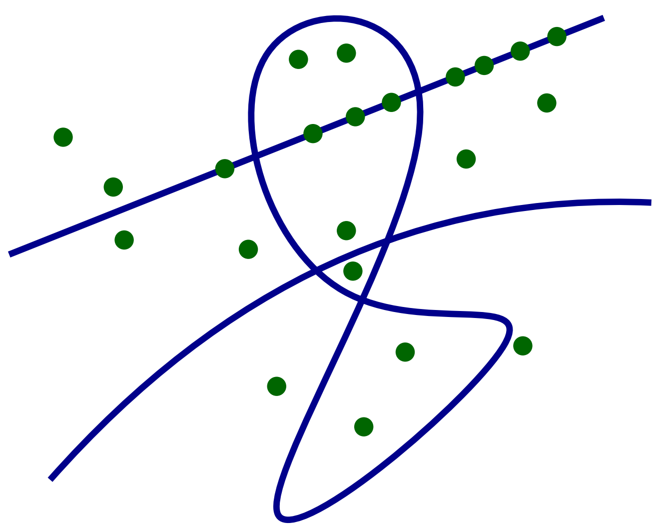
- point location
- range searching
- ray shooting

and of many other algorithmic and combinatorial results

Basic polynomial partitions [Guth-Katz'10]

- P = set of n points in $\mathbb{R}^d$
- D = parameter (can depend on n)

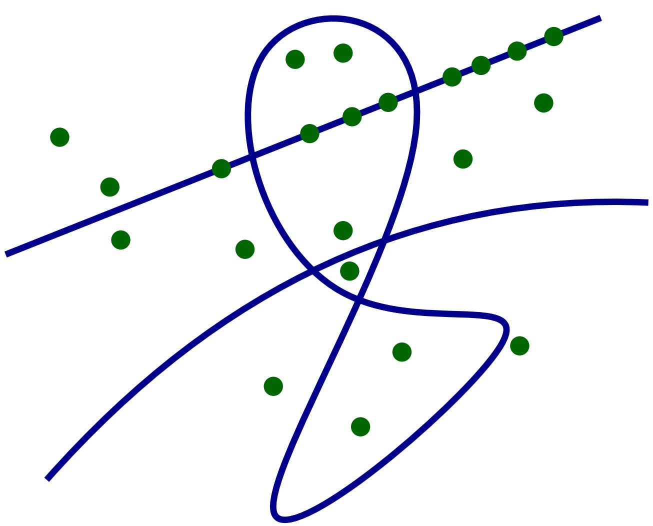
Theorem. There exists a surface $Z(f)$ that is the zero-set of a polynomial of degree at most D such that $\mathbb{R}^d \setminus Z(f)$ consists of $O(D^d)$ cells each containing $O(n/D^d)$ points from P.



Basic polynomial partitions [Guth-Katz'10]

- P = set of n points in $\mathbb{R}^d$
- D = parameter (can depend on n)

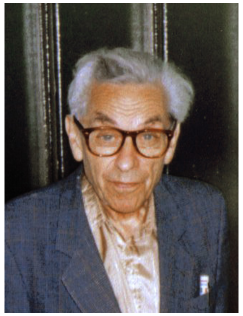
Theorem. There exists a surface $Z(f)$ that is the zero-set of a polynomial of degree at most D such that $\mathbb{R}^d \setminus Z(f)$ consists of $O(D^d)$ cells each containing $O(n/D^d)$ points from P.



Used to (basically) solve

Erdős distinct-distances problem:

any set of n points in the plane defines $\Omega(n/\log n)$ distinct distances



Generalization of polynomial partitions [Guth'15]

- L = set of n lines in $\mathbb{R}^d$
- D = parameter (can depend on n)

Theorem. There exists a surface $Z(f)$ that is the zero-set of a polynomial of degree at most D such that $\mathbb{R}^d \setminus Z(f)$ consists of $O(D^d)$ cells each intersecting $O(n/D^{d-1})$ lines from L.

Generalization of polynomial partitions [Guth'15]

- $L =$ set of n lines in $\mathbb{R}^d$
- $D =$ parameter (can depend on n)

Theorem. There exists a surface $Z(f)$ that is the zero-set of a polynomial of degree at most D such that $\mathbb{R}^d \setminus Z(f)$ consists of $O(D^d)$ cells each intersecting $O(n/D^{d-1})$ lines from L.

for lines in $\mathbb{R}^3$ we get $O(D^3)$ cells, each intersecting $O(n/D^2)$ lines

Generalization of polynomial partitions [Guth'15]

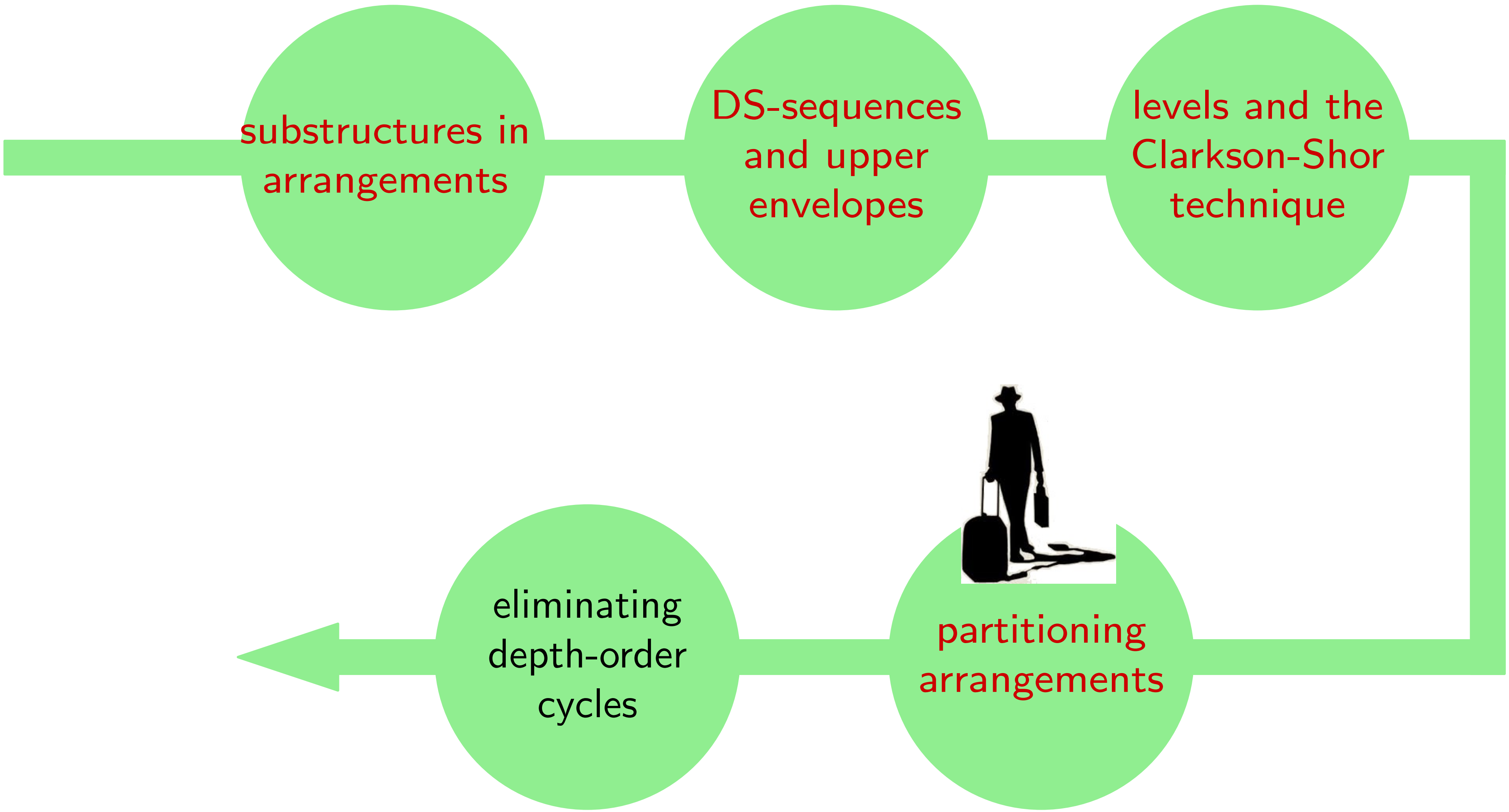
- L = set of n lines in $\mathbb{R}^d$
- D = parameter (can depend on n)

Theorem. There exists a surface $Z(f)$ that is the zero-set of a polynomial of degree at most D such that $\mathbb{R}^d \setminus Z(f)$ consists of $O(D^d)$ cells each intersecting $O(n/D^{d-1})$ lines from L.

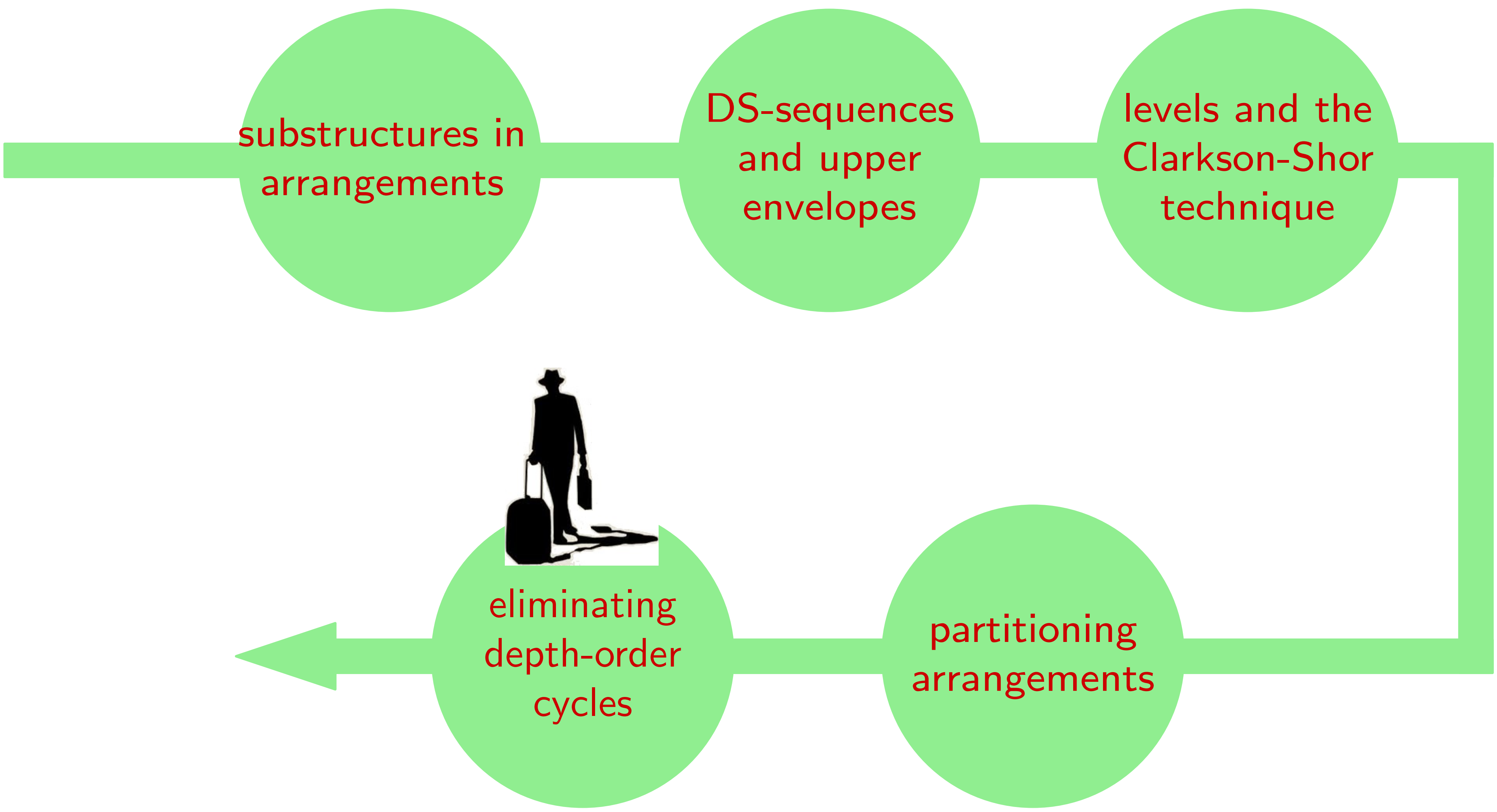
for lines in $\mathbb{R}^3$ we get $O(D^3)$ cells, each intersecting $O(n/D^2)$ lines

(result is actually even more general)

Lecture Overview



Lecture Overview

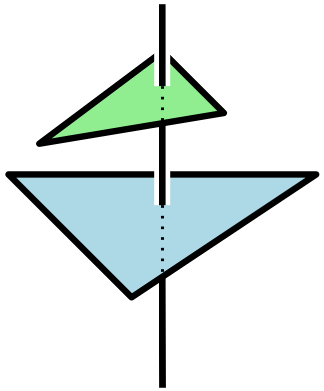


Depth orders

S: set of n disjoint triangles (or other objects) in $\mathbb{R}^3$

T is below T' (notation: $T \prec T'$):

there is a vertical line ℓ such that
$\ell \cap T$ has smaller z-coordinate than $\ell \cap T'$



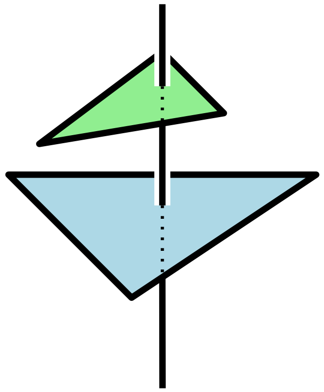
depth order on S: ordering $T_1, \ldots, T_n$ consistent with $\prec$

Depth orders

S: set of n disjoint triangles (or other objects) in $\mathbb{R}^3$

T is below T' (notation: $T \prec T'$):

there is a vertical line ℓ such that
$\ell \cap T$ has smaller z-coordinate than $\ell \cap T'$

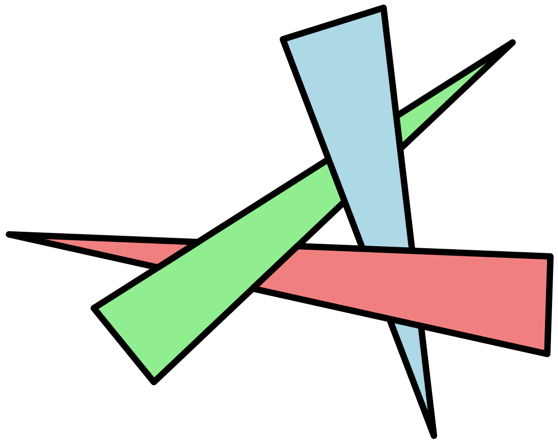


depth order on S: ordering $T_1, \ldots, T_n$ consistent with $\prec$

Applications

- computer graphics (Painter's Algorithm)
- computer-aided design and manufacturing (assembly sequences)

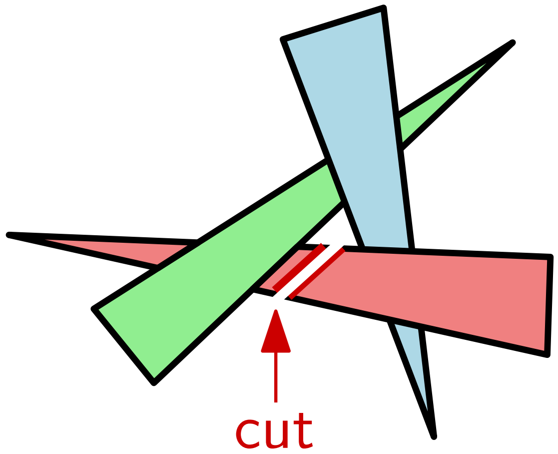
Depth order need not exist, due to cyclic overlap



Questions:

- Decide if a given order $T_1, \ldots, T_n$ is a valid depth order.
- Compute a depth order, or decide that none exists.
- How many cuts are needed, in the worst case, to eliminate all cycles?

Depth order need not exist, due to cyclic overlap

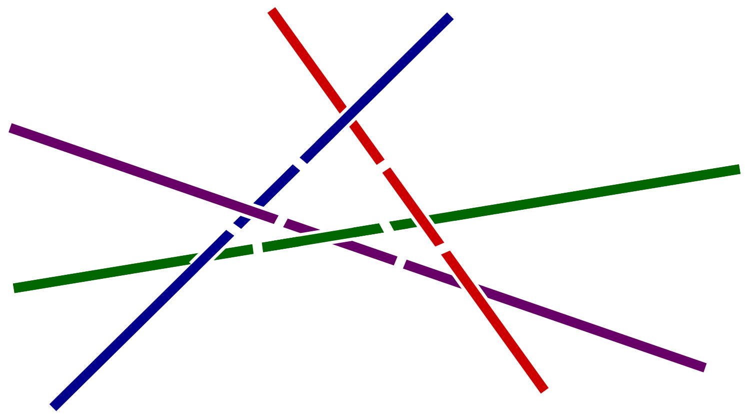


Questions:

- Decide if a given order $T_1, \ldots, T_n$ is a valid depth order.
- Compute a depth order, or decide that none exists.
- How many cuts are needed, in the worst case, to eliminate all cycles?

For line segments in $\mathbb{R}^3$

- we can eliminate all cycles using $O(n^2)$ cuts

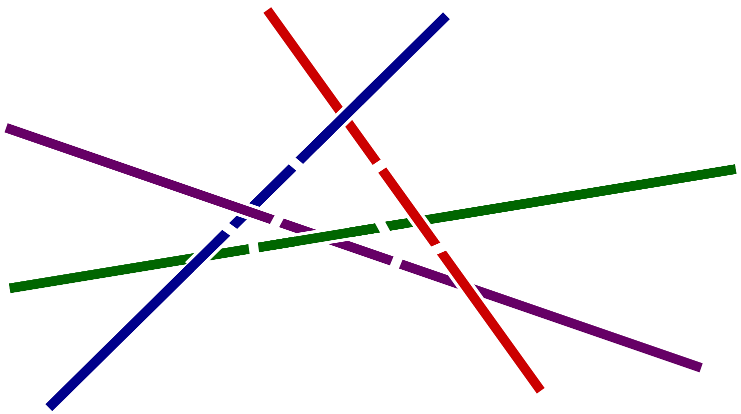


cut each line between any two adjacent intersections in projection

The number of cuts to eliminate all cycles

For line segments in $\mathbb{R}^3$

- we can eliminate all cycles using $O(n^2)$ cuts

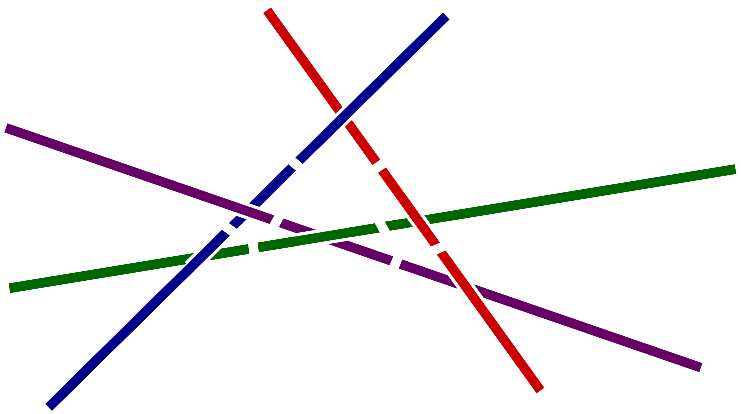


cut each line between any two adjacent intersections in projection

for triangles we can also eliminate all cycles with $O(n^2)$ cuts, with a more complicated procedure

For line segments in $\mathbb{R}^3$

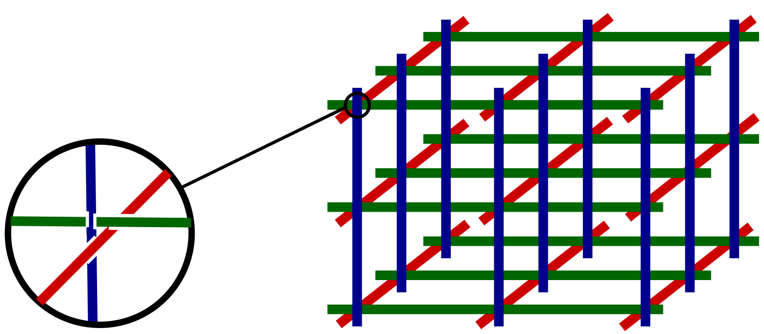
- we can eliminate all cycles using $O(n^2)$ cuts



cut each line between any two adjacent intersections in projection

for triangles we can also eliminate all cycles with $O(n^2)$ cuts, with a more complicated procedure

- in the worst case $\Omega(n^{3/2})$ cuts may be needed

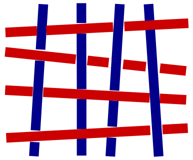


three groups of $\sqrt{n} \times \sqrt{n}$ segments each

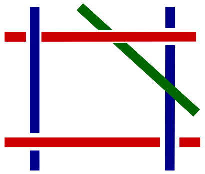
Can we do better than $O(n^2)$ cuts?

Can we do better than $O(n^2)$ cuts?

- $O(n^{9/4})$ for bipartite weavings of line segments
  [Chazelle *et al.*, FOCS'91]



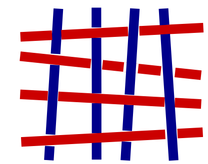
- $O(n^{2-1/69}\log^{16/69} n)$ to get rid of triangular cycles for lines
  [Aronov,Koltun,Sharir STOC'03]



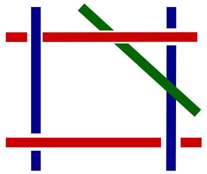
The number of cuts to eliminate all cycles

Can we do better than $O(n^2)$ cuts?

- $O(n^{9/4})$ for bipartite weavings of line segments
  [Chazelle *et al.*, FOCS'91]



- $O(n^{2-1/69}\log^{16/69} n)$ to get rid of triangular cycles for lines
  [Aronov,Koltun,Sharir STOC'03]



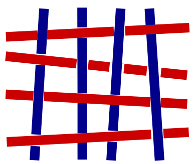
And then a breakthroughs happened

- $O(n^{3/2}\operatorname{polylog} n)$ for line segments
  [Aronov-Sharir STOC'16]
- $O(n^{3/2+\varepsilon})$ for triangles
  [Aronov-Miller-Sharir SODA'17]

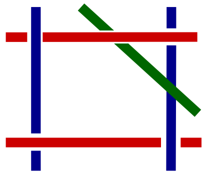
The number of cuts to eliminate all cycles

Can we do better than $O(n^2)$ cuts?

- $O(n^{9/4})$ for bipartite weavings of line segments
  [Chazelle *et al.*, FOCS'91]



- $O(n^{2-1/69} \log^{16/69} n)$ to get rid of triangular cycles for lines
  [Aronov,Koltun,Sharir STOC'03]



And then a breakthroughs happened

- $O(n^{3/2} \operatorname{polylog} n)$ for line segments
  [Aronov-Sharir STOC'16]

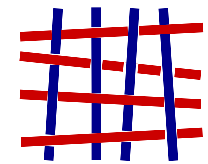
technique uses
polynomial partitions

- $O(n^{3/2+\varepsilon})$ for triangles
  [Aronov-Miller-Sharir SODA'17]

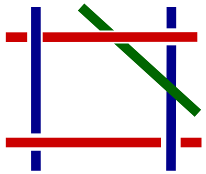
The number of cuts to eliminate all cycles

Can we do better than $O(n^2)$ cuts?

- $O(n^{9/4})$ for bipartite weavings of line segments
  [Chazelle *et al.*, FOCS'91]



- $O(n^{2-1/69} \log^{16/69} n)$ to get rid of triangular cycles for lines
  [Aronov,Koltun,Sharir STOC'03]



And then a breakthroughs happened

- $O(n^{3/2} \operatorname{polylog} n)$ for line segments
  [Aronov-Sharir STOC'16]

technique uses
polynomial partitions

- $O(n^{3/2+\varepsilon})$ for triangles
  [Aronov-Miller-Sharir SODA'17]

uses curved cuts

- $O(n^{7/4} \operatorname{polylog} n)$ for triangles,
  with straight-line cuts [dB, FOCS'18]

combines Aronov-Sharir result
with cuttings

Polynomial Partitions for Lines in $\mathbb{R}^3$

Polynomial Partitions for Lines in $\mathbb{R}^3$

- $L =$ set of n lines, or line segments, in $\mathbb{R}^3$
- $D =$ parameter

Theorem. There exists a surface $Z(f)$ that is the zero-set of a polynomial of degree at most D such that $\mathbb{R}^d \setminus Z(f)$ consists of $O(D^3)$ cells each intersecting $O(n/D^2)$ lines from L.

1. Take polynomial partition $Z(f)$ of degree D, for suitable D.
2. For each line segment $\ell \in L$ (with $\ell \not\subset Z(f)$) do the following:

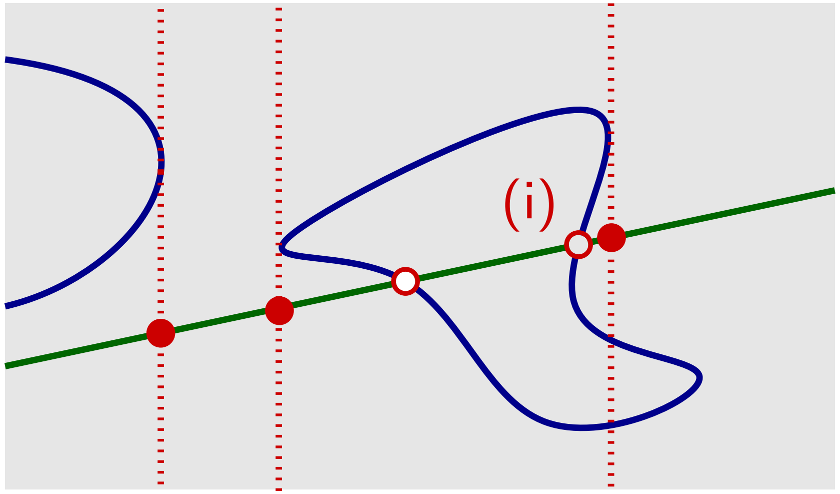
1. Take polynomial partition $Z(f)$ of degree D, for suitable D.

2. For each line segment $\ell \in L$ (with $\ell \not\subset Z(f)$) do the following:

   (i) Cut ℓ at every point where ℓ intersects $Z(f)$. $O(D)$ cuts

   (ii) Take vertical plane $h(\ell)$ containing ℓ. Cut ℓ at points below vertical tangencies of $h(\ell) \cap Z(f)$ and points below singularities.

   $O(D^2)$ cuts



○ type (i) cut

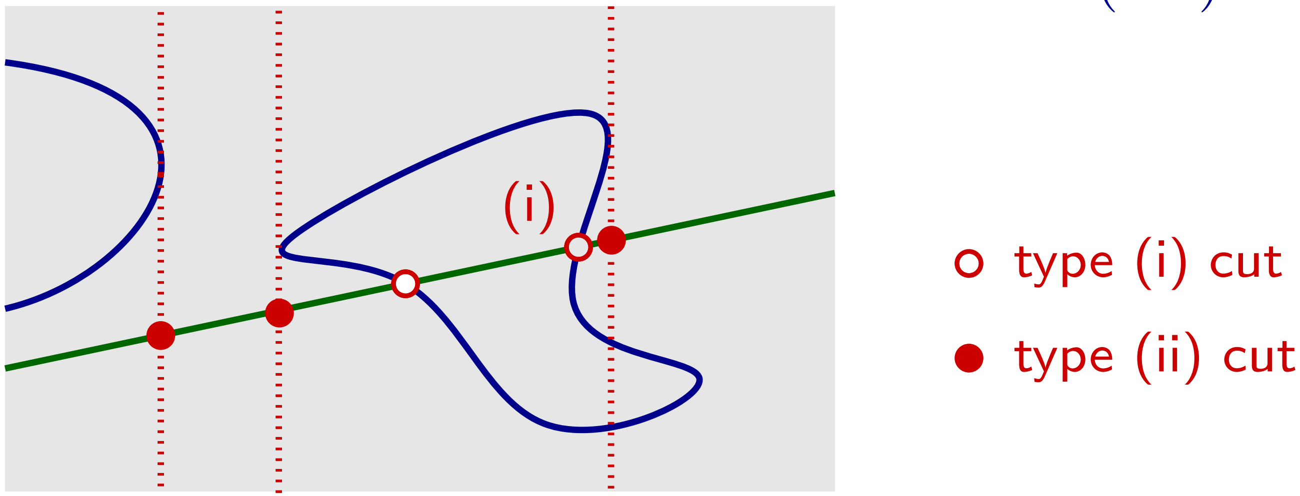
● type (ii) cut

1. Take polynomial partition $Z(f)$ of degree D, for suitable D.

2. For each line segment $\ell \in L$ (with $\ell \not\subset Z(f)$) do the following:

   (i) Cut ℓ at every point where ℓ intersects $Z(f)$. $O(D)$ cuts

   (ii) Take vertical plane $h(\ell)$ containing ℓ. Cut ℓ at points below vertical tangencies of $h(\ell) \cap Z(f)$ and points below singularities. $O(D^2)$ cuts

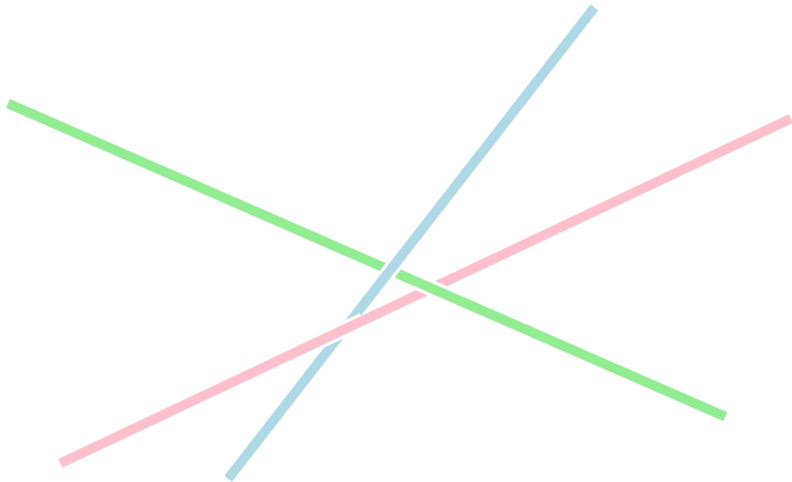


3. Recursively cut lines within each cell.

Lemma. Procedure eliminates all cycles.

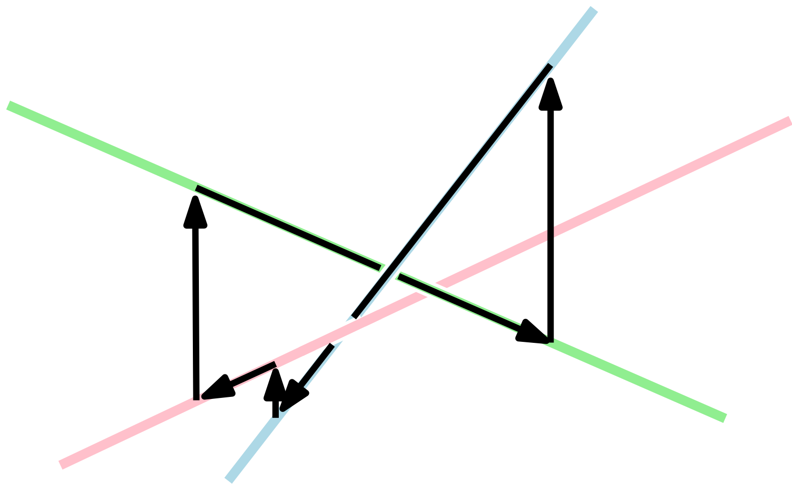
Lemma. Procedure eliminates all cycles.

Proof. Suppose for a contradiction that there is still a cycle



Lemma. Procedure eliminates all cycles.

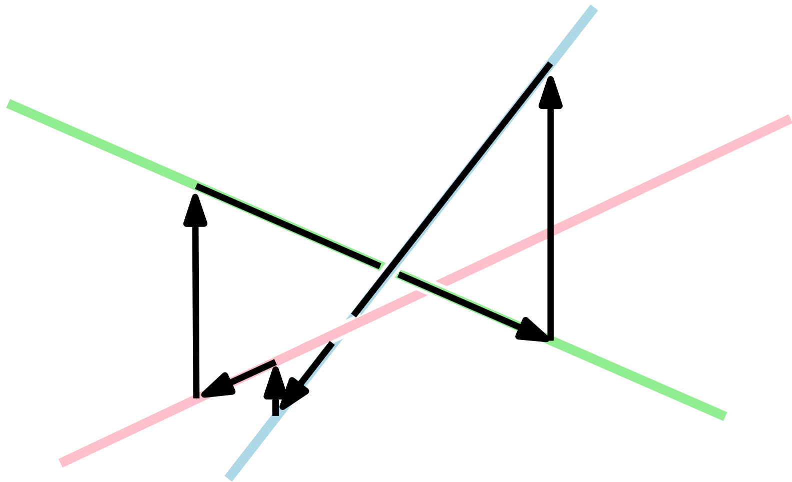
Proof. Suppose for a contradiction that there is still a cycle



- associate 3D polygonal curve Γ to cycle

Lemma. Procedure eliminates all cycles.

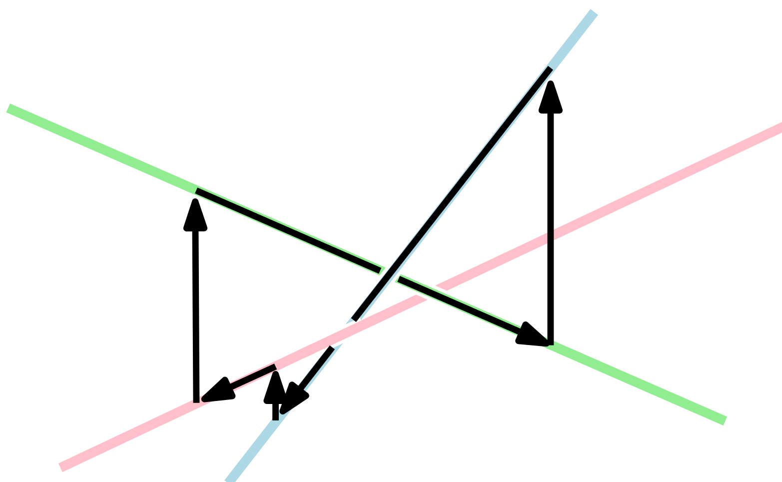
Proof. Suppose for a contradiction that there is still a cycle



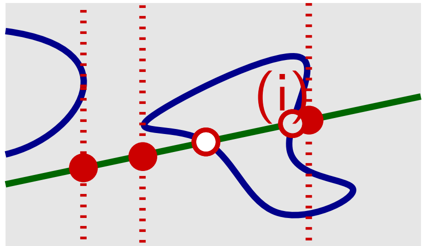
- associate 3D polygonal curve Γ to cycle
- if Γ lies completely inside cell of $\mathbb{R}^3 \setminus Z(f)$ then cycles are removed by induction

Lemma. Procedure eliminates all cycles.

Proof. Suppose for a contradiction that there is still a cycle



- associate 3D polygonal curve Γ to cycle
- if Γ lies completely inside cell of $\mathbb{R}^3 \setminus Z(f)$ then cycles are removed by induction
- otherwise consider how level (number of intersection of upward ray with $Z(f)$) changes as we walk along Γ



The Aronov-Sharir approach for line segments

Lemma. Procedure makes $O(n^{3/2} \operatorname{polylog} n)$ cuts.

The Aronov-Sharir approach for line segments

Lemma. Procedure makes $O(n^{3/2} \text{ polylog } n)$ cuts.

Proof. $$C(n) = O(D^3) \cdot C(n/D^2) + O(nD^2)$$

$$D := \Theta(n^{1/4}) \quad \Longrightarrow \quad C(n) = O(n^{3/2} \text{ polylog } n) \qquad \blacksquare$$

Lemma. Procedure makes $O(n^{3/2} \text{ polylog } n)$ cuts.

Proof.

$$C(n) = O(D^3) \cdot C(n/D^2) + O(nD^2)$$

$$D := \Theta(n^{1/4}) \quad \Longrightarrow \quad C(n) = O(n^{3/2} \text{ polylog } n) \qquad \blacksquare$$

- similar but much more complicated approach works for triangles
- triangles are cut by polynomial, so pieces have curved boundaries

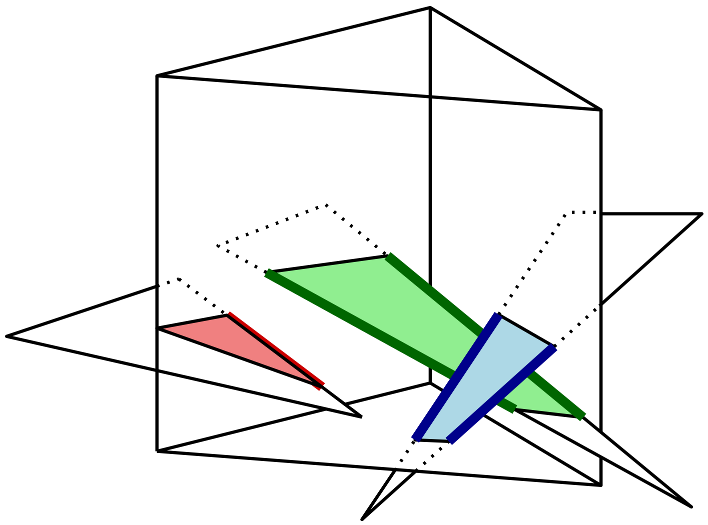
Avoiding curved cuts for triangles

Key idea: Relate depth order for triangles to depth order of its edges

Avoiding curved cuts for triangles

Key idea: Relate depth order for triangles to depth order of its edges

- $\mathcal{T}$ = set of n triangles in $\mathbb{R}^3$
- $\mathcal{E}$ = set of $3n$ edges of the triangles
- C = vertical, triangular column



Lemma. If C does not contain any triangle vertex in its interior and $\mathcal{E} \cap C$ is acyclic, then $\mathcal{T} \cap C$ is acyclic.

Avoiding curved cuts for triangles

- $\mathcal{T}$ = set of n triangles in $\mathbb{R}^3$, $\mathcal{E}$ = set of $3n$ triangle edges

Lemma. If C does not contain any triangle vertex in its interior and $\mathcal{E} \cap C$ is acyclic, then $\mathcal{T} \cap C$ is acyclic.

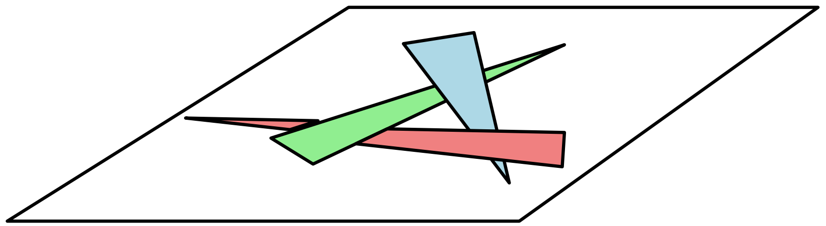
Avoiding curved cuts for triangles

- $\mathcal{T}$ = set of n triangles in $\mathbb{R}^3$, $\mathcal{E}$ = set of $3n$ triangle edges

Lemma. If C does not contain any triangle vertex in its interior and $\mathcal{E} \cap C$ is acyclic, then $\mathcal{T} \cap C$ is acyclic.

How to use the lemma (failed approach)

1. Compute complete cut set X for $\mathcal{E}$ of size of $O(n^{3/2} \operatorname{polylog} n)$.

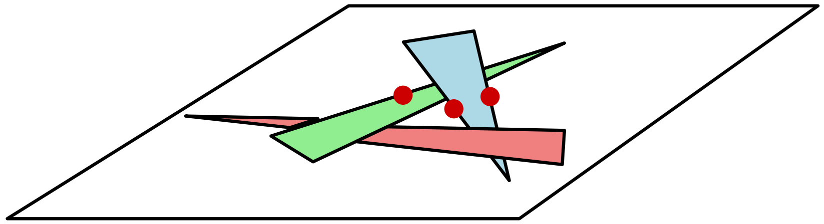


- $\mathcal{T}$ = set of n triangles in $\mathbb{R}^3$, $\mathcal{E}$ = set of $3n$ triangle edges

Lemma. If C does not contain any triangle vertex in its interior and $\mathcal{E} \cap C$ is acyclic, then $\mathcal{T} \cap C$ is acyclic.

How to use the lemma (failed approach)

1. Compute complete cut set X for $\mathcal{E}$ of size of $O(n^{3/2} \operatorname{polylog} n)$.

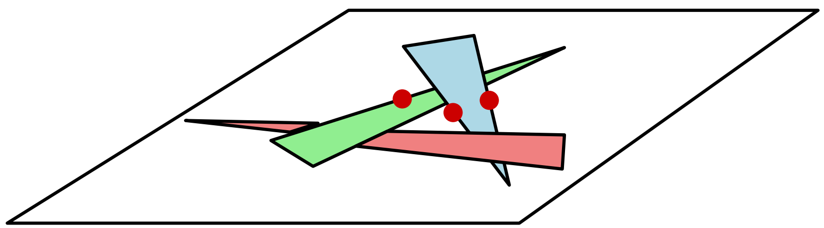


- $\mathcal{T}$ = set of n triangles in $\mathbb{R}^3$, $\mathcal{E}$ = set of $3n$ triangle edges

Lemma. If C does not contain any triangle vertex in its interior and $\mathcal{E} \cap C$ is acyclic, then $\mathcal{T} \cap C$ is acyclic.

Trivial approach:

1. Compute complete cut set X for $\mathcal{E}$ of size of $O(n^{3/2} \operatorname{polylog} n)$.
2. $P :=$ {triangle vertices} $\cup X$.



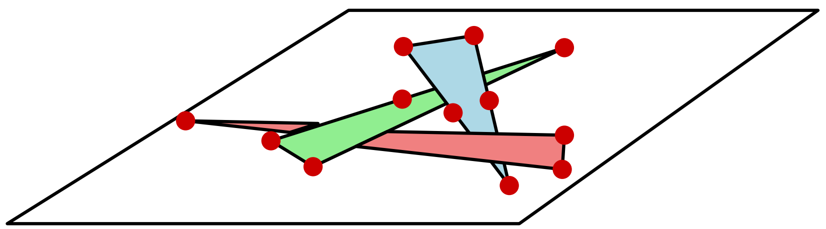
Avoiding curved cuts for triangles

- $\mathcal{T}$ = set of n triangles in $\mathbb{R}^3$, $\mathcal{E}$ = set of $3n$ triangle edges

Lemma. If C does not contain any triangle vertex in its interior and $\mathcal{E} \cap C$ is acyclic, then $\mathcal{T} \cap C$ is acyclic.

Trivial approach:

1. Compute complete cut set X for $\mathcal{E}$ of size of $O(n^{3/2} \operatorname{polylog} n)$.
2. $P := \{$triangle vertices$\} \cup X$.

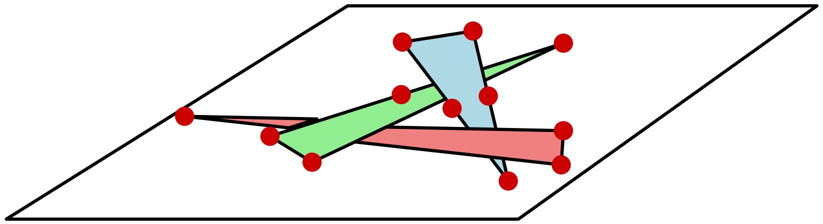


- $\mathcal{T}$ = set of n triangles in $\mathbb{R}^3$, $\mathcal{E}$ = set of $3n$ triangle edges

Lemma. If C does not contain any triangle vertex in its interior and $\mathcal{E} \cap C$ is acyclic, then $\mathcal{T} \cap C$ is acyclic.

Trivial approach:

1. Compute complete cut set X for $\mathcal{E}$ of size of $O(n^{3/2} \operatorname{polylog} n)$.
2. $P :=$ {triangle vertices} $\cup X$.
   $h_p :=$ vertical plane through p
   $H_P := \{h_p : p \in P\}$.

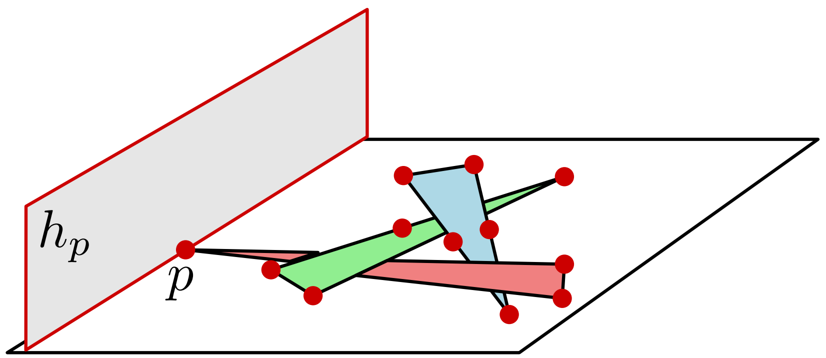


- $\mathcal{T}$ = set of n triangles in $\mathbb{R}^3$, $\mathcal{E}$ = set of $3n$ triangle edges

Lemma. If C does not contain any triangle vertex in its interior and $\mathcal{E} \cap C$ is acyclic, then $\mathcal{T} \cap C$ is acyclic.

Trivial approach:

1. Compute complete cut set X for $\mathcal{E}$ of size of $O(n^{3/2}\,\mathrm{polylog}\, n)$.
2. $P := \{$triangle vertices$\} \cup X$.
   $h_p :=$ vertical plane through p
   $H_P := \{h_p : p \in P\}$.

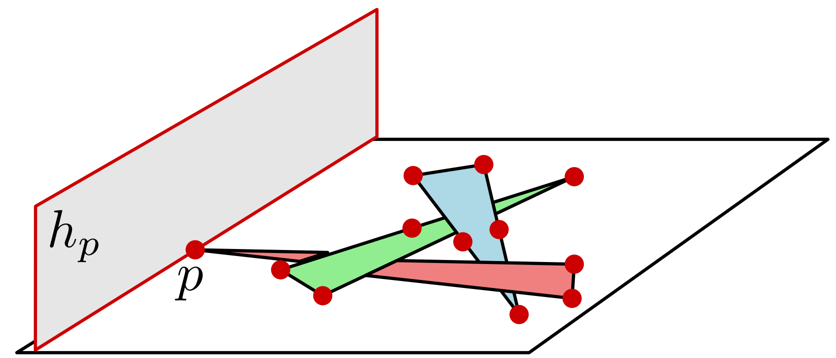


- $\mathcal{T}$ = set of n triangles in $\mathbb{R}^3$, $\mathcal{E}$ = set of $3n$ triangle edges

Lemma. If C does not contain any triangle vertex in its interior and $\mathcal{E} \cap C$ is acyclic, then $\mathcal{T} \cap C$ is acyclic.

Trivial approach:

1. Compute complete cut set X for $\mathcal{E}$ of size of $O(n^{3/2} \operatorname{polylog} n)$.
2. $P :=$ {triangle vertices} $\cup X$.
   $h_p :=$ vertical plane through p
   $H_P := \{h_p : p \in P\}$.
3. Cut all triangles with planes in H_p
   $\Longrightarrow$ Removes all cycles

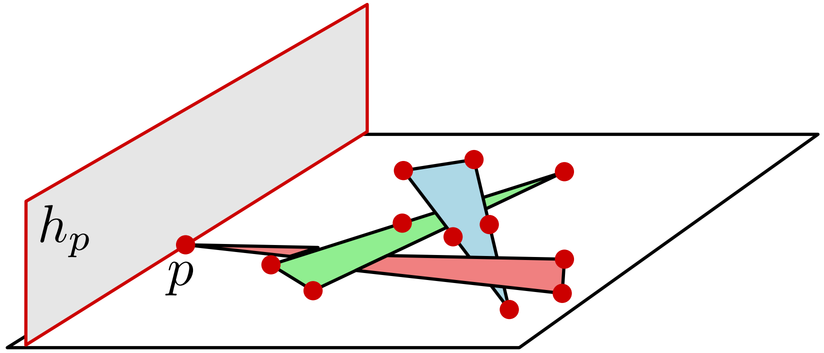


- $\mathcal{T}$ = set of n triangles in $\mathbb{R}^3$, $\mathcal{E}$ = set of $3n$ triangle edges

Lemma. If C does not contain any triangle vertex in its interior and $\mathcal{E} \cap C$ is acyclic, then $\mathcal{T} \cap C$ is acyclic.

Trivial approach:

1. Compute complete cut set X for $\mathcal{E}$ of size of $O(n^{3/2} \operatorname{polylog} n)$.
2. $P :=$ {triangle vertices} $\cup X$.
   $h_p :=$ vertical plane through p
   $H_P := \{h_p : p \in P\}$.
3. Cut all triangles with planes in H_p
   $\Longrightarrow$ Removes all cycles



but may result in $O(n^{2.5} \operatorname{polylog} n)$ fragments.

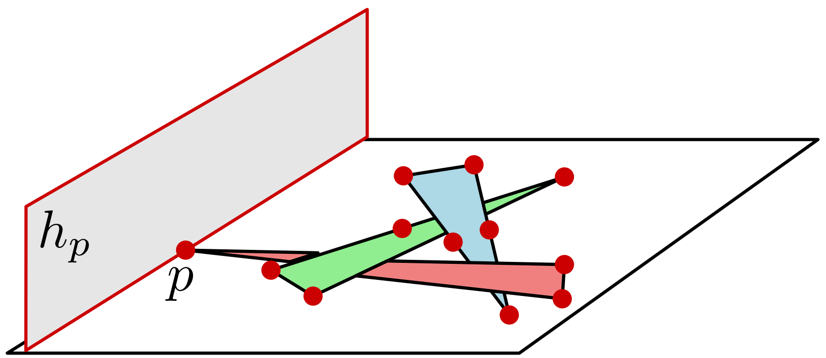
Avoiding curved cuts for triangles

- $\mathcal{T}$ = set of n triangles in $\mathbb{R}^3$, $\mathcal{E}$ = set of $3n$ triangle edges

Lemma. If C does not contain any triangle vertex in its interior and $\mathcal{E} \cap C$ is acyclic, then $\mathcal{T} \cap C$ is acyclic.

Trivial approach:

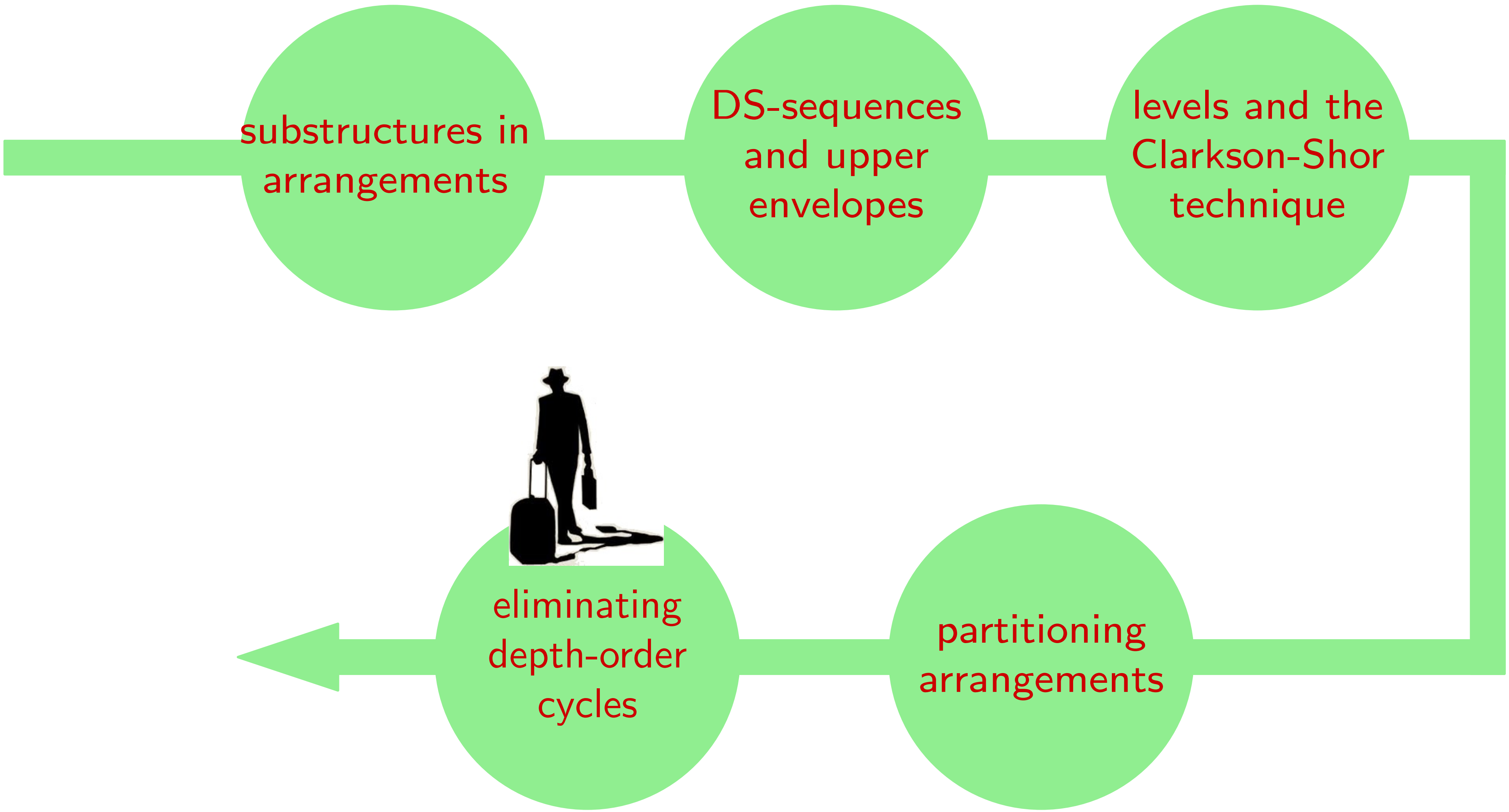
1. Compute complete cut set X for $\mathcal{E}$ of size of $O(n^{3/2} \operatorname{polylog} n)$.
2. $P :=$ {triangle vertices} $\cup X$.
   $h_p :=$ vertical plane through p
   $H_P := \{h_p : p \in P\}$.
3. Cut all triangles with planes in H_p
   $\Longrightarrow$ Removes all cycles



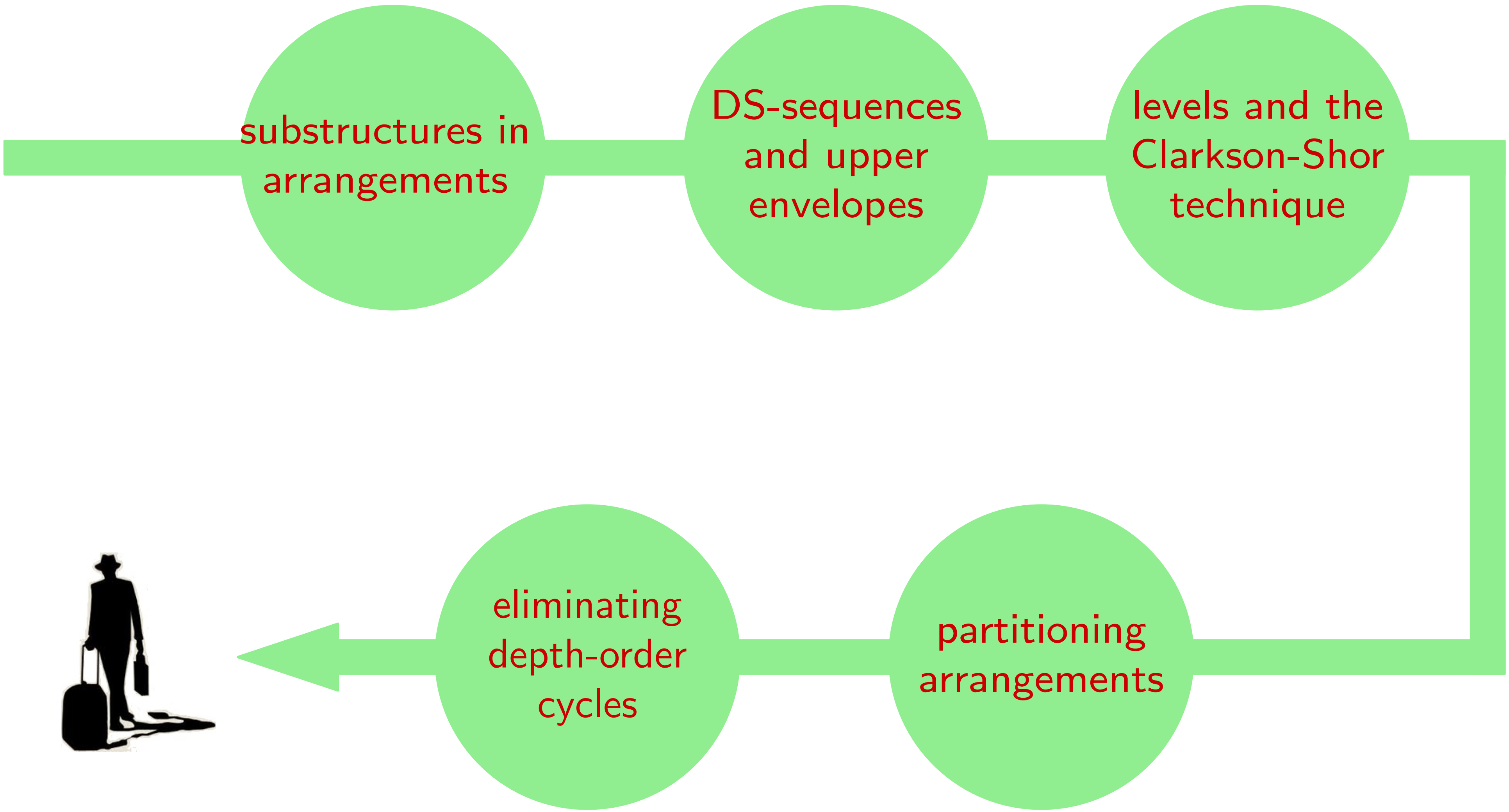
but may result in $O(n^{2.5} \operatorname{polylog} n)$ fragments.

using hierarchical gives $O(n^{7/4} \operatorname{polylog} n)$ fragments.

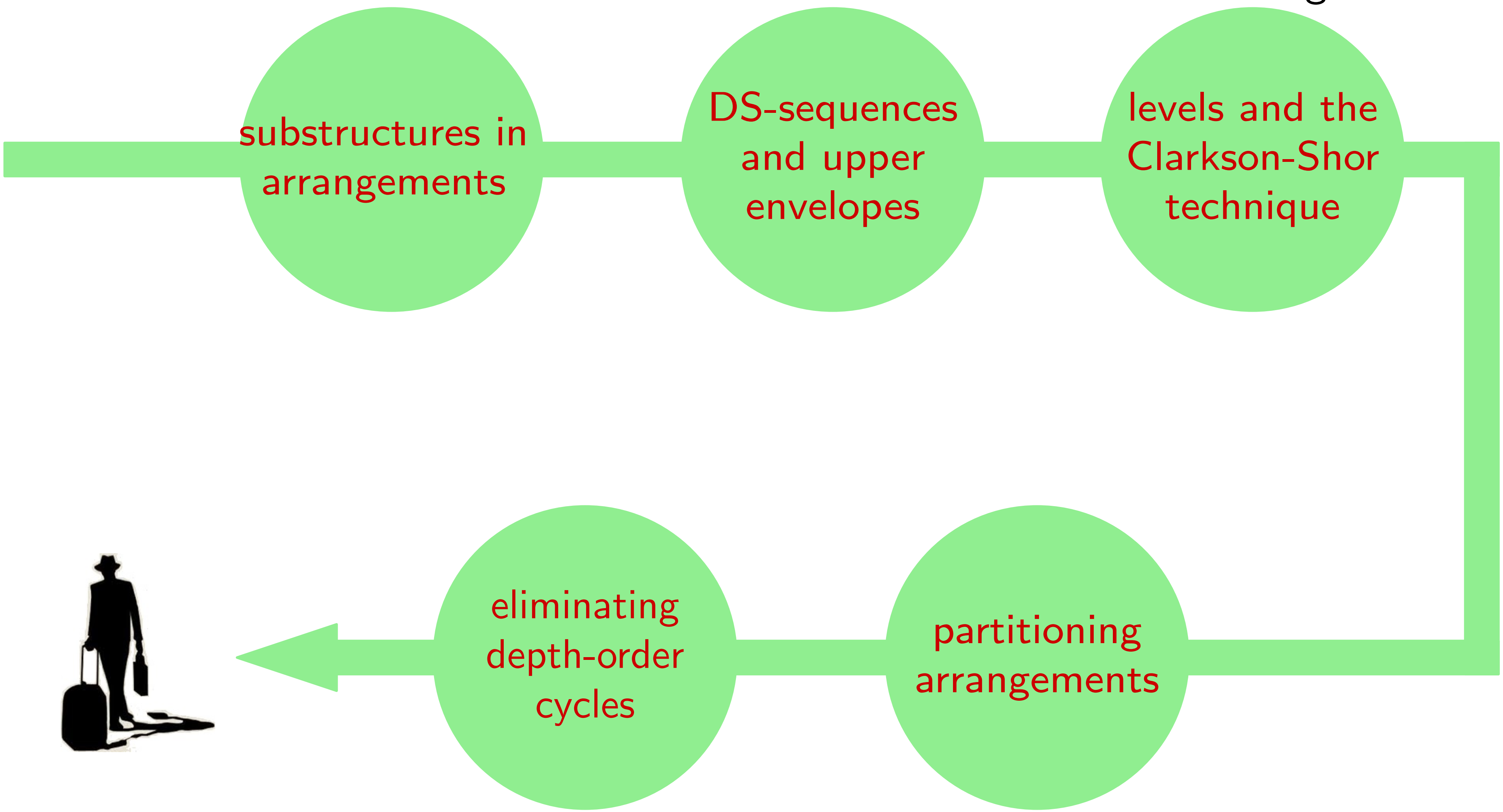
Lecture Overview



Lecture Overview

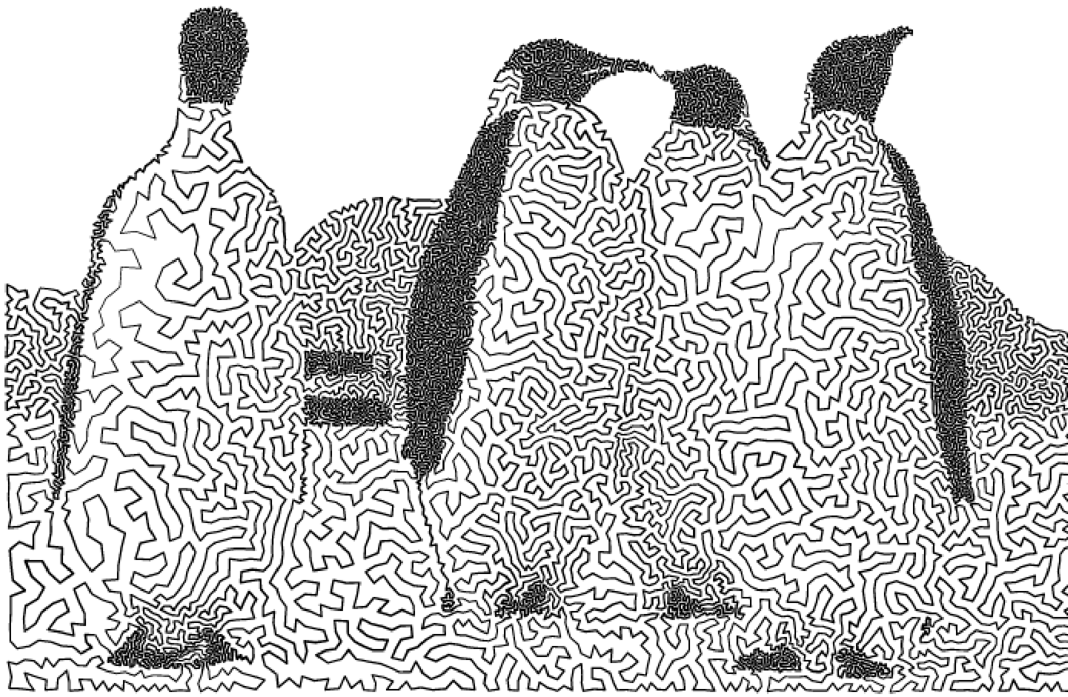


open problem:
complexity of k-level
in line arrangement



open problem:
eliminate cycles in triangles with
$o(n^{7/4})$ straight-line cuts

Thanks for your attention!



TSP Art by Carig Kaplan and Robert Bosch