Delaunay Triangulations with Predictions

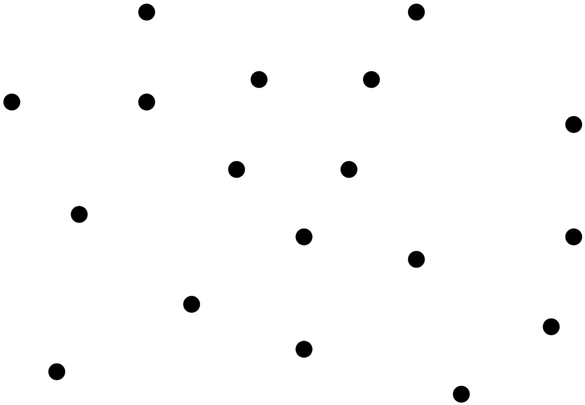


Sergio Cabello — UNIVERZA V LJUBLJANI / University of Ljubljana
Timothy M. Chan — UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN
Panos Giannopoulos — CITY ST GEORGE'S UNIVERSITY OF LONDON

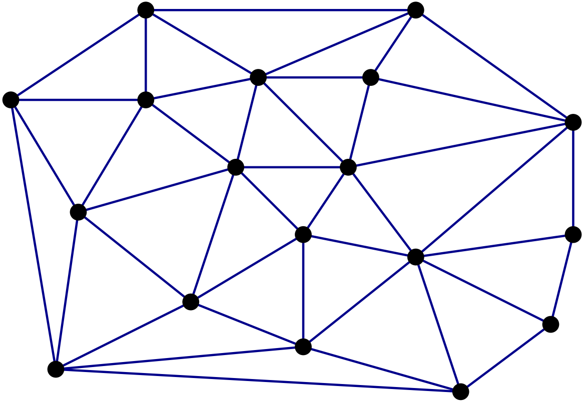
17th Innovations in Theoretical Computer Science (ITCS), 2026

KARST
erc European Research Council
aris Slovenian Research and Innovation Agency

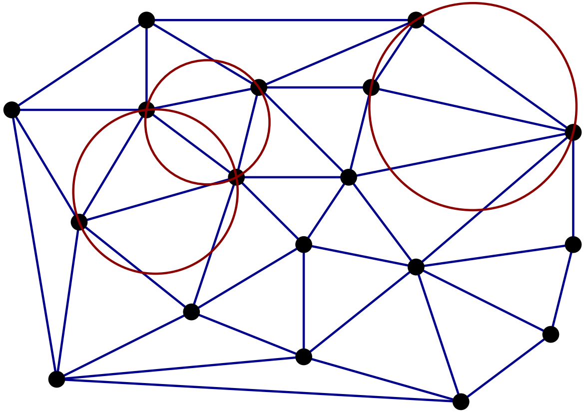
Delaunay triangulations



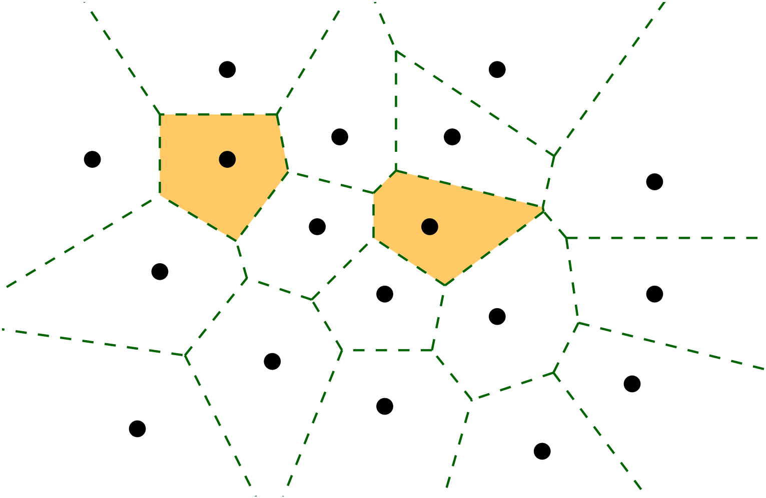
Delaunay triangulations



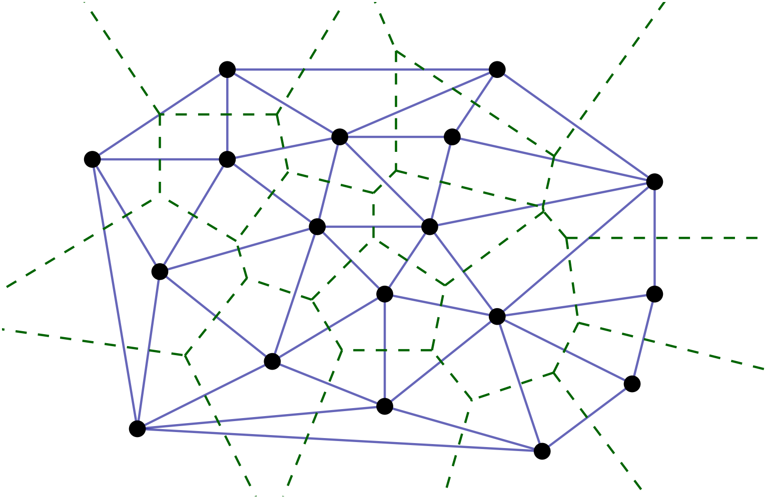
Delaunay triangulations



Delaunay triangulations



Delaunay triangulations

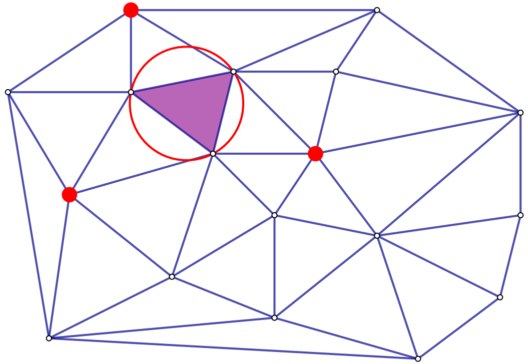


Delaunay triangulations

- triangles with empty circumcircle
- dual to Voronoi diagram
- lexicographically maximizes the vector of non-decreasing angles
- partition of the convex hull
- basic and ubiquotous geometric structure
- computable in $O(n \log n)$, and lower bound of $\Omega(n \log n)$
- several paradigms: incremental, randomized incremental, divide and conquer, via Voronoi diagram, via CH in 3d, etc.
- a 2-dimensional "extension" of sorting

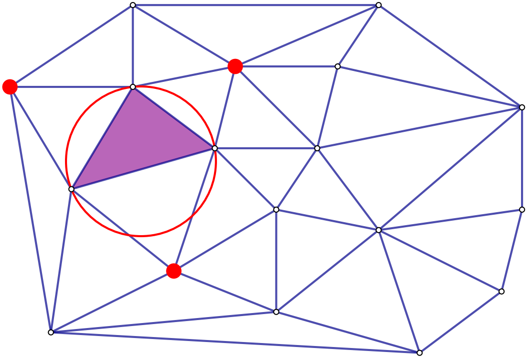
Delaunay triangulations - locality

- testing whether a given triangulation is a Delaunay triangulation
- suffices local test of each triangle with its ≤ 3 neighbor triangles



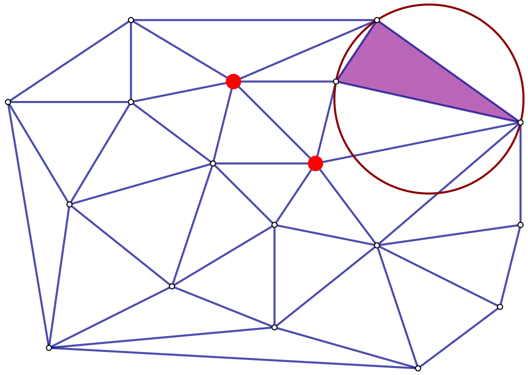
Delaunay triangulations - locality

- testing whether a given triangulation is a Delaunay triangulation
- suffices local test of each triangle with its ≤ 3 neighbor triangles



Delaunay triangulations - locality

- testing whether a given triangulation is a Delaunay triangulation
- suffices local test of each triangle with its ≤ 3 neighbor triangles



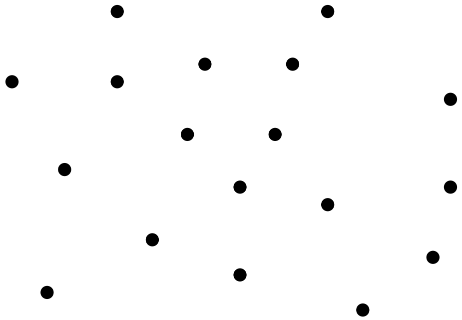
Delaunay triangulations - locality

- testing whether a given triangulation is a Delaunay triangulation
- suffices local test of each triangle with its ≤ 3 neighbor triangles

- $\Theta(n \log n)$ to build Delaunay triangulation
- $\Theta(n)$ to test if a given triangulation is Delaunay

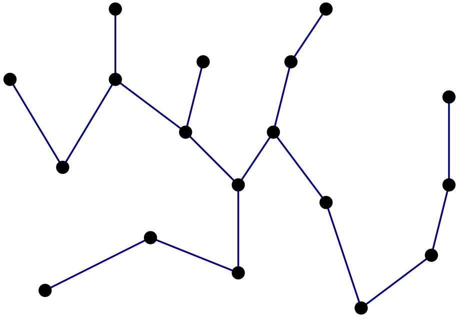
Euclidean MST

- Given a set P of points in $\mathbb{R}^2$, compute its Euclidean MST



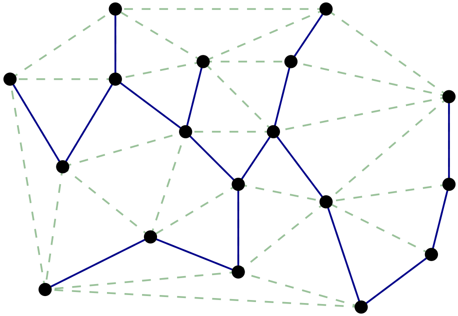
Euclidean MST

- Given a set P of points in $\mathbb{R}^2$, compute its Euclidean MST



Euclidean MST

- Given a set P of points in $\mathbb{R}^2$, compute its Euclidean MST



- $\mathrm{EMST}(P) \subset \mathrm{DT}(P)$
- $\mathrm{EMST}(P)$ in $O(n \log n)$ time via $\mathrm{DT}(P)$
- Non-obvious local testing for a candidate $\mathrm{EMST}(P)$
  $O(n \log^* n)$ expected time via $\mathrm{DT}(P)$ [Devillers 1992]

Predictions

- Prediction: additional information, possibly inaccurate or noisy, which may assist an algorithm to be more effective
(= 'better' solution, faster, etc.)
- Analyze the performance of the algorithm parameterized by how good is the prediction

- Easy example: sorting a list of numbers that is nearly sorted
- Prediction: the list itself
- Parameter: a measure on how sorted the input list is

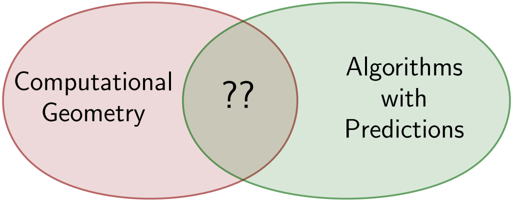
Predictions

- Prediction: additional information, possibly inaccurate or noisy, which may assist an algorithm to be more effective (= 'better' solution, faster, etc.)
- Analyze the performance of the algorithm parameterized by how good is the prediction

- Easy example: sorting a list of numbers that is nearly sorted
- Prediction: the list itself
- Parameter: a measure on how sorted the input list is

- Algorithms with predictions [Mitzenmacher, Vassilvitskii '20]
- Multiple focused events since 2020
- `https://algorithms-with-predictions.github.io/`
  List of 364 papers as of January 2026

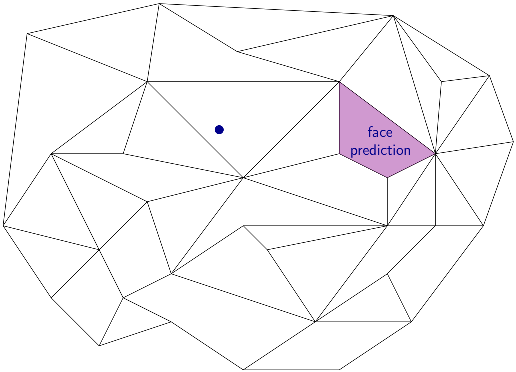
Our aim



- focus in fundamental object: Delaunay triangulation (and EMST)

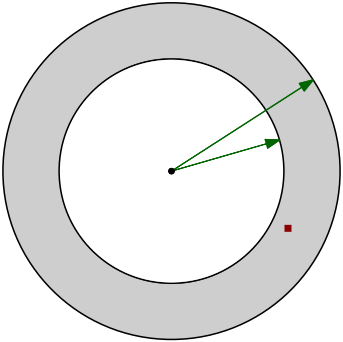
Related work

- Point location with a finger [Iacono, Langerman '03]



Related work

- Point location with a finger [Iacono, Langerman '03]
- Searching a point target in $\mathbb{R}^d$ with an estimate of the distance to the target [Cabello, Giannopoulos '24]



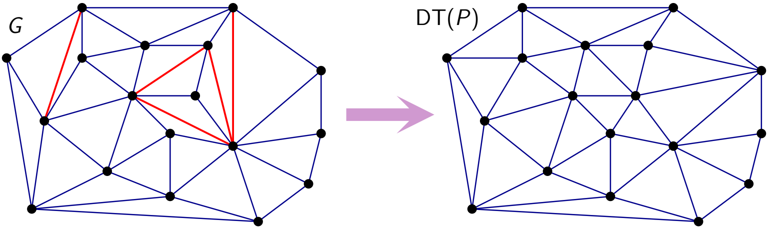
Related work

- Point location with a finger [Iacono, Langerman '03]
- Searching a point target in $\mathbb{R}^d$ with an estimate of the distance to the target [Cabello, Giannopoulos '24]

- Computing $DT(P)$ with additional information:
  - DT(P) from EMST(P) [Devillers '92]
  - merging: DT($P \cup Q$) from DT(P) and DT(Q) [Chazelle '92, Chan '16]
  - splitting: DT(P) from DT($P \cup Q$) [Chazelle et al '02; Chazelle, Mulzer '11]
- (Extension of) verification of structures
  - Given a spanning tree of P, is it EMST(P)?

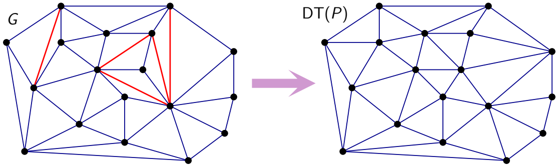
Sample of our results

- Input: a set P of n points and
  a triangulation G of P, hopefully close to $\mathrm{DT}(P)$
- Measure of the prediction:
  (unknown) number D of edges in $E(G) \setminus \mathrm{DT}(P)$



Sample of our results

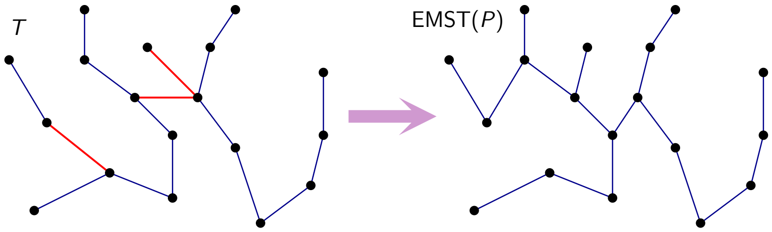
- Input: a set P of n points and
  a triangulation G of P, hopefully close to $\mathrm{DT}(P)$
- Measure of the prediction:
  (unknown) number D of edges in $E(G) \setminus \mathrm{DT}(P)$



- We can compute $\mathrm{DT}(P)$ in $O(n + D \log n)$ expected time
- Smooth interpolation between $O(n)$ for $D = 0$
  and $O(n \log n)$ for $D = \Theta(n)$.
- Optimal, randomized
- Deterministic in $O(n + D \log^3 n)$

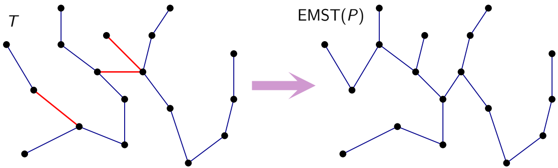
Sample of our results II

- Input: a set P of n points and
  a spanning tree T of P, hopefully similar to EMST(P)
- Measure of the prediction:
  (unknown) number D_{emst} of edges in $E(T) \setminus E(\mathrm{EMST}(P))$



Sample of our results II

- Input: a set P of n points and a spanning tree T of P, hopefully similar to $\mathrm{EMST}(P)$
- Measure of the prediction: (unknown) number D_{emst} of edges in $E(T) \setminus E(\mathrm{EMST}(P))$

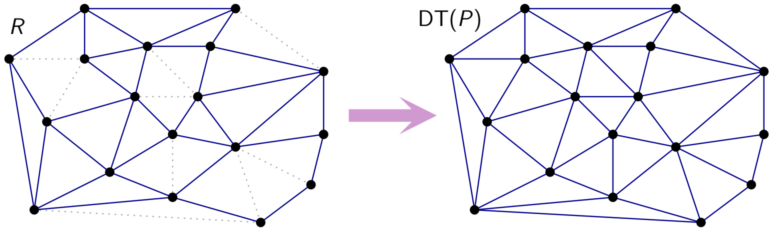


- We can compute $\mathrm{EMST}(P)$ in $O\big((n + D_{\mathrm{emst}} \log n) \log^* n\big)$ expected time
- When $D = 0$, we recover the $O(n \log^* n)$ for testing

Sample of our results III

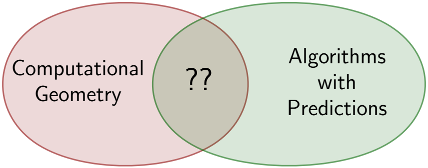
Model with probabilistic noise

- $R \subset_{\text{random}} E(\mathrm{DT}(P))$
  each edge selected independently with probability ρ
- G a triangulation extending R, or R with info about embedding
- Measure of the prediction: (unknown) probability ρ



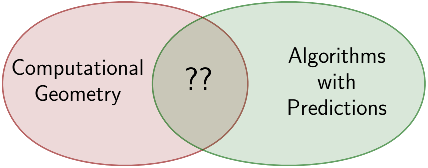
- We can compute $\mathrm{DT}(P)$ in $O(n \operatorname{loglog} n + n \log(1/\rho))$ whp
- For $\rho = 1/\log n$ a small fraction of $E(DT(P))$ is kept

Conclusions



- New paradigm in Computational Geometry
- Basic premise: checking correctness faster than constructing from scratch
- We also use other measures of how close to DT(P)
- Some of our optimal algorithms are randomized. Deterministic?
- Some other algorithms are (not expected?) to be optimal $\log^* n$ and $\log\log n$ factors?

Conclusions



- New paradigm in Computational Geometry
- Basic premise: checking correctness faster than constructing from scratch
- We also use other measures of how close to DT(P)
- Some of our optimal algorithms are randomized. Deterministic?
- Some other algorithms are (not expected?) to be optimal $\log^* n$ and $\log\log n$ factors?

THANKS for your time!!