What Else Can Voronoi Diagrams Do For Diameter In Planar Graphs?

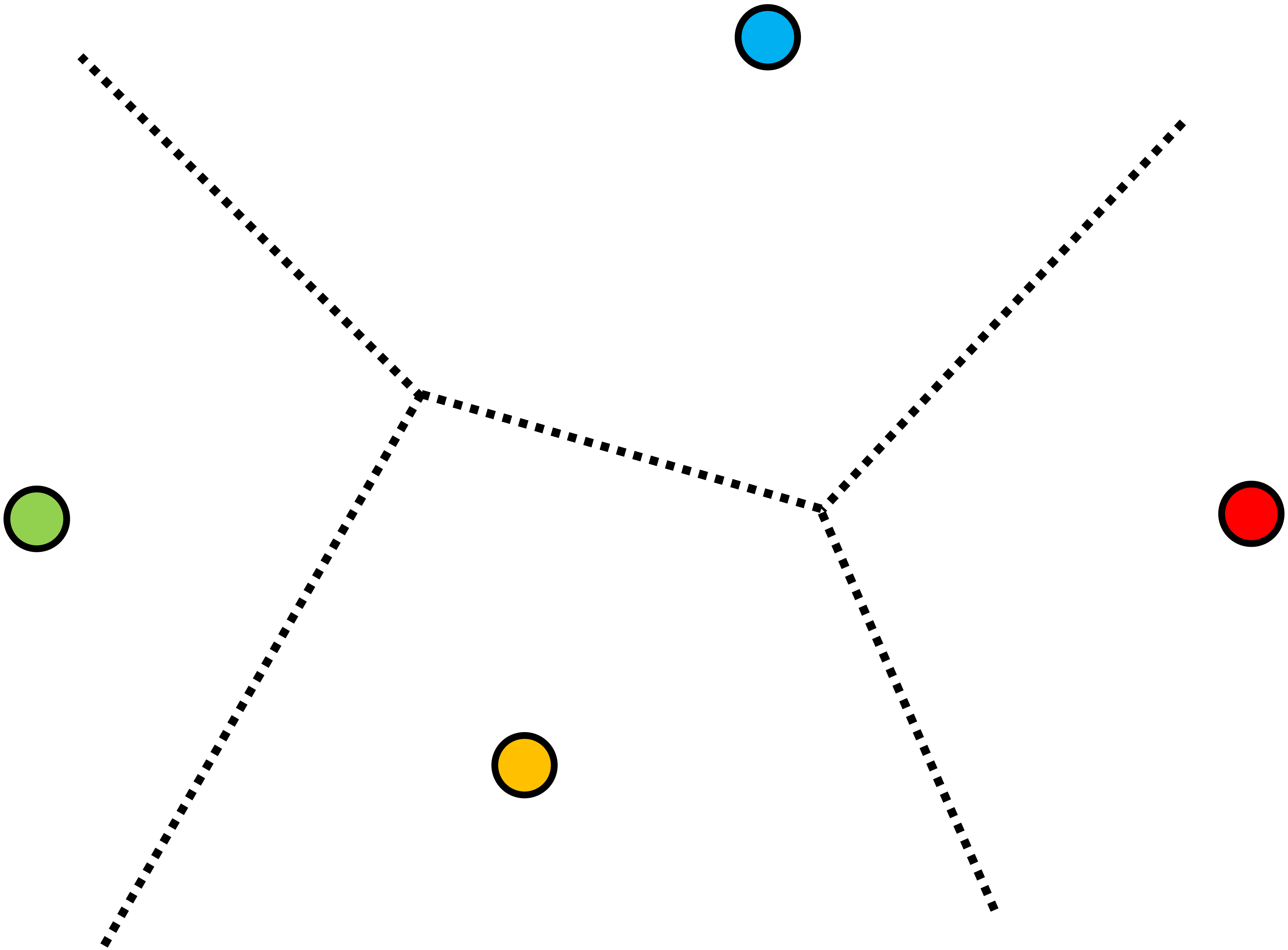
Amir Abboud, Shay Mozes and Oren Weimann

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

אוניברסיטת רייכמן
Reichman University

אוניברסיטת חיפה
University of Haifa
جامعة حيفا

Voronoi diagrams

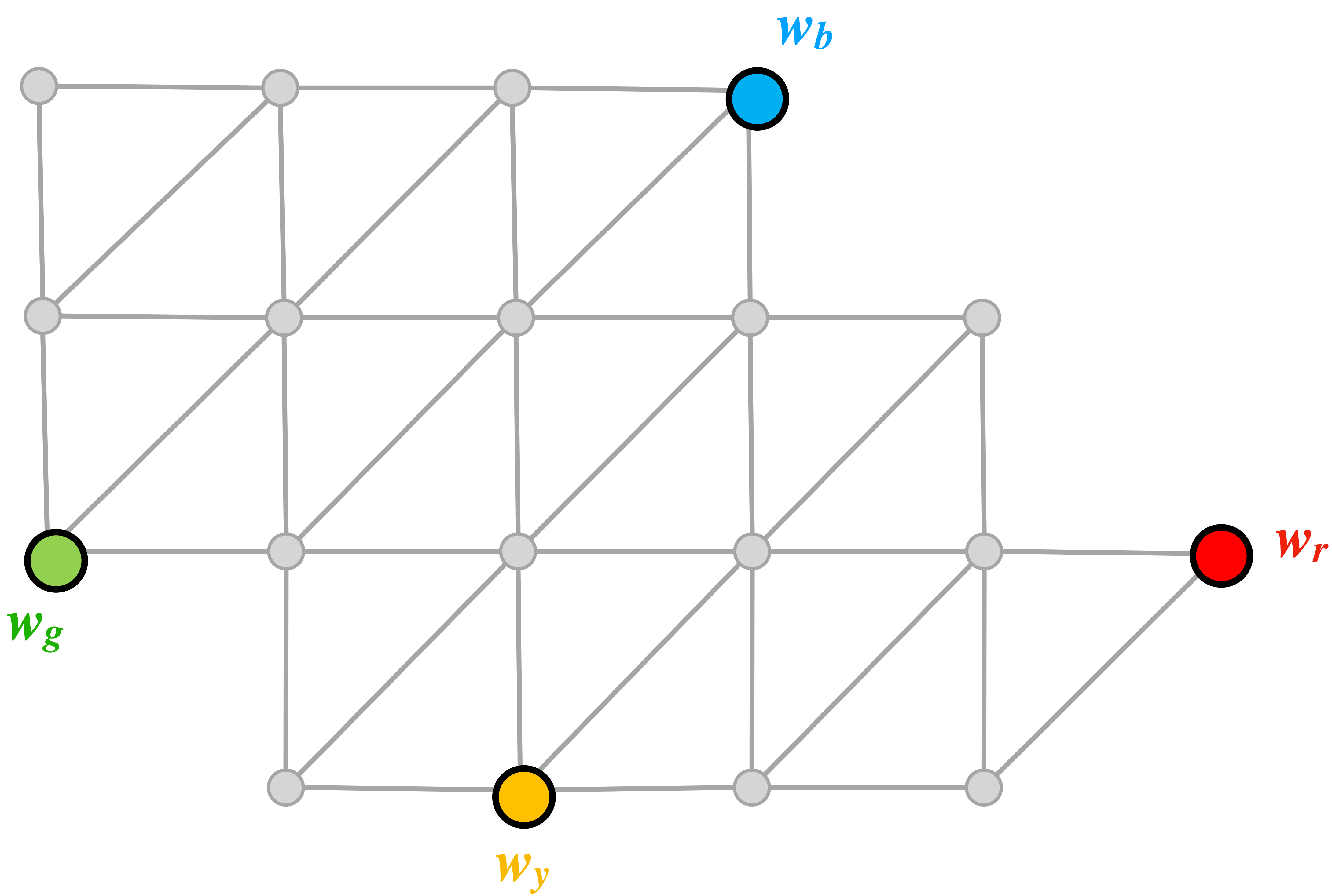


Voronoi
1908

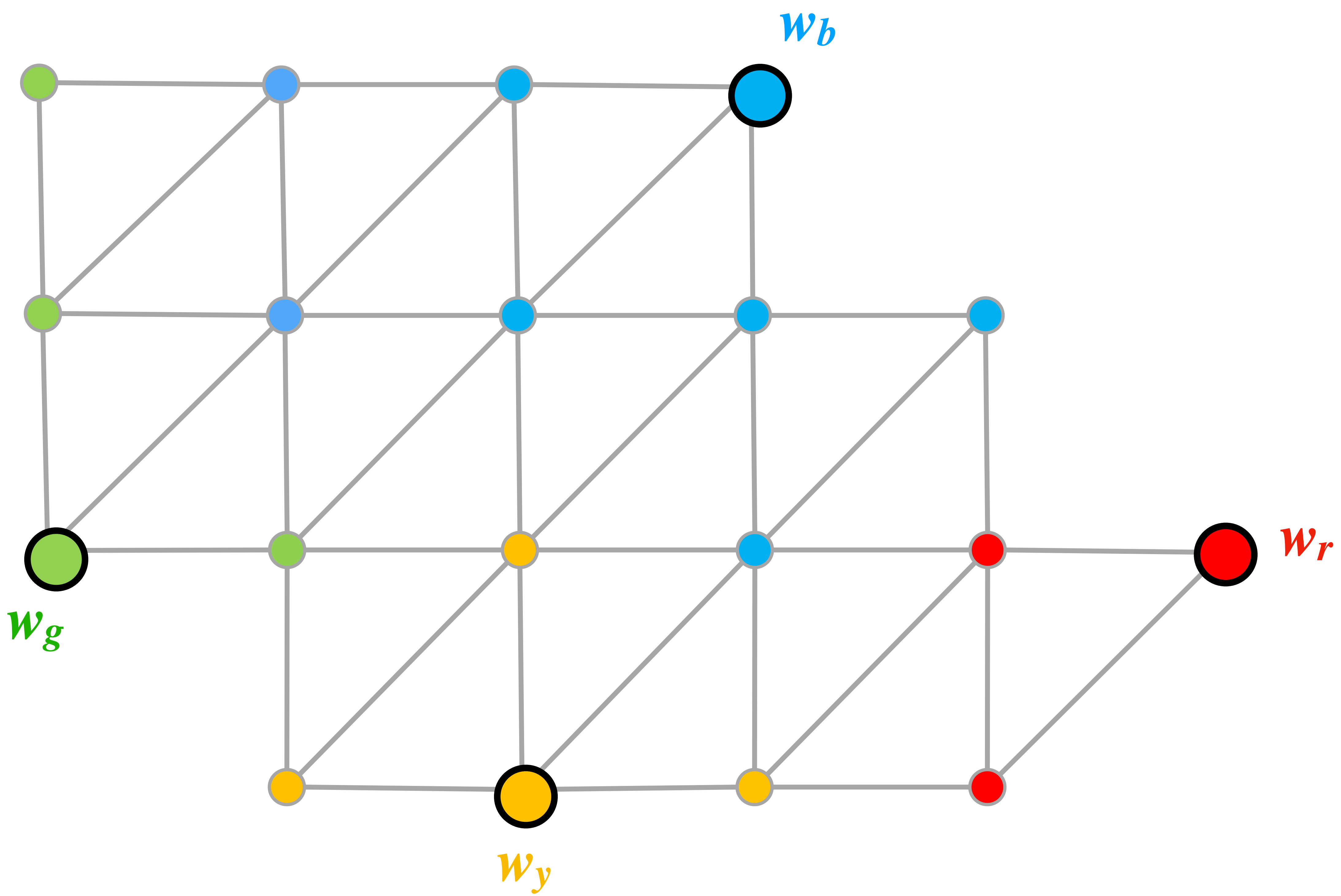
Descartes
1644

Dirichlet
1850

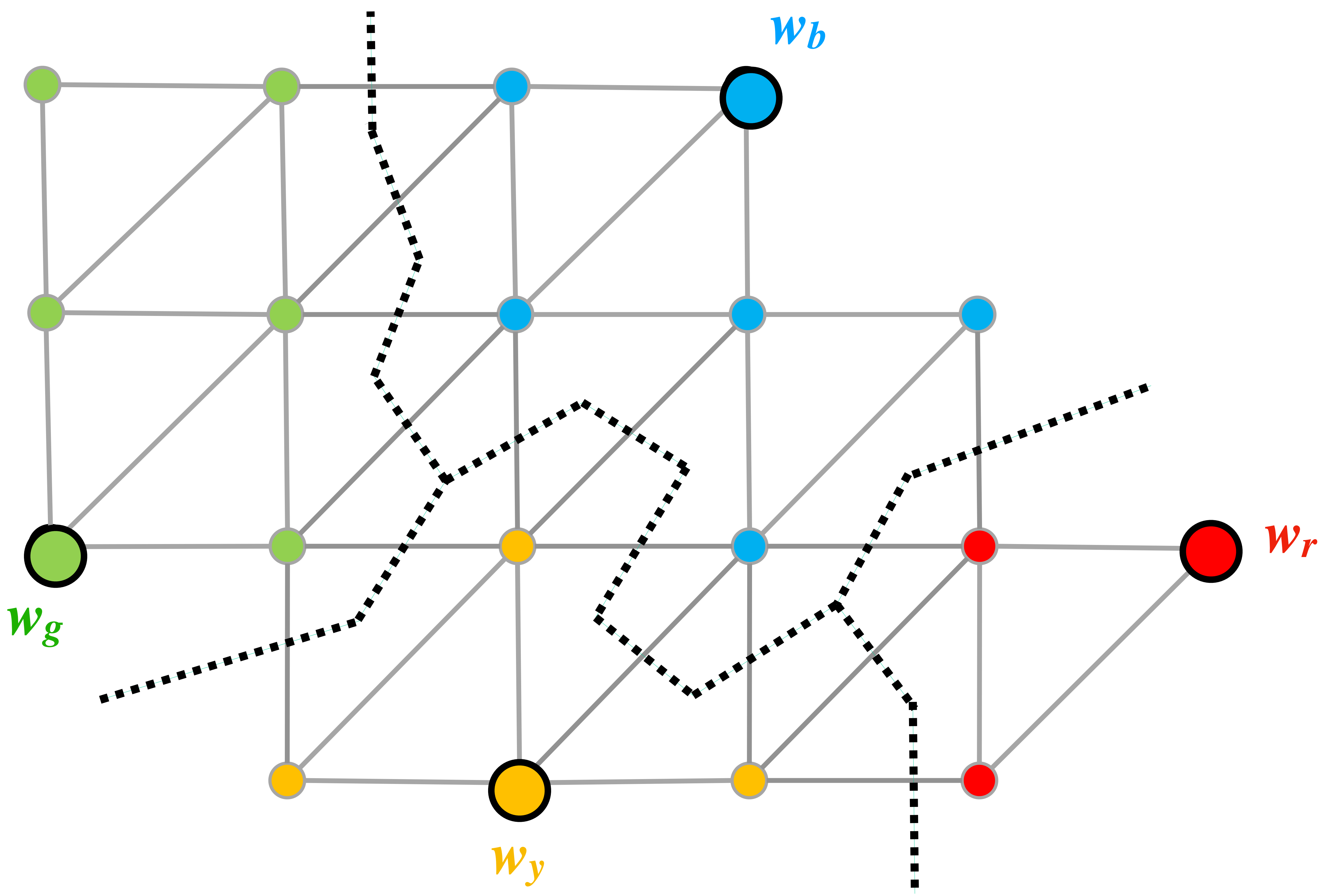
Voronoi diagrams on planar graphs



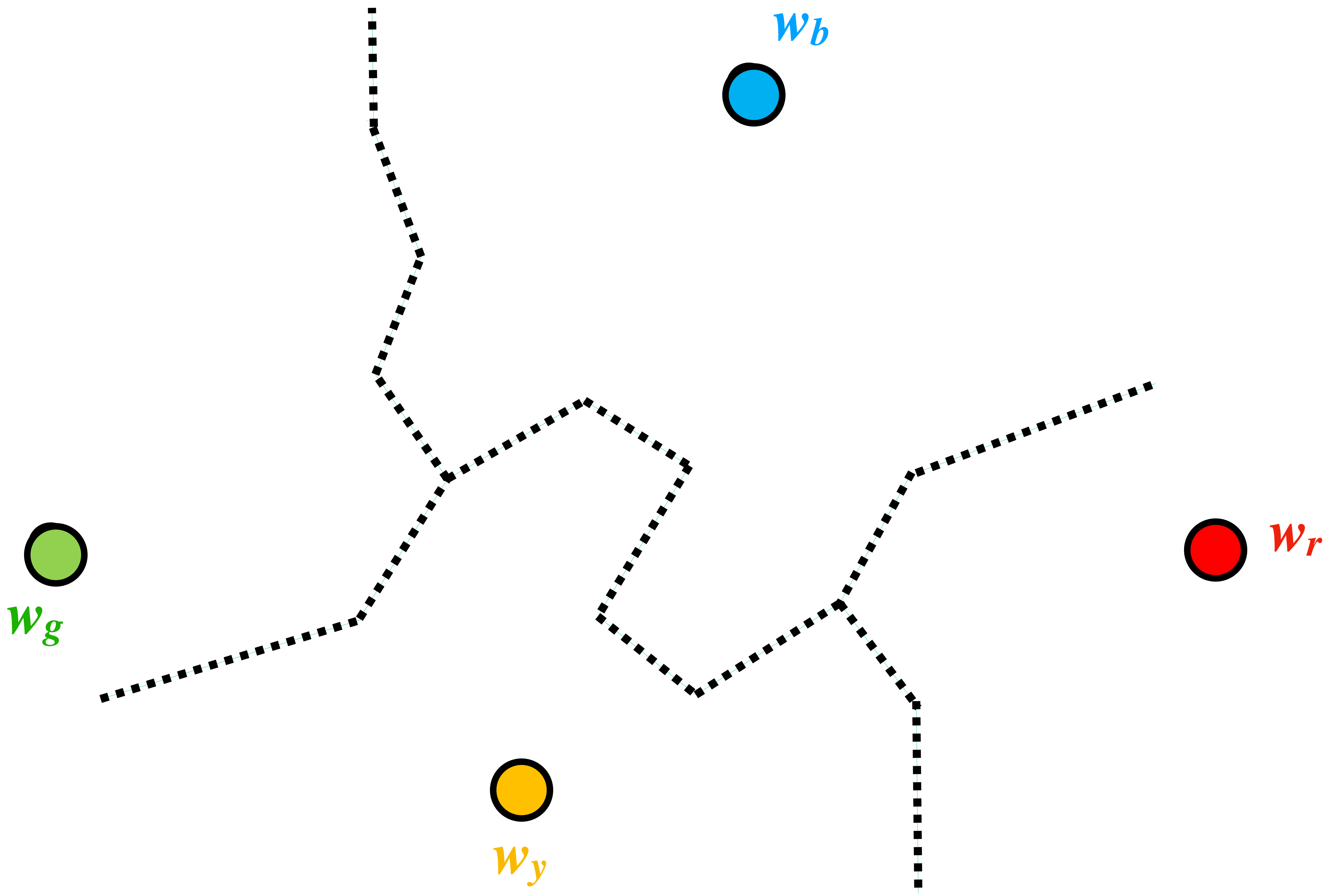
Voronoi diagrams on planar graphs



Voronoi diagrams on planar graphs



Voronoi diagrams on planar graphs



Planar diameter via Voronoi Diagrams

- Cabello's breakthrough (best paper in SODA 2017)
  - After some preprocessing, can quickly construct Voronoi diagrams on planar graphs
  - Can use Voronoi diagrams to compute the diameter (max length shortest path) in sub-quadratic time
- Best current algorithm for planar diameter uses this approach and runs in $\tilde{O}(n^{5/3})$ time [GKMSW SODA'18]

Questions

- **What Else Can Voronoi Diagrams Do For Diameter In Planar Graphs?**
- What other problems are Voronoi diagrams good for in planar graphs?

- **This talk**

- Almost optimal distance oracles [CDW FOCS'17, GMWW SODA'18, CGMW STOC'19, PL SODA'21]

What Else Can Voronoi Diagrams Do For Diameter In Planar Graphs?

- Can we improve on $\tilde{O}(n^{5/3})$ for static planar diameter?
- Can we dynamically maintain the diameter under updates of the planar graph?
- Can we prove any non-trivial lower bounds for planar diameter?

New upper bounds using Voronoi Diagrams for undirected unweighted planar graphs

- Faster static diameter algorithms, parametrized by the diameter D:
  - $\tilde{O}(nD^2)$ time (faster for $D < n^{1/3}$)
  - $n^{3+o(1)}/D^2$ time (faster for $D > n^{2/3}$)
- Fault-tolerant: algorithm for replacement diameter in $n^{7/3+o(1)}$ time (instead of $\tilde{O}(n^{8/3})$)
- Dynamic: algorithm for maintaining the diameter under edge insertions in total $n^{7/3+o(1)}$ time (instead of $\tilde{O}(n^{8/3})$)

New lower bound for weighted planar graphs

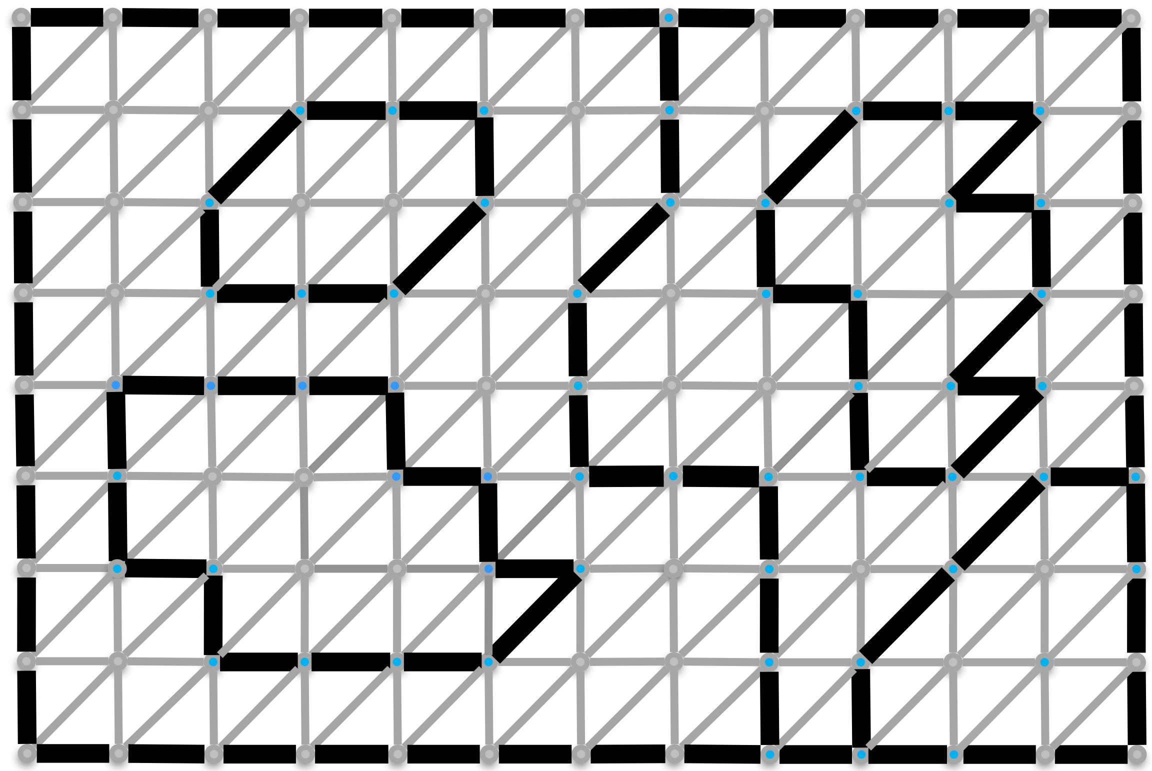
- Conditional lower bound (on SETH), ruling out updates in $O(n^{1-\epsilon})$ amortized time (even for just incremental or decremental updates)

Technical novelty

- Step towards dynamic VDs: in the replacement diameter algorithm: algorithm that updates an existing VD faster than recomputing it from scratch.
- So far VDs were only used for small cycle separators. We use VDs with different small cycles.
- First dynamic planar lower bound conditioned on SETH.

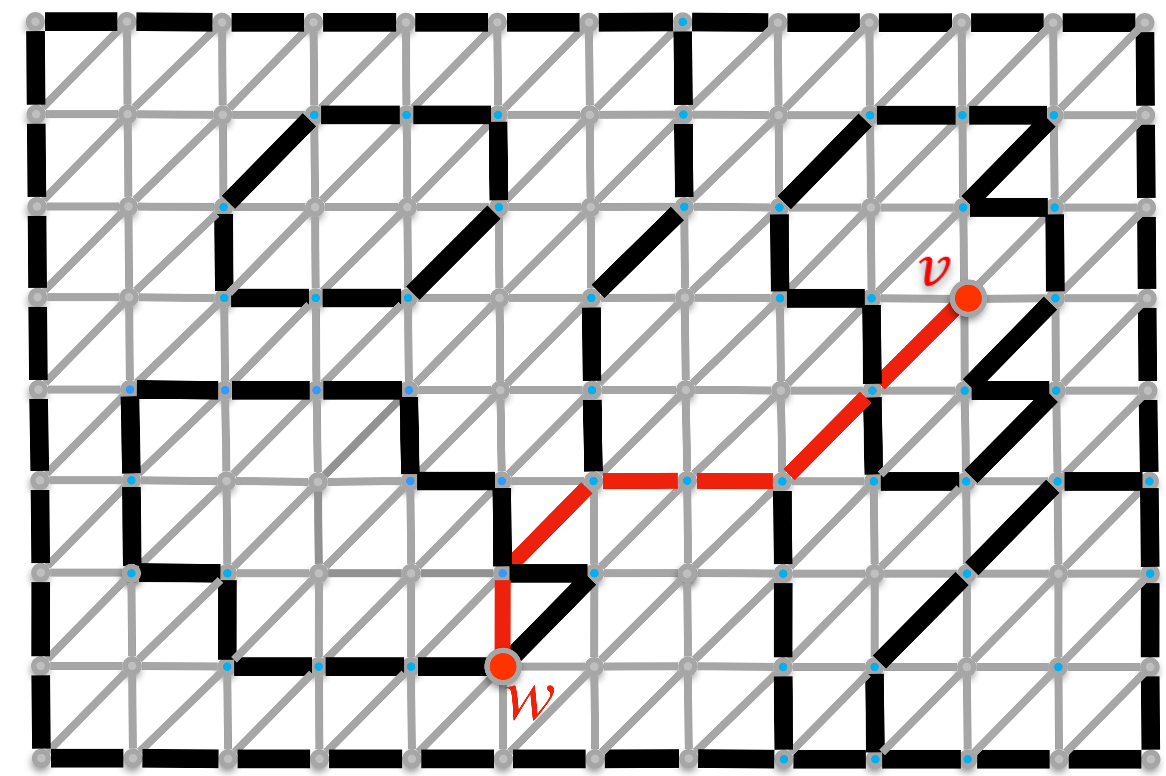
High-level approach for diameter

- compute an r-division: $O(n/r)$ pieces, each with $O(r)$ vertices and $O(r^{1/2})$ boundary vertices



High-level approach for diameter

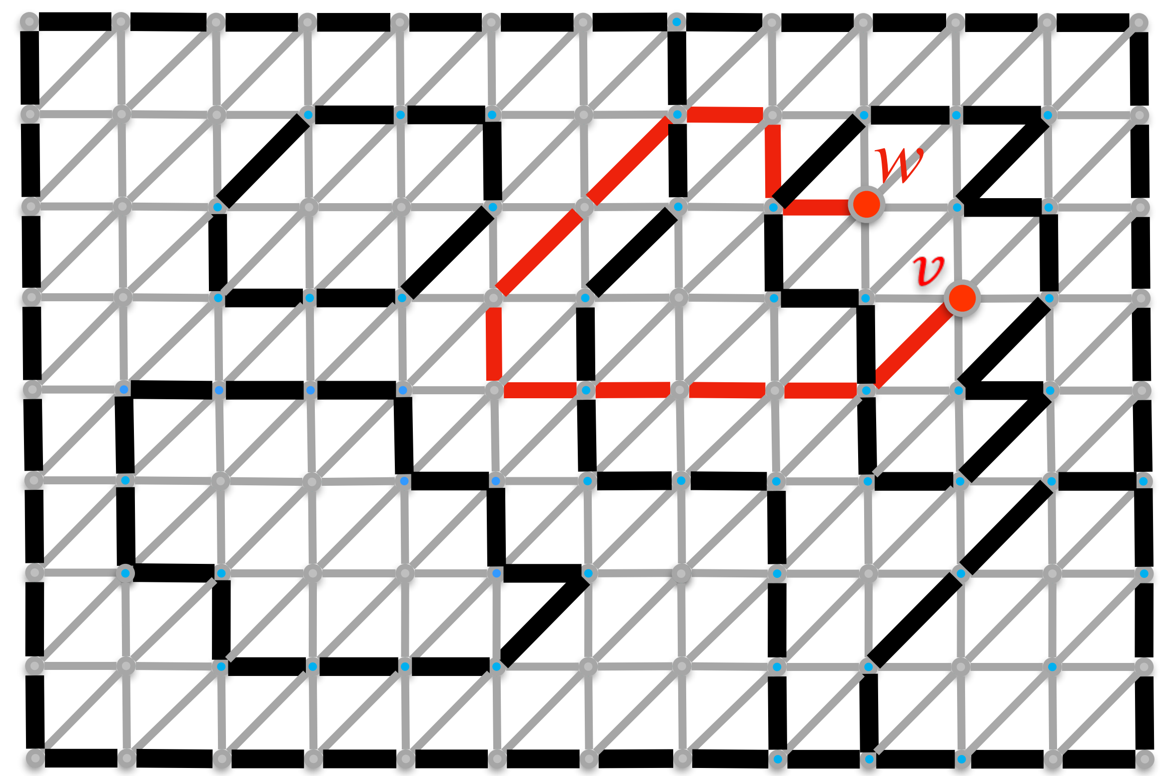
- compute an r-division: $O(n/r)$ pieces, each with $O(r)$ vertices and $O(r^{1/2})$ boundary vertices



- there are three types of distances:
  - between a vertex and a boundary vertex

High-level approach for diameter

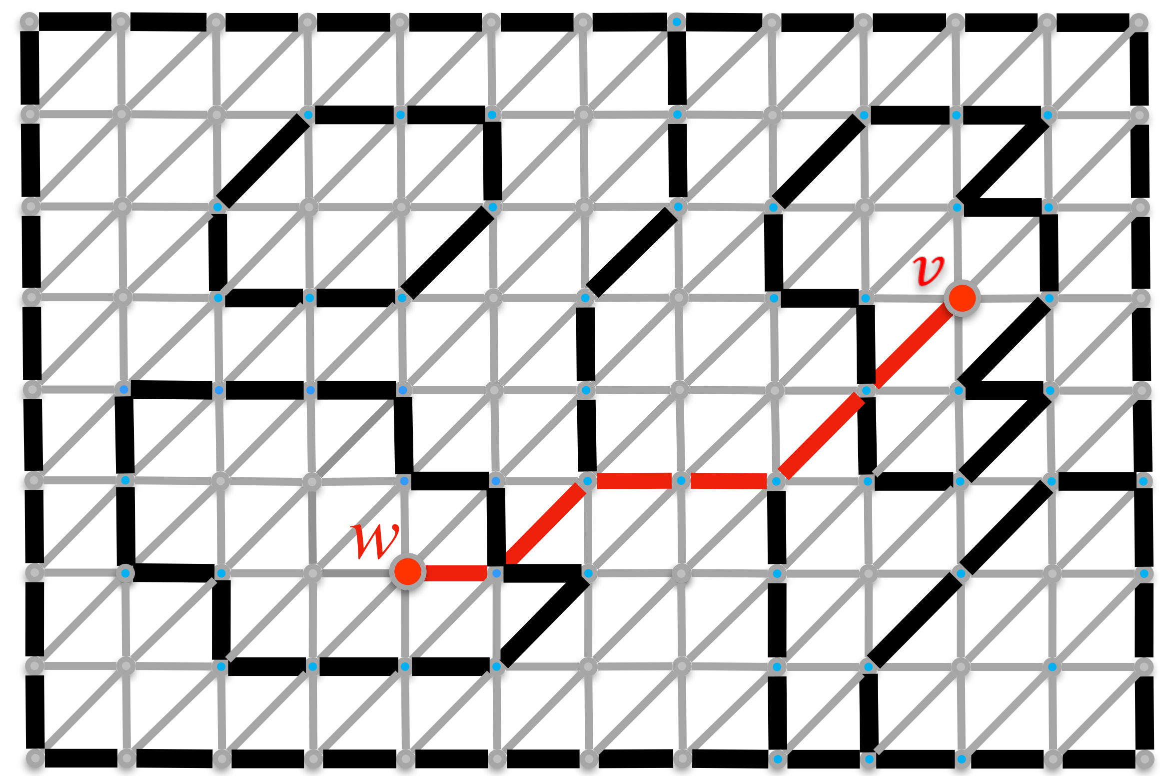
- compute an r-division:
  $O(n/r)$ pieces, each
  with $O(r)$ vertices and
  $O(r^{1/2})$ boundary vertices



- there are three types of distances:
  - between a vertex and a boundary vertex
  - between two vertices in the same piece

High-level approach for diameter

- compute an r-division: $O(n/r)$ pieces, each with $O(r)$ vertices and $O(r^{1/2})$ boundary vertices



- there are three types of distances:
  - between a vertex and a boundary vertex
  - between two vertices in the same piece
  - between two vertices in different pieces

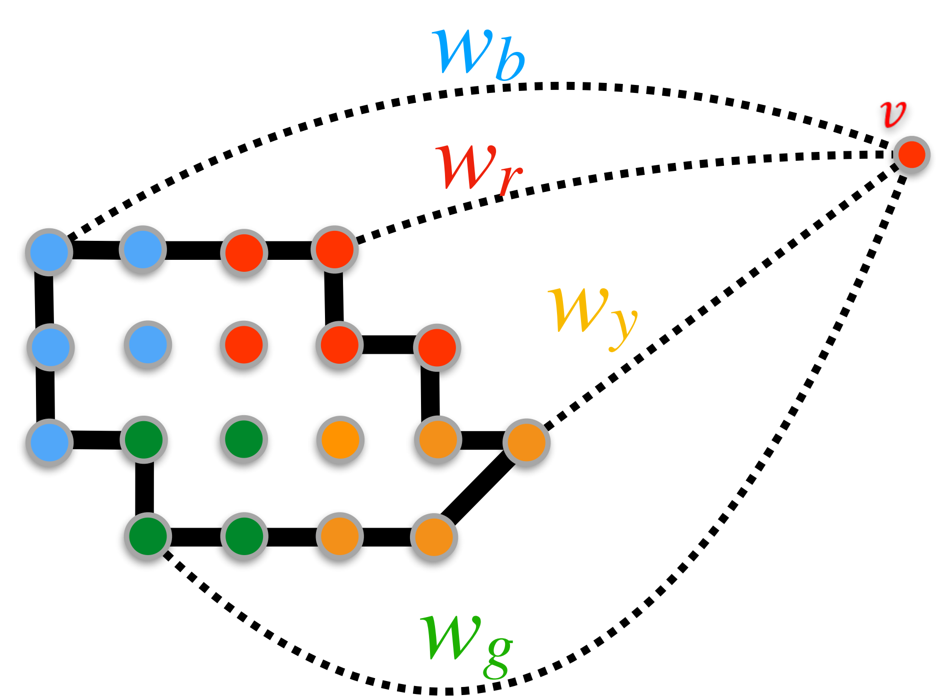
Dist. between vertices in different pieces

- already computed distances from v to boundary nodes of the other piece
- compute additively weighted Voronoi diagram for the other piece in $\tilde{O}(r^{1/2})$ time
- use Voronoi diagram to return the node furthest from each boundary site (in its cell) in $\tilde{O}(1)$ time per site
- total $\tilde{O}(n \cdot n/r \cdot r^{1/2}) = \tilde{O}(n^2/r^{1/2})$

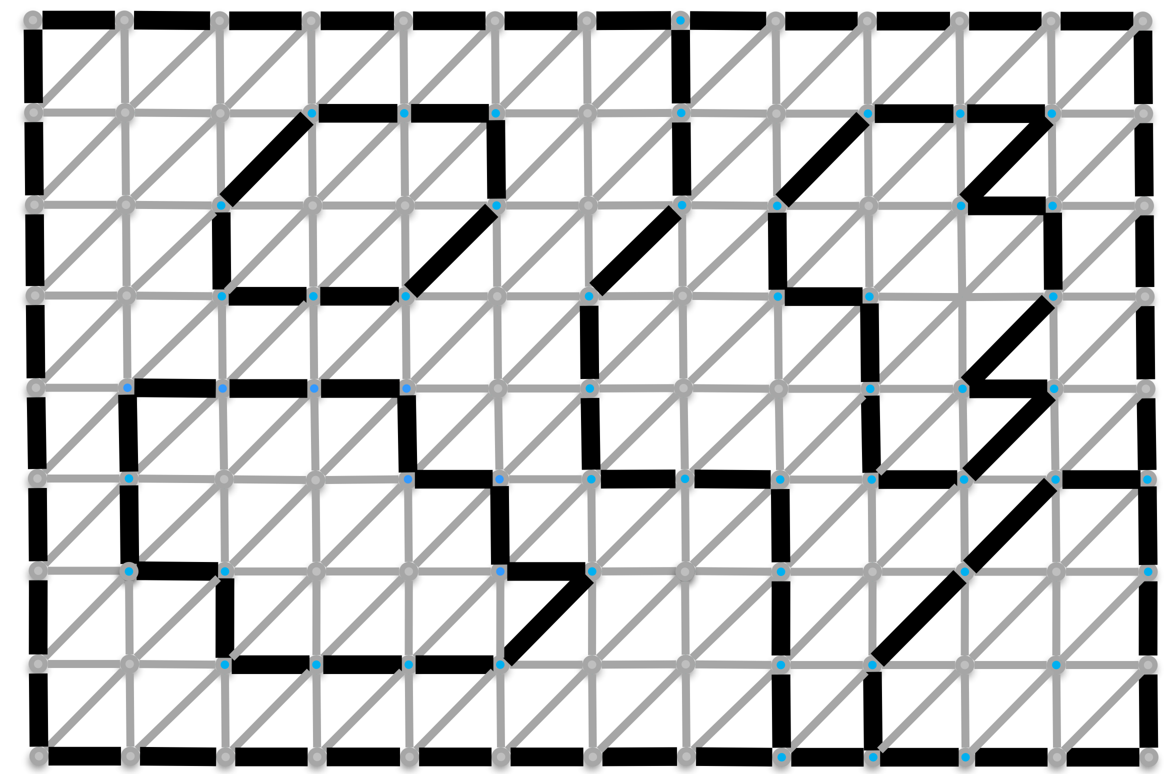
vertices

pieces

- requires $\tilde{O}(n/r \cdot r^2) = O(nr)$ preprocessing

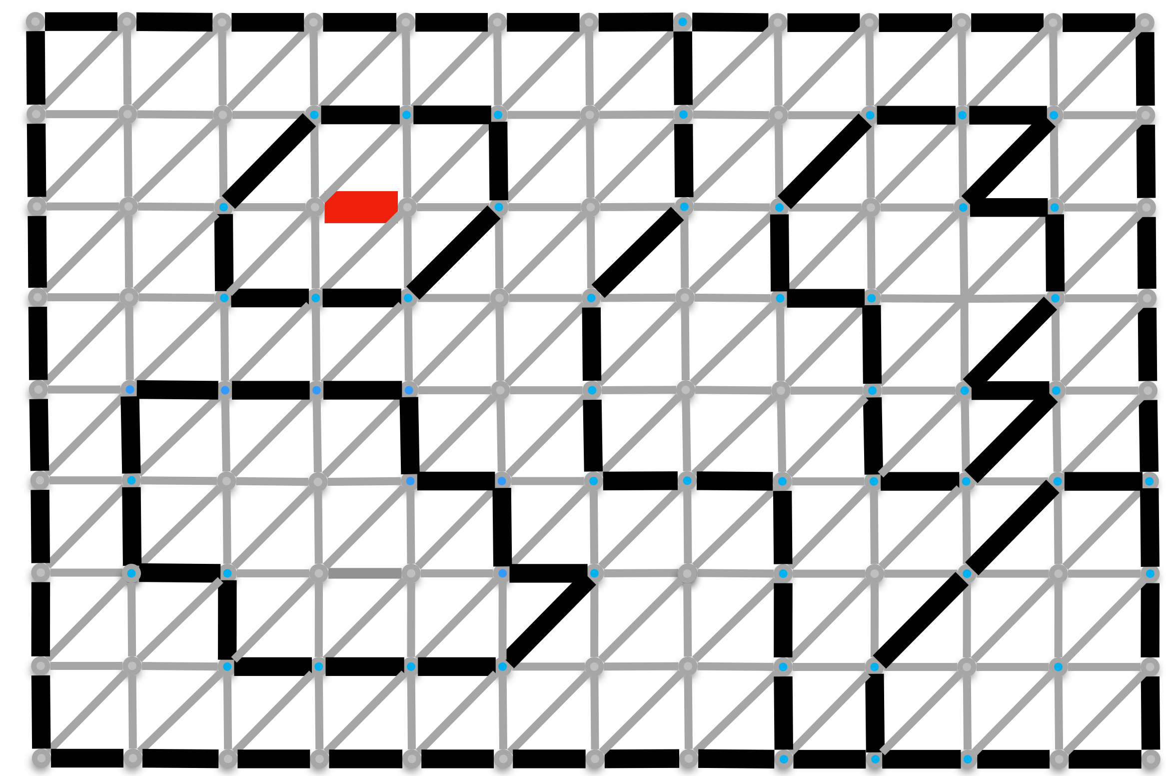


High-level approach for diameter



- compute an r-division
- three types of distance:
  - between a vertex and a boundary vertex $O(n^2/r^{1/2})$ time
  - between two vertices inside the same piece $O(nr)$ time
  - between two vertices in different pieces
    - preprocess each piece for VDs $\tilde{O}(n/r \cdot r) = \tilde{O}(nr)$
    - use a VD for each vertex and each piece $\tilde{O}(n^2/r^{1/2})$ time
- setting $r = n^{2/3}$ yields total running time of $\tilde{O}(n^{5/3})$

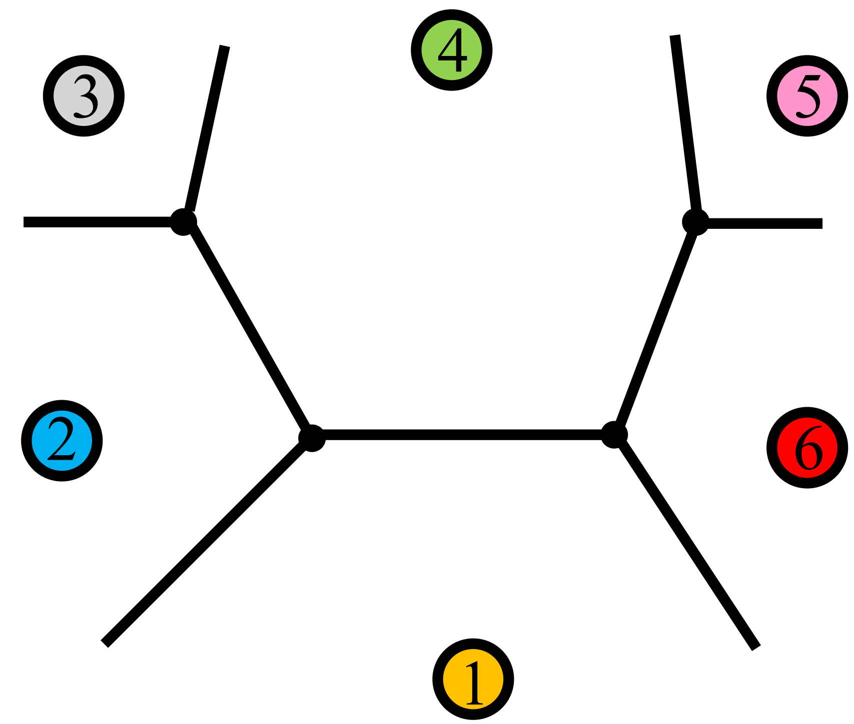
What happens upon an update?



- When edge e in piece P is updated
  - between a vertex and a boundary vertex — $\tilde{O}(n^2/r^{1/2})$ time
  - between two vertices inside the same piece — ~~$\tilde{O}(nr)$~~ $\tilde{O}(nr^{2/3})$ time
  - between two vertices in different pieces
    - preprocess ~~each~~ only affected piece for VDs — $\tilde{O}(r^2)$ ~~$= \tilde{O}(nr)$~~
    - use a VD for each vertex and each piece — $\tilde{O}(n^2/r^{1/2})$ time
- setting $r =$ ~~$n^{2/3}$~~ $n^{4/5}$ yields update time of ~~$\tilde{O}(n^{5/3})$~~ $\tilde{O}(n^{1.6})$

Updating an existing VD

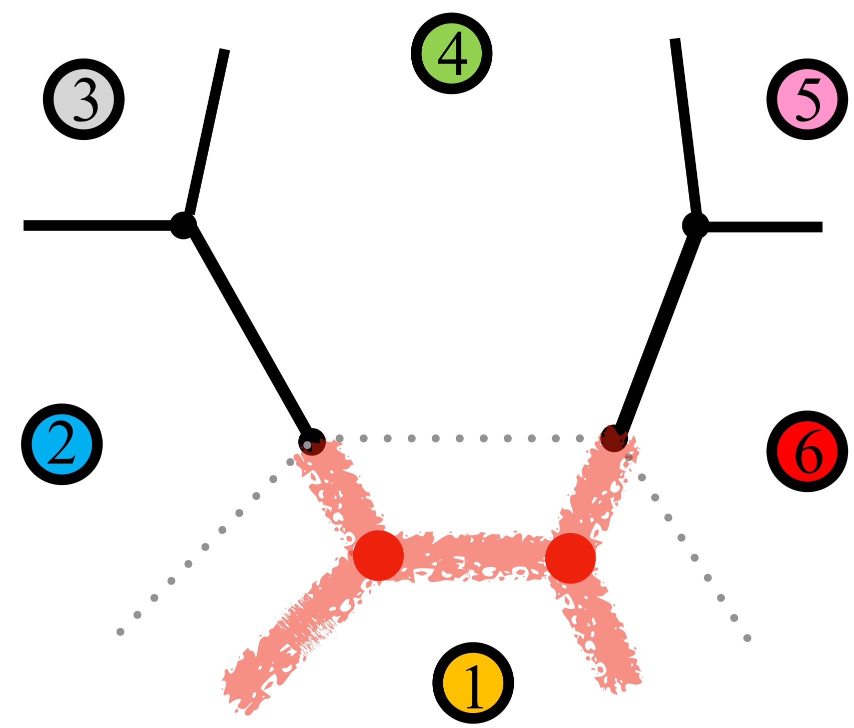
- When an edge e fails distances in the graph only increase.
- So in a piece not containing e, additive distances of some sites may increase



weight of this site increases

Updating an existing VD

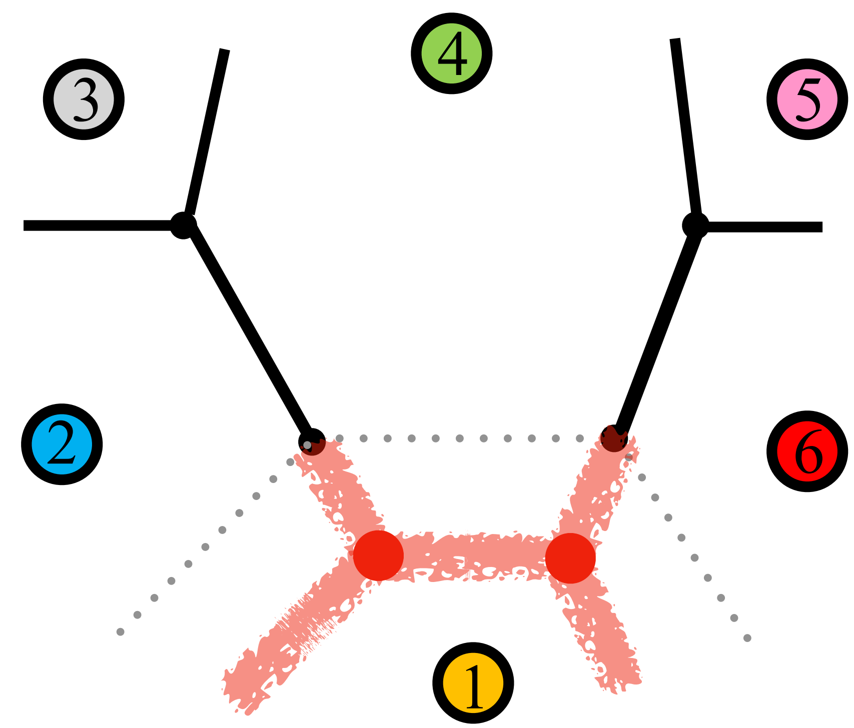
- When an edge e fails distances in the graph only increase.
- So in a piece not containing e, additive distances of some sites may increase



weight of this site increases

- We show how to update the VD in time proportional to the number of neighbors of the cells whose sites have their additive weight increased.

Replacement diameter in unweighted undirected planar graphs



- Goal: compute the diameter after deleting each edge of G (individually)
- Observation: for a given VD, the additive weight of each site is increased at most D times (rather than n)
- Sum of number of neighbors over all sites is order of the number of sites
- Leads to an $n^{7/3+o(1)}$ time algorithm for replacement diameter (instead of naive $\tilde{O}(n^{8/3})$)

What Else Can Voronoi Diagrams Do For Diameter In Planar Graphs?

- Can we improve on $\tilde{O}(n^{5/3})$ for static planar diameter?
  - can we get faster algorithms parameterized by the diameter D also for $n^{1/3} \leq D \leq n^{2/3}$
- Can we dynamically maintain the diameter under updates of the planar graph?
  - faster, for more general dynamic updates?
- Can we prove matching lower bounds for planar diameter?