OI-OPPA. European Social Fund
Prague & EU: We invest in your future.

DCGI

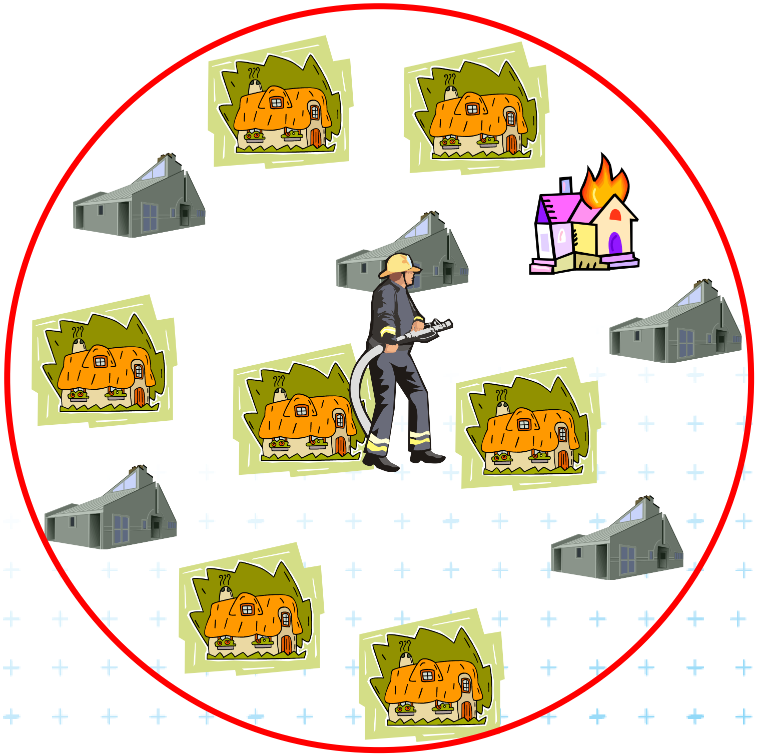
KATEDRA POČÍTAČOVÉ GRAFIKY A INTERAKCE

Smallest enclosing circle

Jan Volný - 26.10.2012

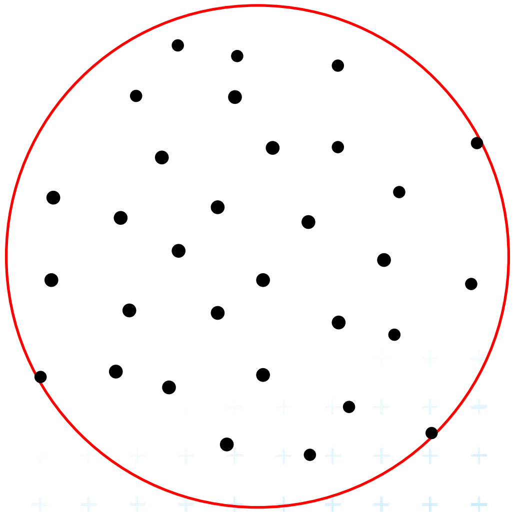
Overview

- Smallest enclosing circle (SEC) problem
- Applications of SEC
- Brute-force algorithm
- Faster algorithms
- Summary



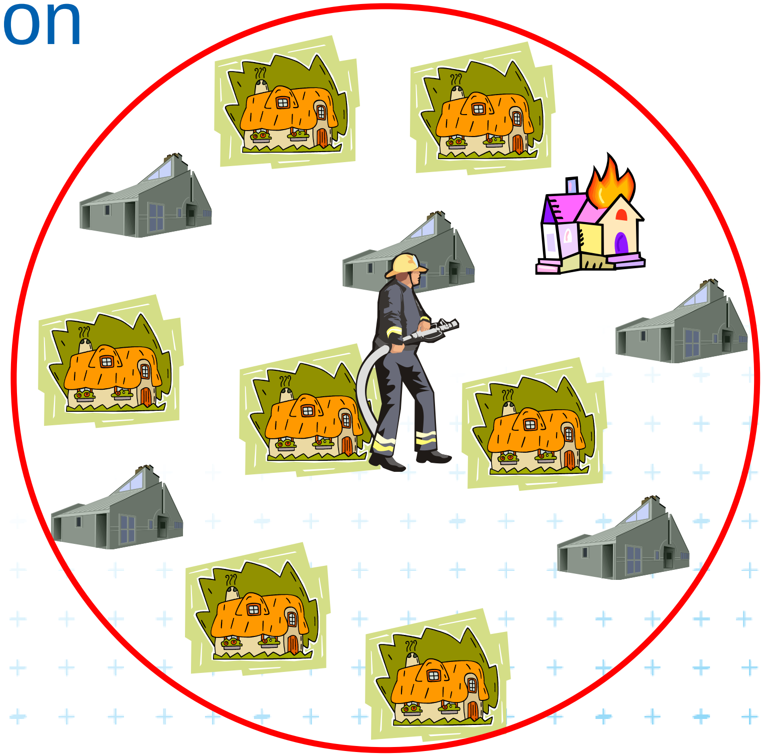
SEC problem

- Let us assume a set of points in a plane. The smallest enclosing circle (minimum enclosing circle) is such a minimal circle that covers all these points



Applications

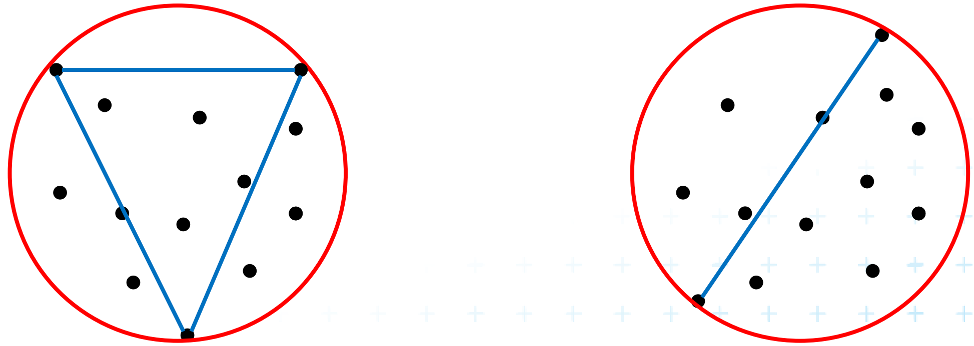
- Facility location problem
- Bomb Problem
- Radio transmitter position



Basic principle

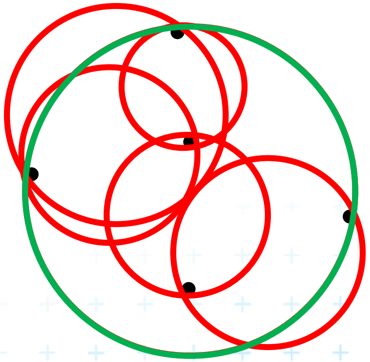
- The smallest enclosing circle is unique and:
  - is either circumcircle of some (at least) three points

    OR
  - is defined by two points as a diameter



Brute-force algorithm

- Makes circles of all pairs and triplets of all given points of the set
  - Finds the smallest circle, which covers all points
  - $O(n^4)$
- Very slow method



Faster algorithms

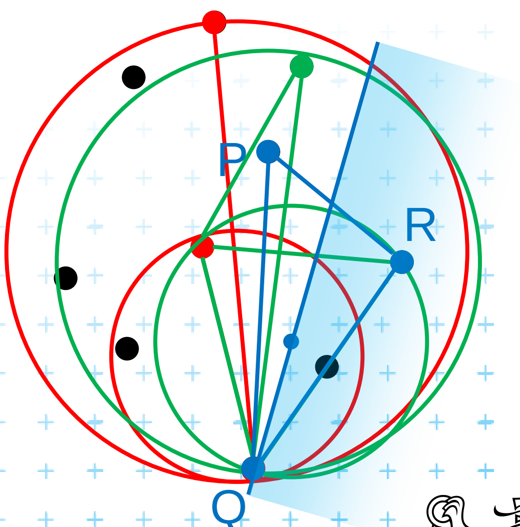
- Based on the minimization of the maximal distance from the center of the circle

$$\min_{p_0} \max_{i} \ (x_i - x_0)^2 + (y_i - y_0)^2$$

- Elzinga and Hearn (1972)
  - $O(n^2)$
- Shamos and Hoey (1977)
  - O(n·log(n))
- Nimrod Megiddo (1983)
  - O(n)

Elzinga & Hearn

1. Pick any 2 points of the set
2. Let them make a diameter of a circle
   if the circle covers all points ➡ STOP
   else choose the third point and go to step 3
3. if the triangle is *right* or *obtuse*
   drop the point at the angle $\geq 90°$, go to step 2
   else go to step 4
4. if the circle covers all points ➡ STOP
   else choose 1 point (*P*) out of circle, get the farthest vertex (*Q*), extend the diameter through this vertex, choose the vertex (*R*) that is in the half plane opposite to the point, go to step 3

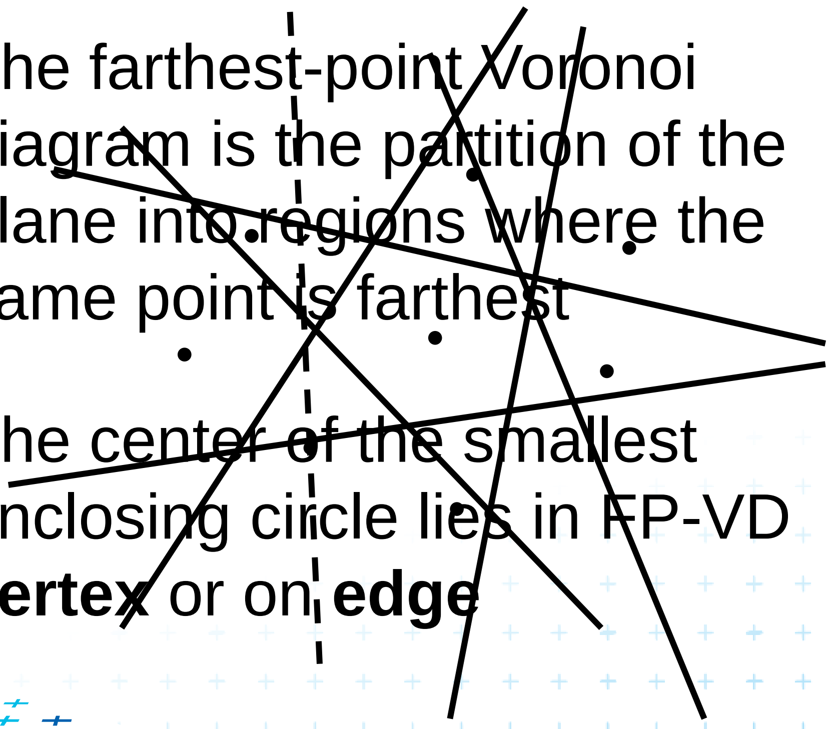


Elzinga & Hearn - summary

- Improved method of the brute force algorithm
- Increasing radius of the circle makes the algorithm finite
- Complexity $O(n^2)$

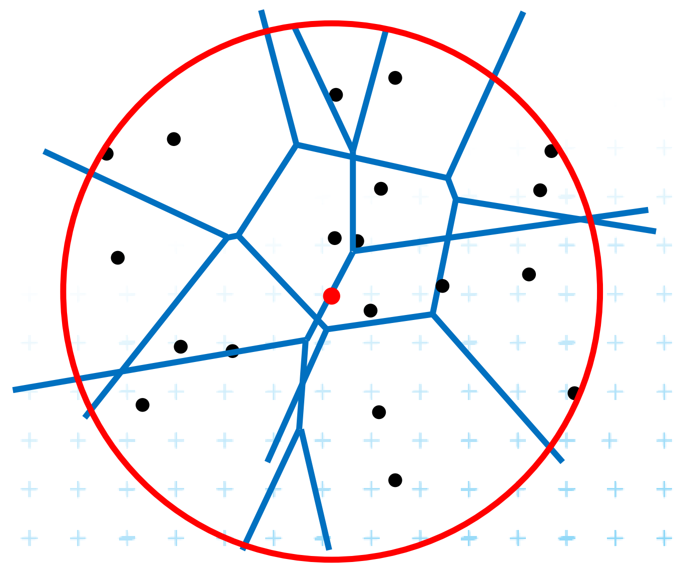
Shamos & Hoey

- Algorithm using the Voronoi diagram
- Voronoi diagram describes the nearest area of each point
- The center of the smallest enclosing circle must be a vertex of the farthest-point Voronoi diagram



The farthest-point Voronoi diagram is the partition of the plane into regions where the same point is farthest

The center of the smallest enclosing circle lies in FP-VD **vertex** or on **edge**



Shamos & Hoey

- The construction of Farthest-point Voronoi diagram of the set of points
  - O(n·log(n))
- Finding the center of the circle
  - O(n)
- The final complexity
  - O(n·log(n)+n) ⇒ O(n·log(n))

Nimrod Megiddo

- Algorithm using linear programming for minimization problems
- Prune and search method
- Works in linear time
- In each step it reduces the input size by a constant fraction *1/f*
- Uses methods *median()*, *MEC-center()* for pruning
- Then the time is $O(n)*(1+(1-f)+(1-f)^2+\ldots) \Longrightarrow O(f \cdot n)$

Summary

- Problem of minimax
- The naïve algorithm works in $O(n^4)$, with the improvement in $O(n^2)$
- The best algorithms can be linear

- If you can implement it in linear time…

… just do it

source: wordpress.com

References

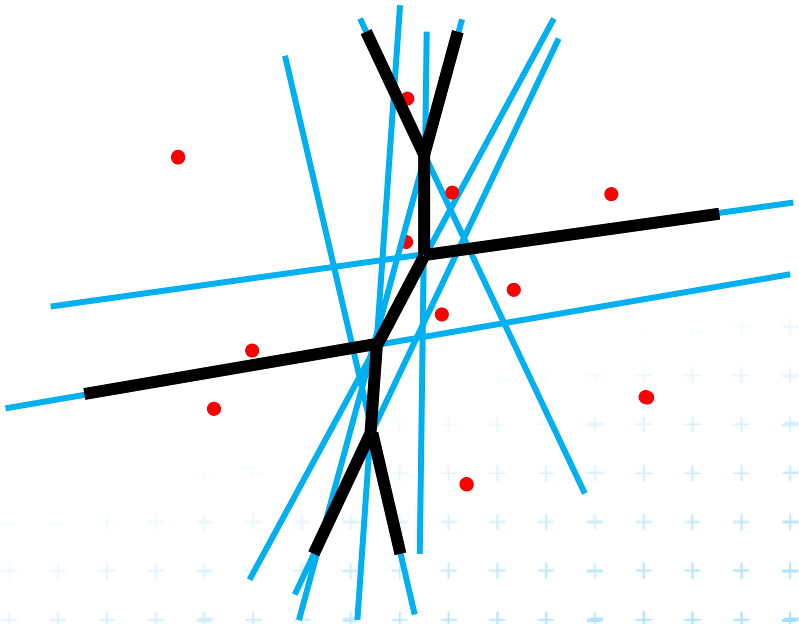
- F. P. Preparata, M. I. Shamos, „Computational Geometry – An Introduction“, *Springer Verlag*, 2nd edition, 1985
- J. Helzinga, D. W. Hearn, „Geometrical Solutions for Some Minimax Location Problems“, *Transportation Science*, 1972
- I. M. Shamos, D. Hoey, „Closest-point problems“, *Foundations of Computer Science*, 1975
- X. Sheng, R. M. Freund, J. Sun, „Solution Methodologies for the Smallest Enclosing Circle Problem“, *Computational Optimization and Applications*, 2003
- N. Megiddo, „Linear-Time Algorithms for Linear Programming in R^3 and Related Problems“, *Society for Industrial and Applied Mathematics*, 1983

Thank you for your attention

Jan Volný, 26.10.2012

How to build FP VD

- Higher order VD – the cell describes the nearest area to the set of points
- FP VD is the VD of the *(n-1)*-order



[Taken from Utrecht University]

Elzinga & Hearn - proof

- The improvement of the brute-force algorithm is based on finding the 2 farthest points of the set
- Finding the farthest two points requires computing $(m^2 - m)/2$ distances
- That gives us the complexity $O(n^2)$

Shamos & Hoey - proof

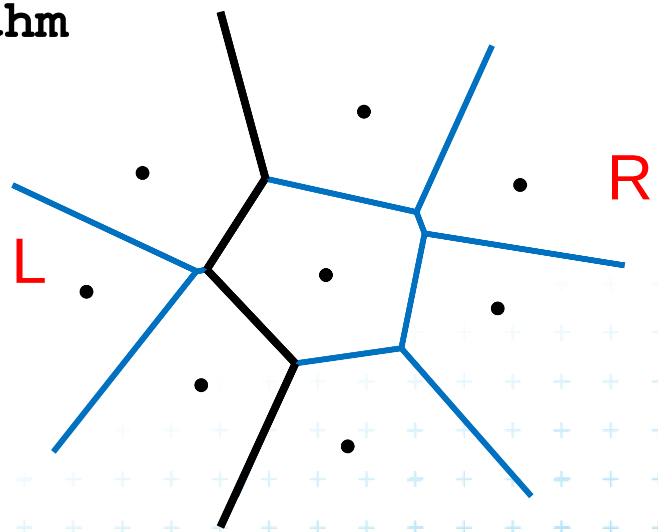
- The farthest-point Voronoi diagram is built in *O(n·log(n))* and has *O(n)* edges and vertices
- We find two farthest points in *O(n)*
- If the circle determined by these 2 points encloses all the points, we are done
- Otherwise the center is a vertex of the FP VD (there are at most *n* vertices, so all the circumradii can be found in *O(n)*)

Nimrod Megiddo - pseudocode

- Arbitrarily pair up the n points in S to get n/2 pairs
- Construct a bisecting line for each pair of points, to get n/2 bisectors
- Call *median()* to find the bisector with median slope. Call this slope m_{mid}
- Pair up each bisector of slope ≥ m_{mid} with another of slope < m_{mid}, to get n/4 intersection points. Call the set I
- Call *median()* to find the point in I with median y-value. Call this y-value y_{mid}
- Call *MEC-center()* to find which side of the line $y=y_{mid}$ the MEC-center C lies on. (Without loss of generality, suppose it lies above.)
- Let I' be the subset of points of I whose y-values are less than y_{mid}. (I' contains n/8 points.)
- Find a line L with slope m_{mid} such that half the points in I' lie to L's left, half to its right.
- Call *MEC-center()* on L. Without loss of generality, suppose C lies on L's right.
- Let I'' be the subset of I' whose points lie to the left of L. (I'' contains n/16 points.)

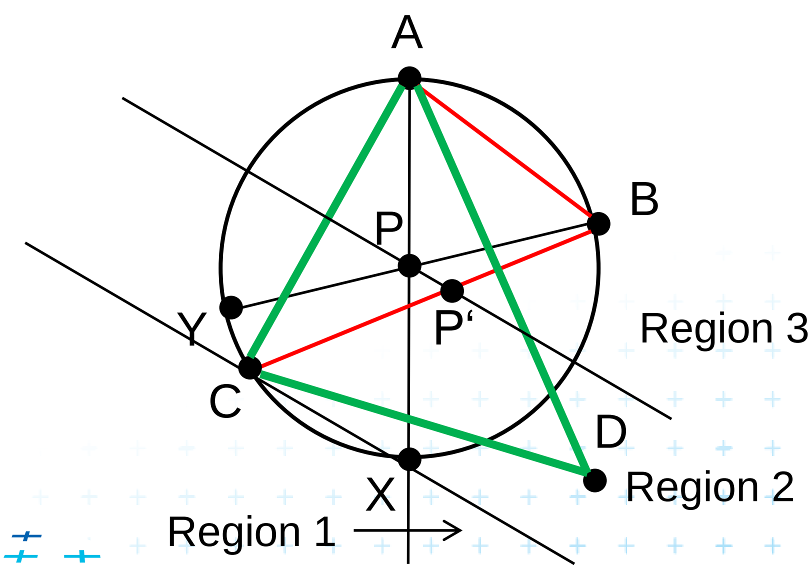
Voronoi diagram – proof

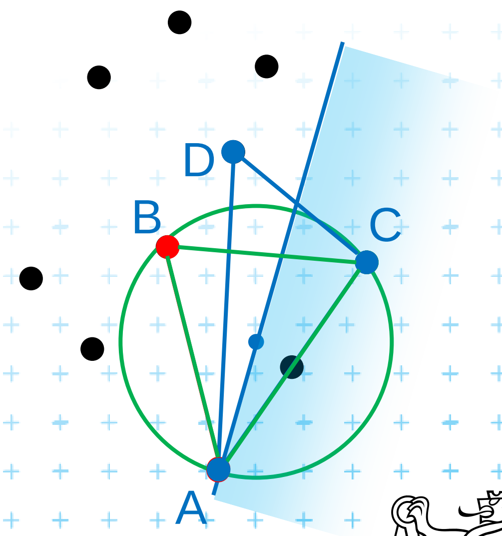
- Suppose that the set S of *n* points is divided into two subsets L and R, each containing *n/2* points
- Assume that we already possess the Voronoi diagrams *V(L)* and *V(R)* of L and R separately
- If these can be merged in linear time to form the diagram *V(S)* of the entire set, then splitting the problem recursively will give an *O(N log N)* algorithm
- Merge:



Elzinga & Hearn - update

- In the last step of the algorithm we choose the opposite vertex to the point P
- Proof: In Region 2 the angle by C is acute, otherwise it is obtuse ⟹ same half-plane doesn‘t contain any Region 2





OI-OPPA. European Social Fund
Prague & EU: We invest in your future.