On Implementing Multiplicatively Weighted Voronoi Diagrams

Martin Held[1]   Stefan de Lorenzo[1]

[1] University of Salzburg, Department of Computer Science

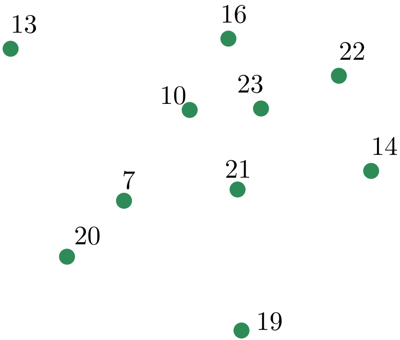
March 16, 2020

UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Problem Specification

Problem

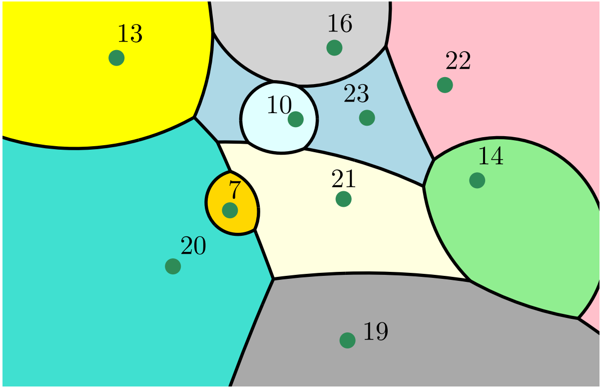
Given: A set S of n input points in the plane, where every $s \in S$ is associated with a weight $w(s) > 0$.



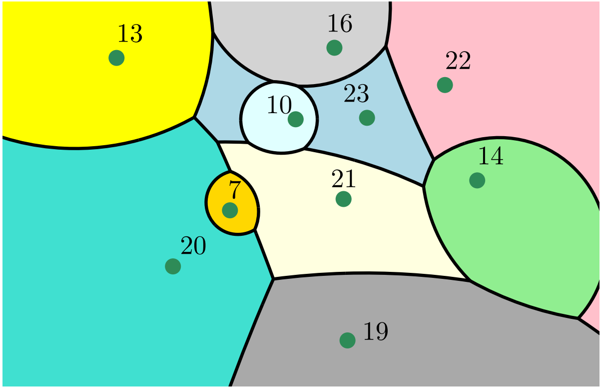
Problem Specification

Problem

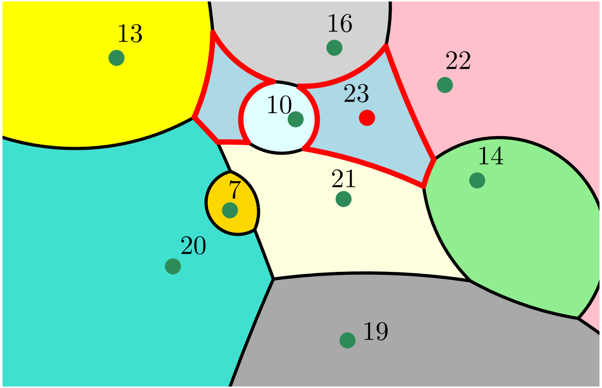
Given: A set S of n input points in the plane, where every $s \in S$ is associated with a weight $w(s) > 0$.
Compute: The multiplicatively weighted Voronoi diagram (MWVD) $\mathcal{VD}_w(S)$ of S.



- The Voronoi edges are formed by straight-line segments, rays, and circular arcs.

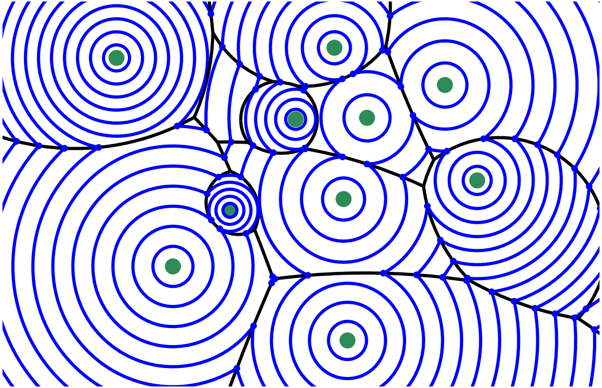


- The Voronoi edges are formed by straight-line segments, rays, and circular arcs.
- The Voronoi regions are (possibly) disconnected.
- The MWVD has a quadratic combinatorial complexity in the worst case.



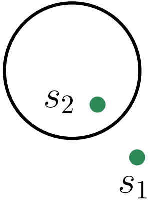
Overview

- We present a wavefront-based approach for computing MWVDs.
- The ***wavefront*** covers an increasing portion of the plane over time.
- It consists of ***wavefront arcs*** and ***wavefront vertices***.
- Whenever a wavefront arc vanishes or spawns, a new Voronoi node is discovered.



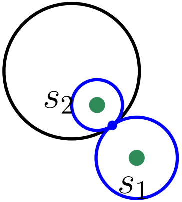
Offset Circles

- Every site is associated with an ***offset circle***.



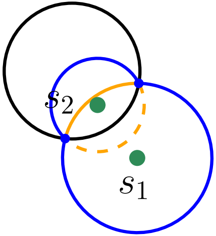
Offset Circles

- Every site is associated with an ***offset circle***.
- Two ***moving intersection points*** trace out the bisector as time progresses.



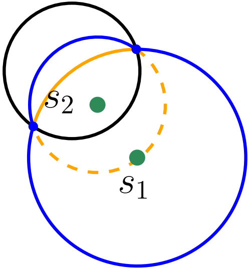
Offset Circles

- Every site is associated with an ***offset circle***.
- Two ***moving intersection points*** trace out the bisector as time progresses.
- ***Inactive arcs*** along the offset circles are eliminated.
- The ***active arcs*** are stored in sorted angular order.



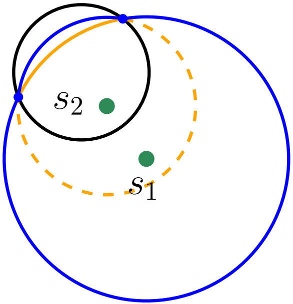
Offset Circles

- Every site is associated with an ***offset circle***.
- Two ***moving intersection points*** trace out the bisector as time progresses.
- ***Inactive arcs*** along the offset circles are eliminated.
- The ***active arcs*** are stored in sorted angular order.



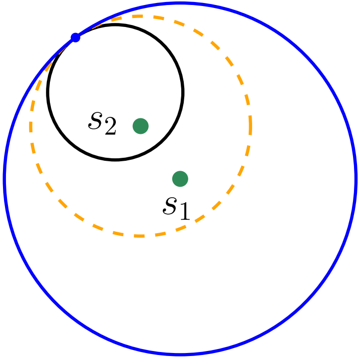
Offset Circles

- Every site is associated with an ***offset circle***.
- Two ***moving intersection points*** trace out the bisector as time progresses.
- ***Inactive arcs*** along the offset circles are eliminated.
- The ***active arcs*** are stored in sorted angular order.



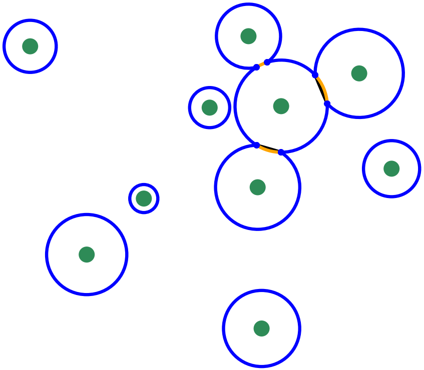
Offset Circles

- Every site is associated with an ***offset circle***.
- Two ***moving intersection points*** trace out the bisector as time progresses.
- ***Inactive arcs*** along the offset circles are eliminated.
- The ***active arcs*** are stored in sorted angular order.



Event Handling

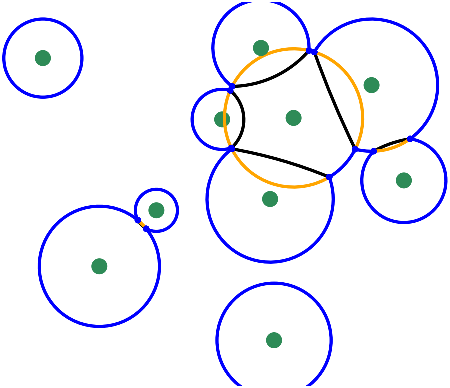
- ***Collision*** and ***domination events*** mark the initial and last contact of a two offset circles.
- ***Arc events*** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $\mathcal{Q}$.
- The angular order of active arcs only changes at events.



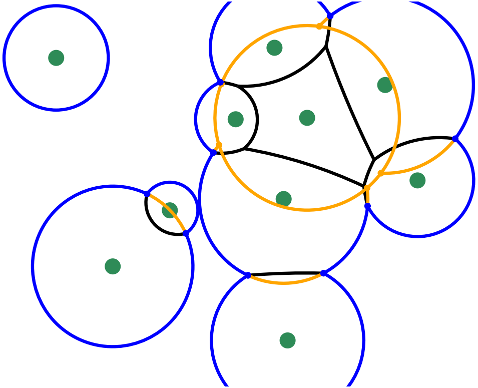
UNIVERSITÄT SALZBURG
Computational Geometry and Applications Lab

Event Handling

- ***Collision*** and ***domination events*** mark the initial and last contact of a two offset circles.
- ***Arc events*** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $\mathcal{Q}$.
- The angular order of active arcs only changes at events.

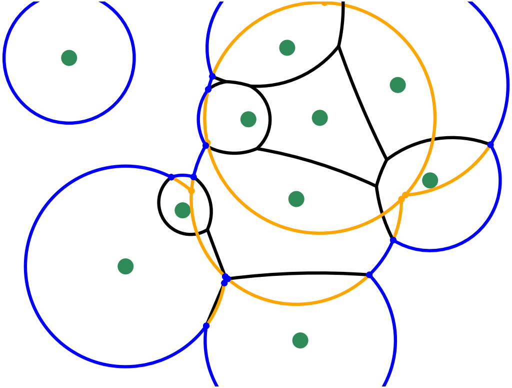


- ***Collision*** and ***domination events*** mark the initial and last contact of a two offset circles.
- ***Arc events*** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $\mathcal{Q}$.
- The angular order of active arcs only changes at events.

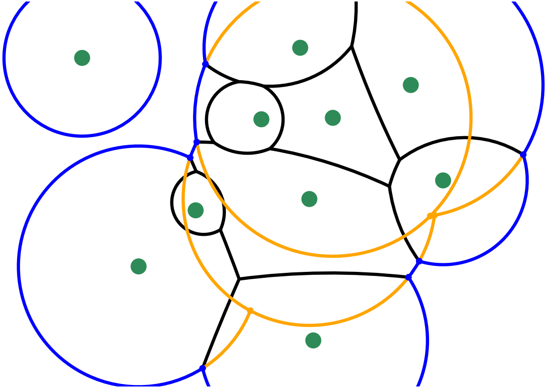


Event Handling

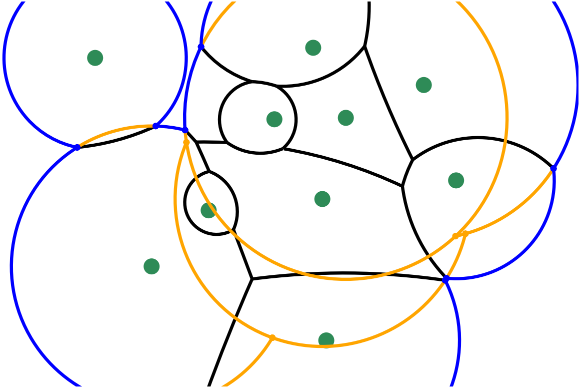
- ***Collision*** and ***domination events*** mark the initial and last contact of a two offset circles.
- ***Arc events*** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $\mathcal{Q}$.
- The angular order of active arcs only changes at events.



- ***Collision*** and ***domination events*** mark the initial and last contact of a two offset circles.
- ***Arc events*** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $\mathcal{Q}$.
- The angular order of active arcs only changes at events.

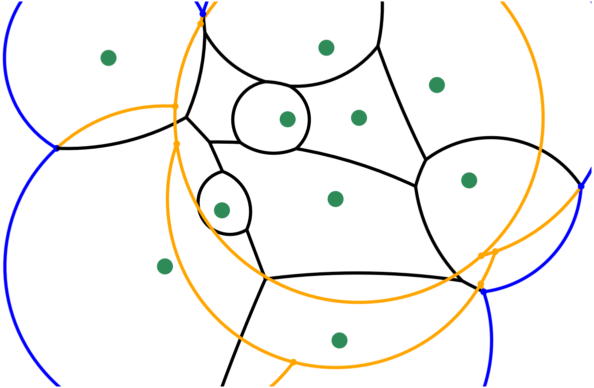


- ***Collision*** and ***domination events*** mark the initial and last contact of a two offset circles.
- ***Arc events*** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $\mathcal{Q}$.
- The angular order of active arcs only changes at events.

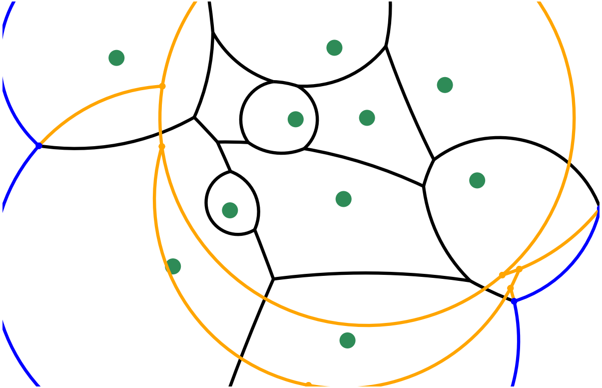


Event Handling

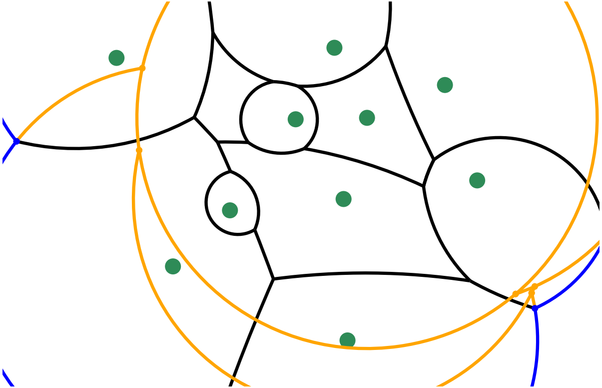
- ***Collision*** and ***domination events*** mark the initial and last contact of a two offset circles.
- ***Arc events*** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $\mathcal{Q}$.
- The angular order of active arcs only changes at events.



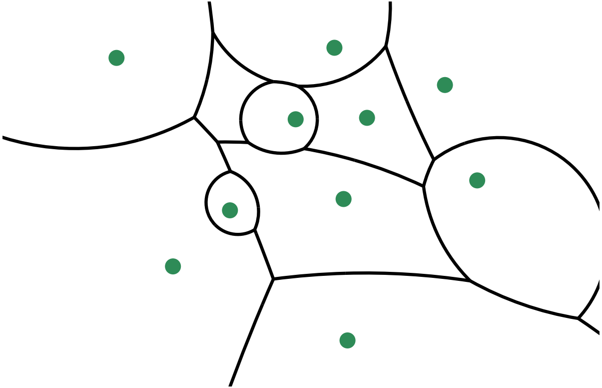
- ***Collision*** and ***domination events*** mark the initial and last contact of a two offset circles.
- ***Arc events*** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $\mathcal{Q}$.
- The angular order of active arcs only changes at events.



- ***Collision*** and ***domination events*** mark the initial and last contact of a two offset circles.
- ***Arc events*** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $\mathcal{Q}$.
- The angular order of active arcs only changes at events.



- ***Collision*** and ***domination events*** mark the initial and last contact of a two offset circles.
- ***Arc events*** happen whenever active arcs vanish or spawn.
- These events are stored in a priority queue $\mathcal{Q}$.
- The angular order of active arcs only changes at events.

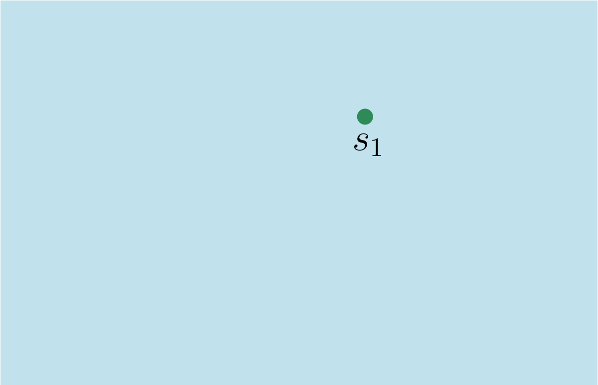


Event Handling

- All topological changes of the wavefront are properly detected.
- A quadratic number of collision events are computed in any case.
- A moving intersection can be charged with a constant number of arc events.
- In the worst case $\mathcal{O}(n^2)$ arc events take place.
- All events can be handled in $\mathcal{O}(\log n)$ time.
- Therefore, the algorithms runtime is $\mathcal{O}(n^2 \log n)$ in the worst case.

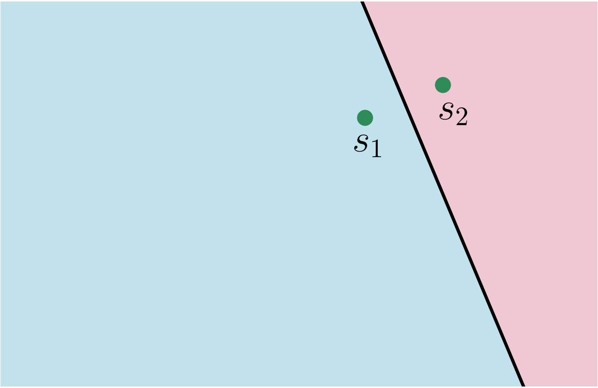
Reducing the Number of Collisions

- A vast number of collisions are invalid for general input.
- The calculation of all possible collision requires a high computational effort.
- Invalid collision are filtered in an additional preprocessing step.
- Thus, the average case behavior of the algorithm is improved.



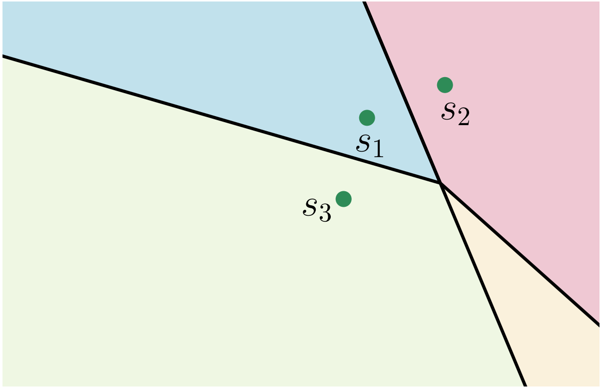
Reducing the Number of Collisions

- A vast number of collisions are invalid for general input.
- The calculation of all possible collision requires a high computational effort.
- Invalid collision are filtered in an additional preprocessing step.
- Thus, the average case behavior of the algorithm is improved.



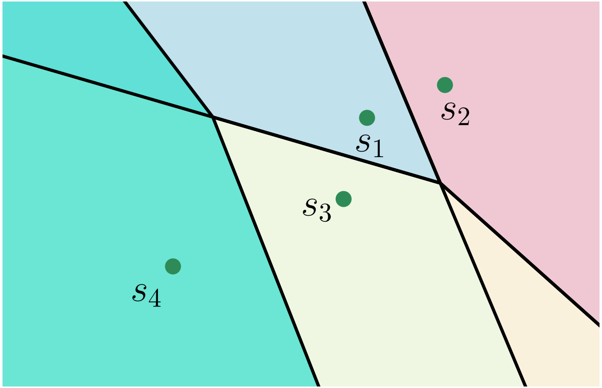
Reducing the Number of Collisions

- A vast number of collisions are invalid for general input.
- The calculation of all possible collision requires a high computational effort.
- Invalid collision are filtered in an additional preprocessing step.
- Thus, the average case behavior of the algorithm is improved.



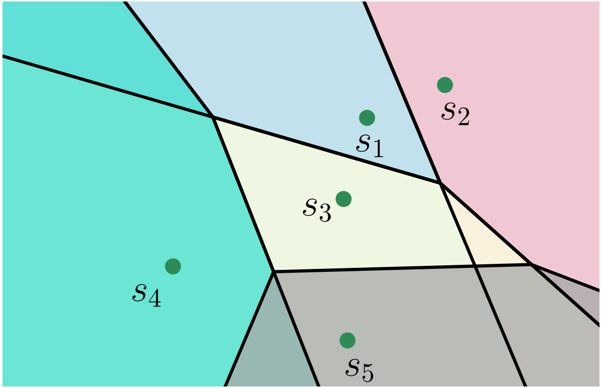
Reducing the Number of Collisions

- A vast number of collisions are invalid for general input.
- The calculation of all possible collision requires a high computational effort.
- Invalid collision are filtered in an additional preprocessing step.
- Thus, the average case behavior of the algorithm is improved.

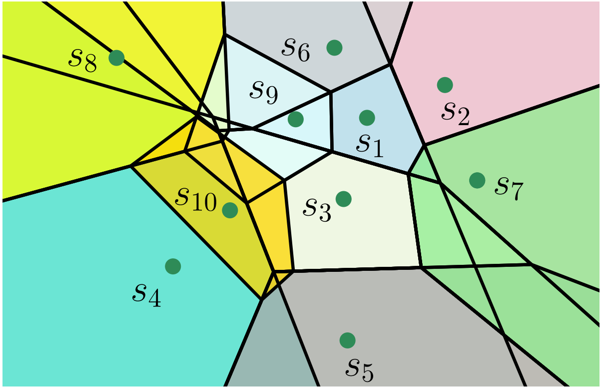


Reducing the Number of Collisions

- A vast number of collisions are invalid for general input.
- The calculation of all possible collision requires a high computational effort.
- Invalid collision are filtered in an additional preprocessing step.
- Thus, the average case behavior of the algorithm is improved.

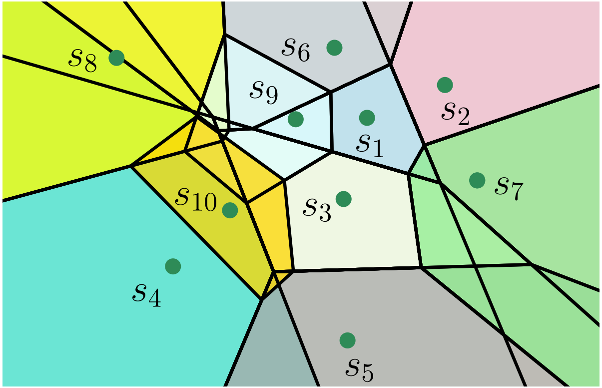


- A vast number of collisions are invalid for general input.
- The calculation of all possible collision requires a high computational effort.
- Invalid collision are filtered in an additional preprocessing step.
- Thus, the average case behavior of the algorithm is improved.



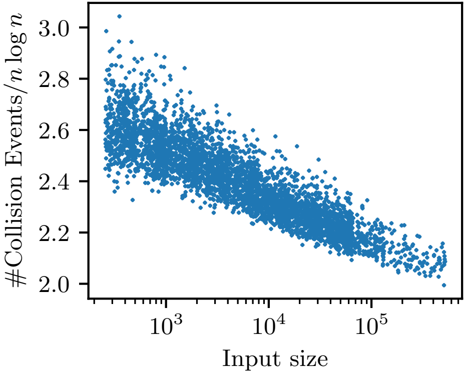
Overlay Arrangement

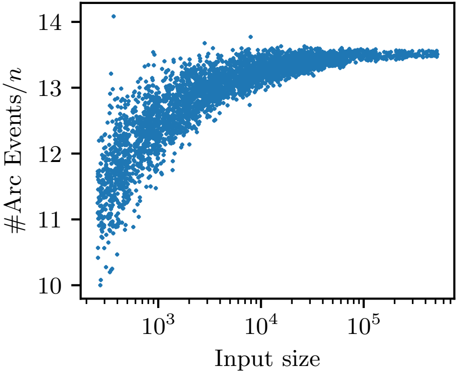
- Only sites within the same candidate set may collide.
- A candidate set contains $\mathcal{O}(\log n)$ many sites in the expected case [HPR15].
- The expected complexity of overlay arrangement is bound by $\mathcal{O}(n \log n)$ [KRS11].
- Thus, it is necessary to compute $\mathcal{O}(n \log^3 n)$ many collisions.



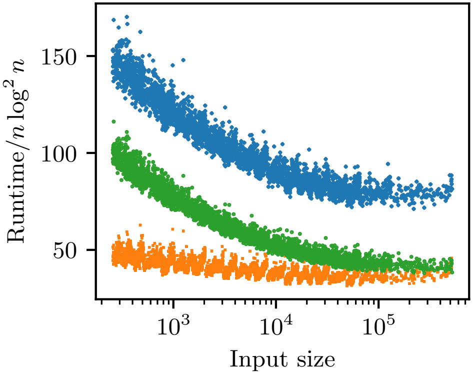
Experimental Evaluation

- The implementation is based on the Computational Geometry Algorithms Library (CGAL).
- We tested our strategy on 3800 randomly generated inputs.
- All tests were carried out on an Intel Core i7-6700 clocked at 3.40GHz.





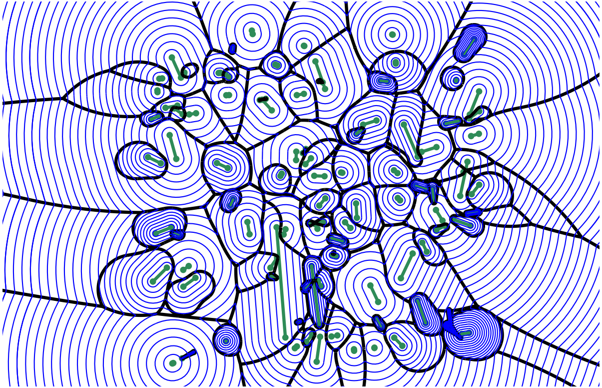
- The implementation is based on the Computational Geometry Algorithms Library (CGAL).
- We tested our strategy on 3800 randomly generated inputs.
- All tests were carried out on an Intel Core i7-6700 clocked at 3.40GHz.



Overlay arrangement Event queue Overall runtime

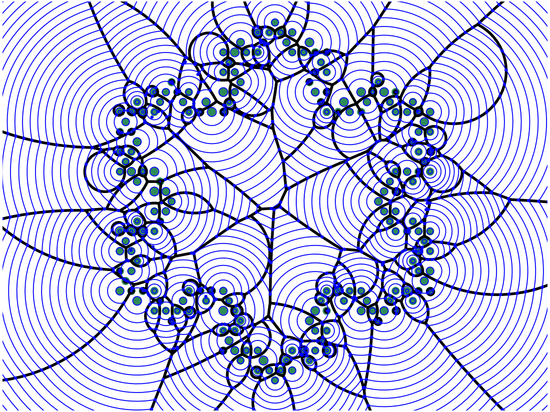
Extensions

- Our (basic) algorithm is also able to deal with additive weights.
- The strategy can be easily extended to also handle weighted straight-line segments.

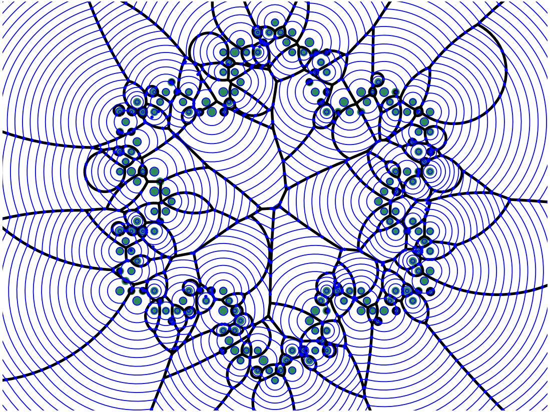


Discussion

- We propose a fast, practical strategy to compute MWVDs.
- The expected runtime is improved by using an overlay arrangement.
- We provide a robust implementation using exact arithmetic.



- We propose a fast, practical strategy to compute MWVDs.
- The expected runtime is improved by using an overlay arrangement.
- We provide a robust implementation using exact arithmetic.



Thank you for your attention!

Sariel Har-Peled and Benjamin Raichel.
On the Complexity of Randomly Weighted Multiplicative Voronoi Diagrams.
Discrete Comput. Geom., 53(3):547–568, 2015.

Haim Kaplan, Edgar Ramos, and Micha Sharir.
The Overlay of Minimization Diagrams in a Randomized Incremental Construction.
Discrete Comput. Geom., 45(3):371–382, 2011.