Algorithms for Voronoi Construction

Naïve Approach for VD construction

- Construction of its Voronoi polygons one at a time
- Since each Voronoi polygon is intersection of n-1 half lines, each polygon can be constructed in O(n log n) time
- As there are n Voronoi polygons/ regions, overall time to construct a Voronoi diagram is

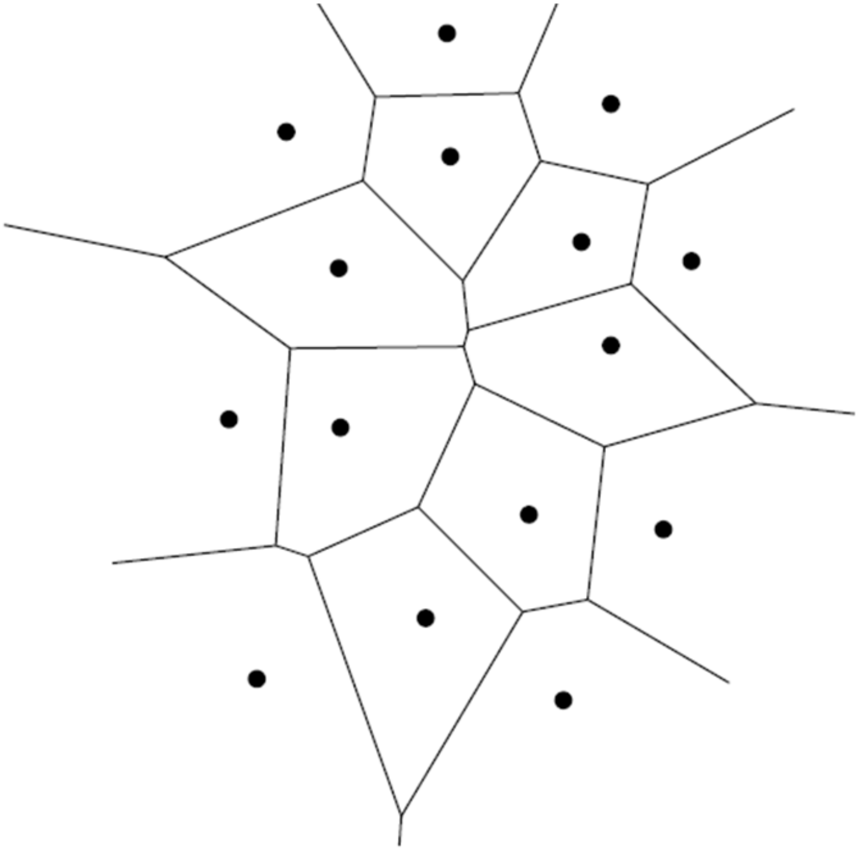
  O (n^2 log n)
- Try for a better algorithm

Incremental Algorithm

- How does an Incremental Algorithm for Voronoi diagram work?

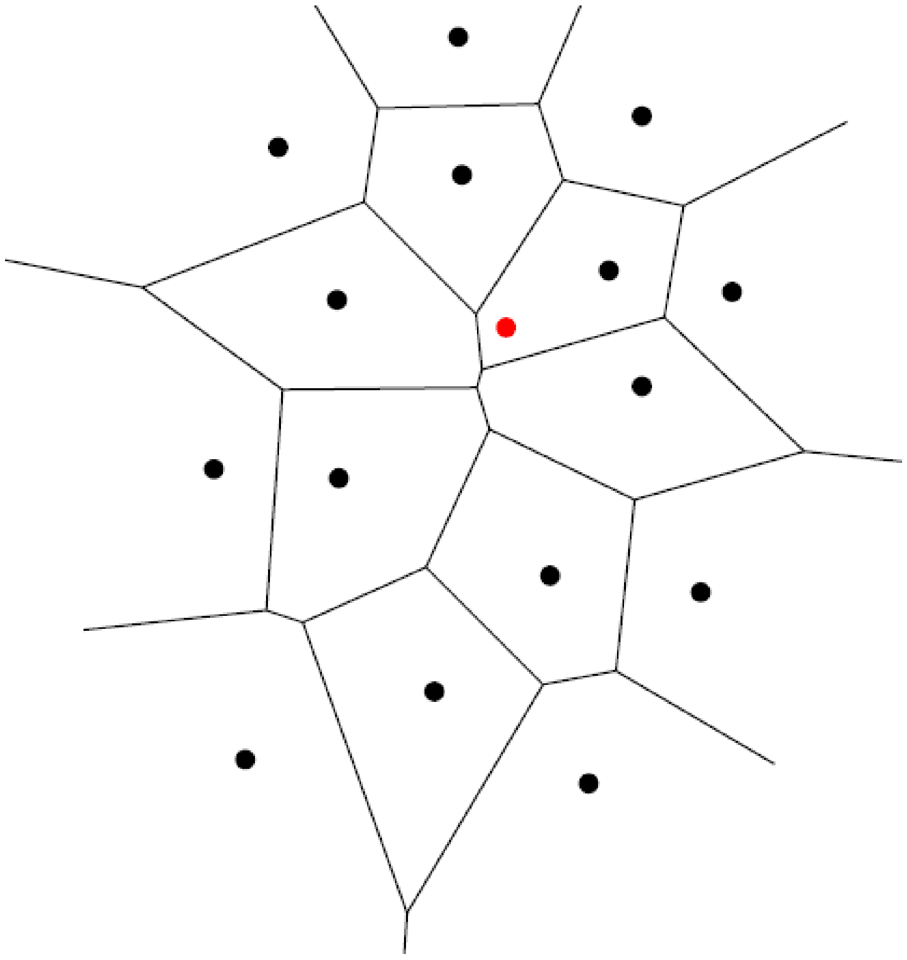
Incremental Algorithm

- Starts with a Voronoi diagram of $\{p_1, p_2, p_3, \ldots, p_i\}$



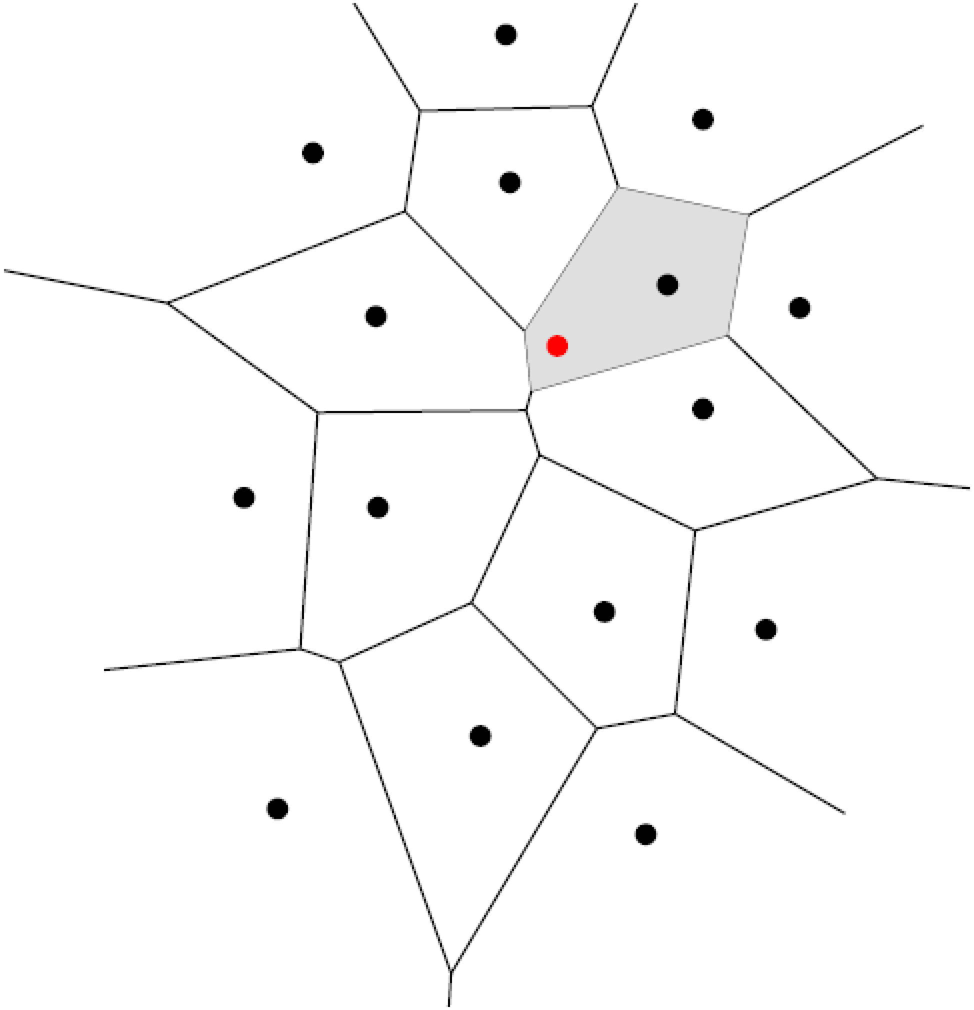
Incremental Algorithm –(contd.)

- Add a point p_{i+1}



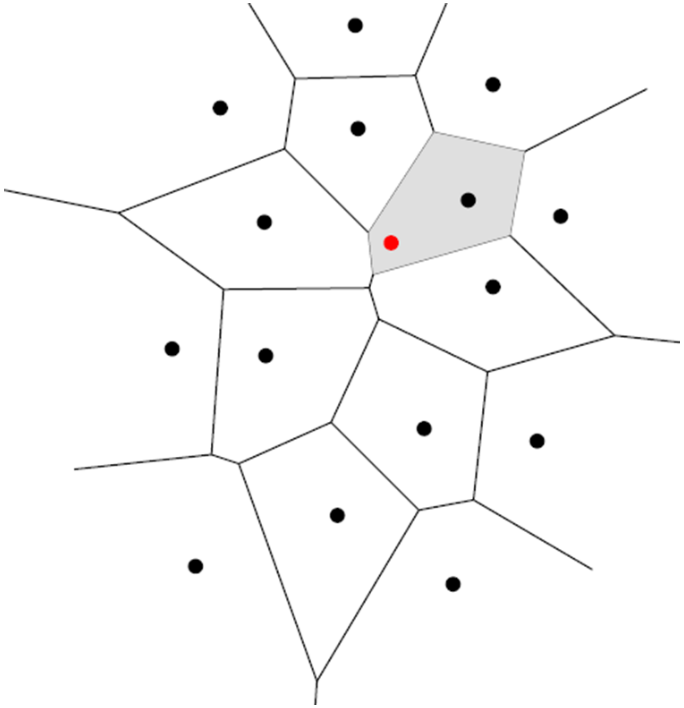
Incremental Algorithm –(contd.)

- Explore all possibilities to find the point p_j closest to p_{i+1}



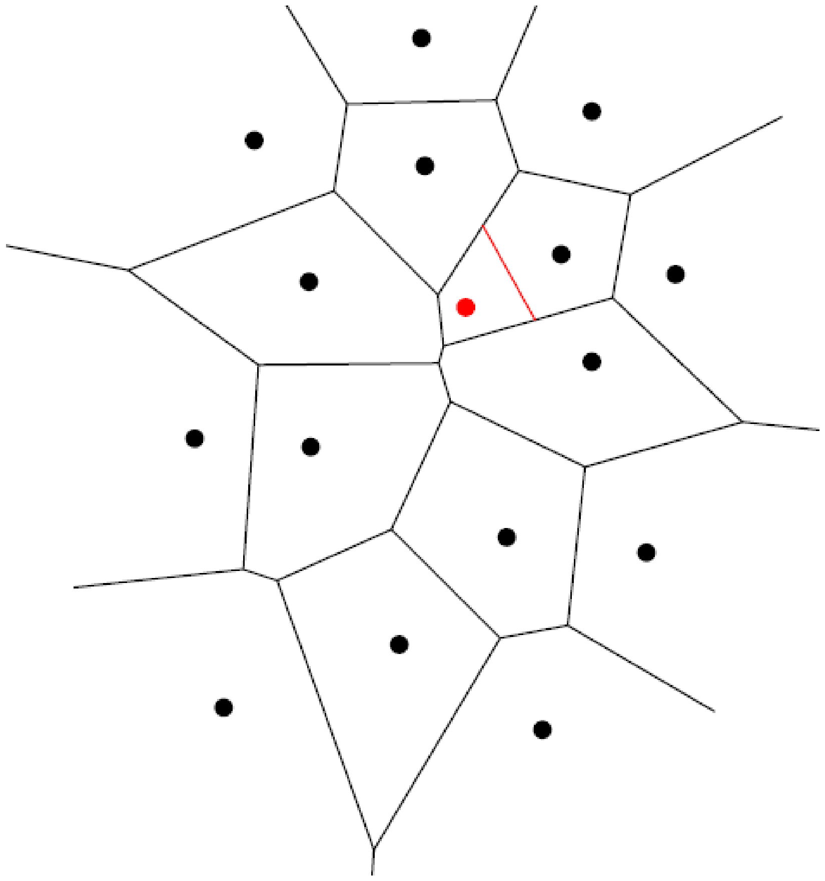
Incremental Algorithm –(contd.)

- Compute p_{i+1}'s Voronoi polygon/ region
- Bisectors of every closest pair of points



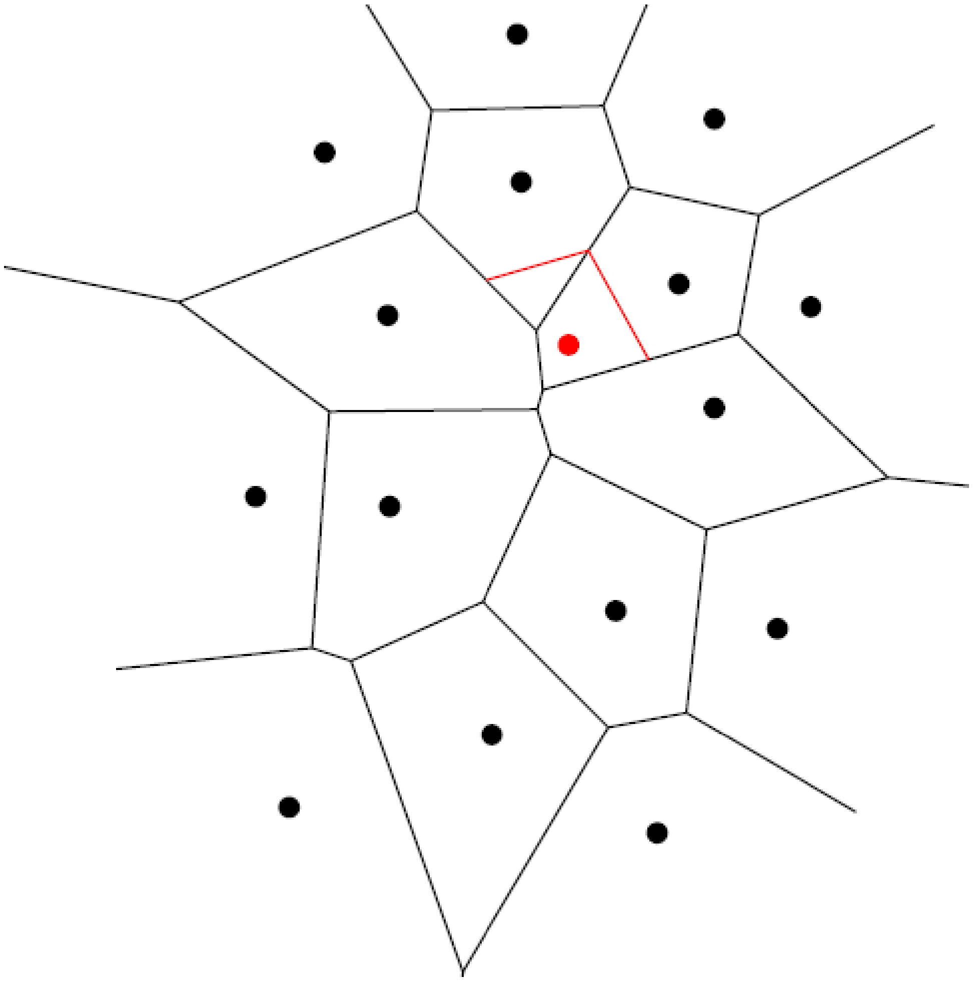
Incremental Algorithm –(contd.)

- Build its boundary starting from $b_{i+1,j}$



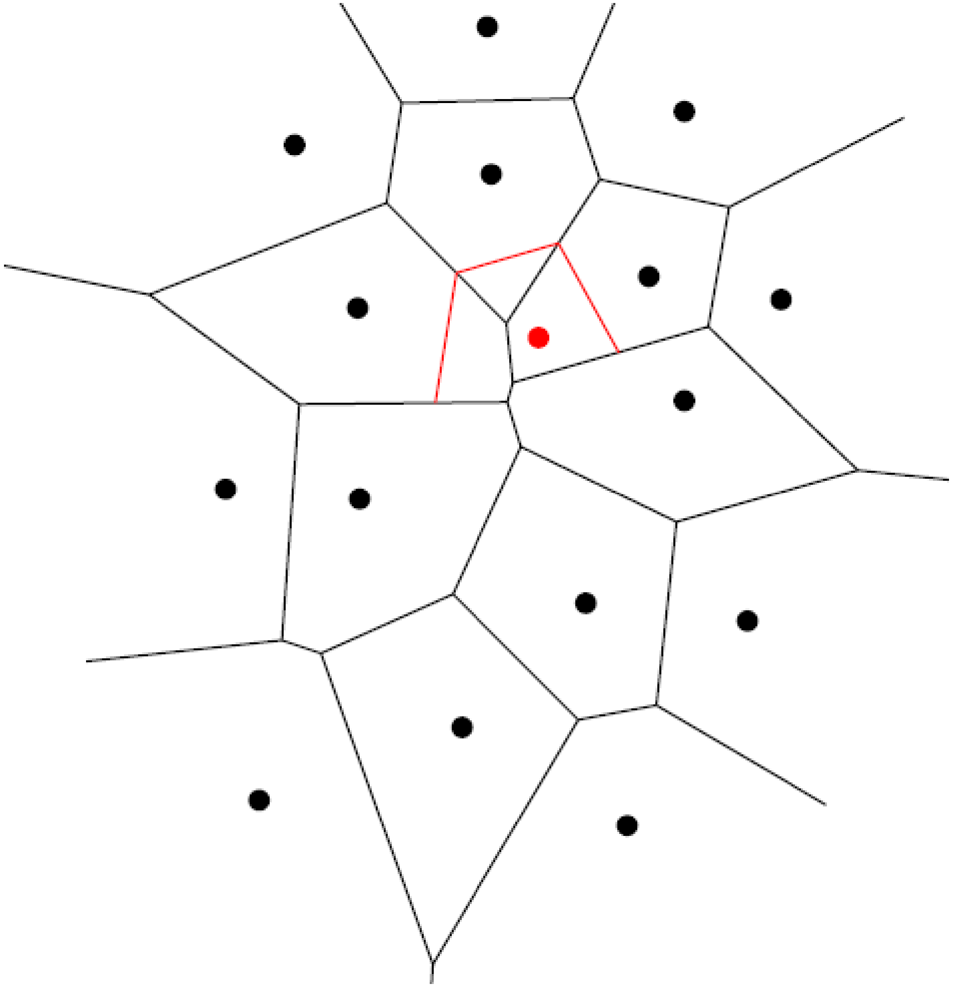
Incremental Algorithm –(contd.)

- Build its other boundaries



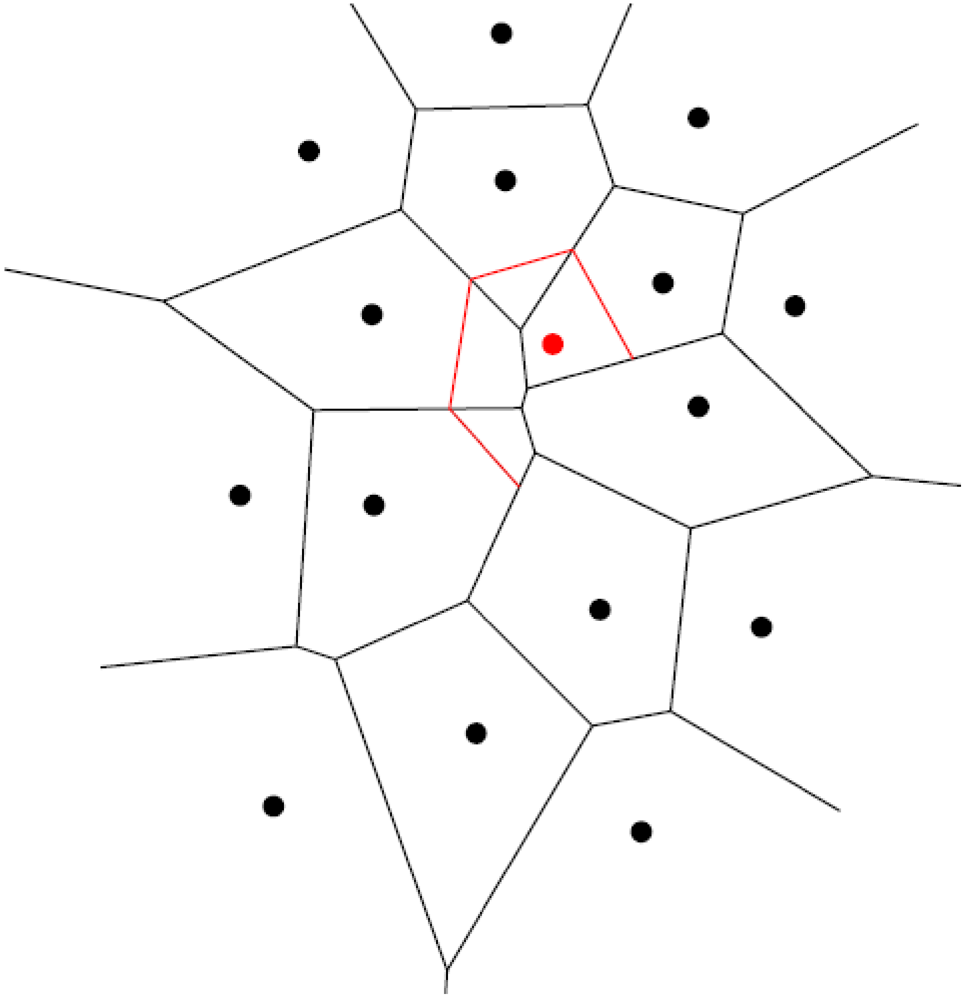
Incremental Algorithm –(contd.)

- Build its other boundaries



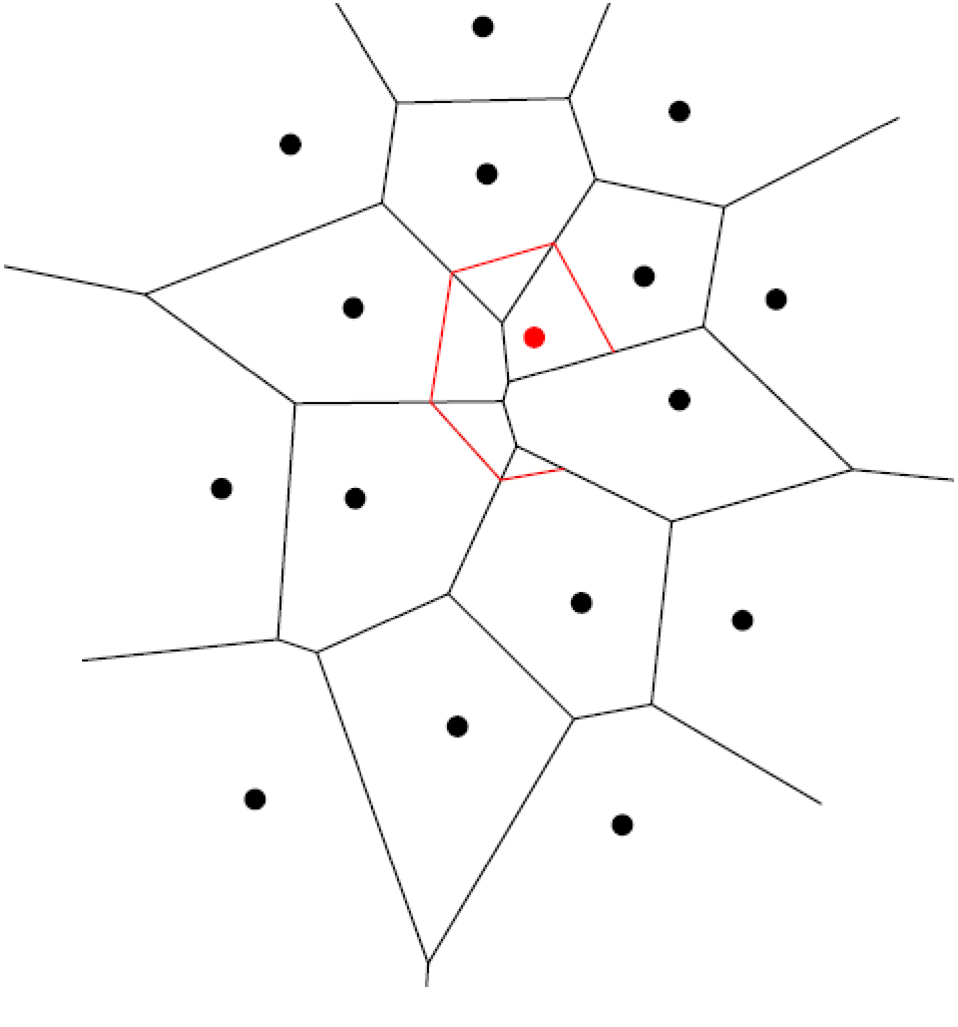
Incremental Algorithm –(contd.)

- Build its other boundaries



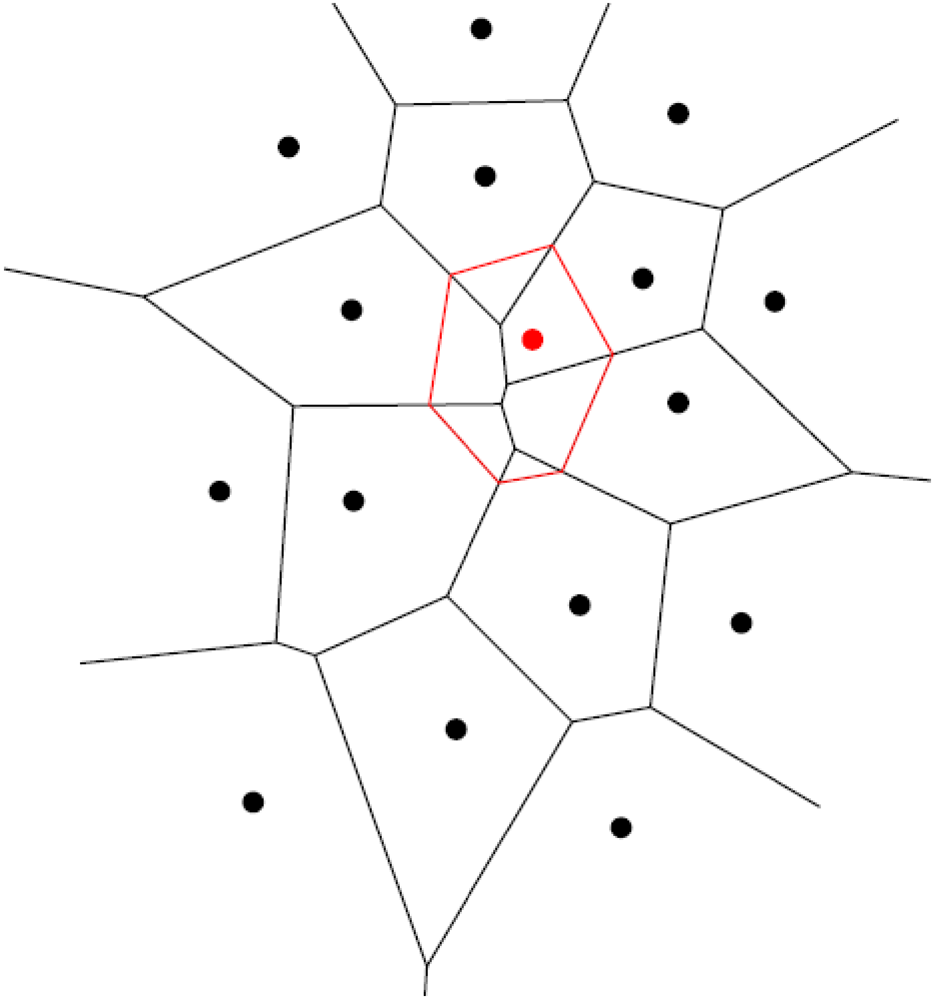
Incremental Algorithm –(contd.)

- Build its other boundaries



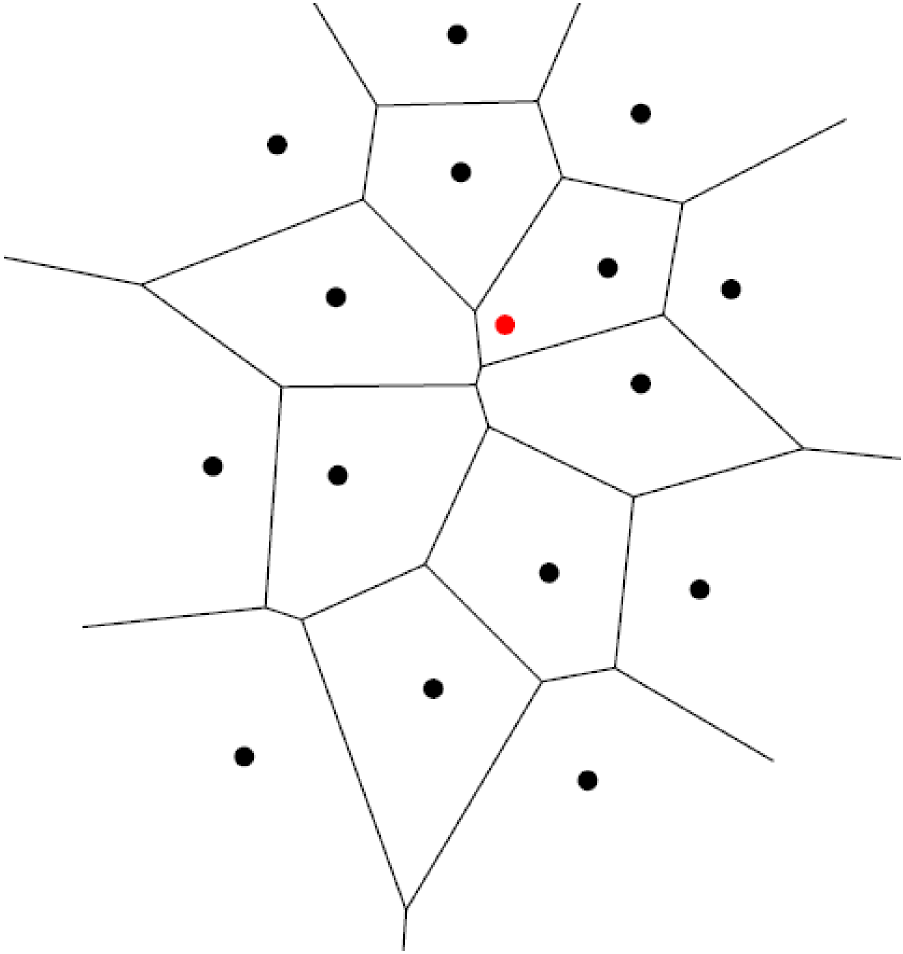
Incremental Algorithm –(contd.)

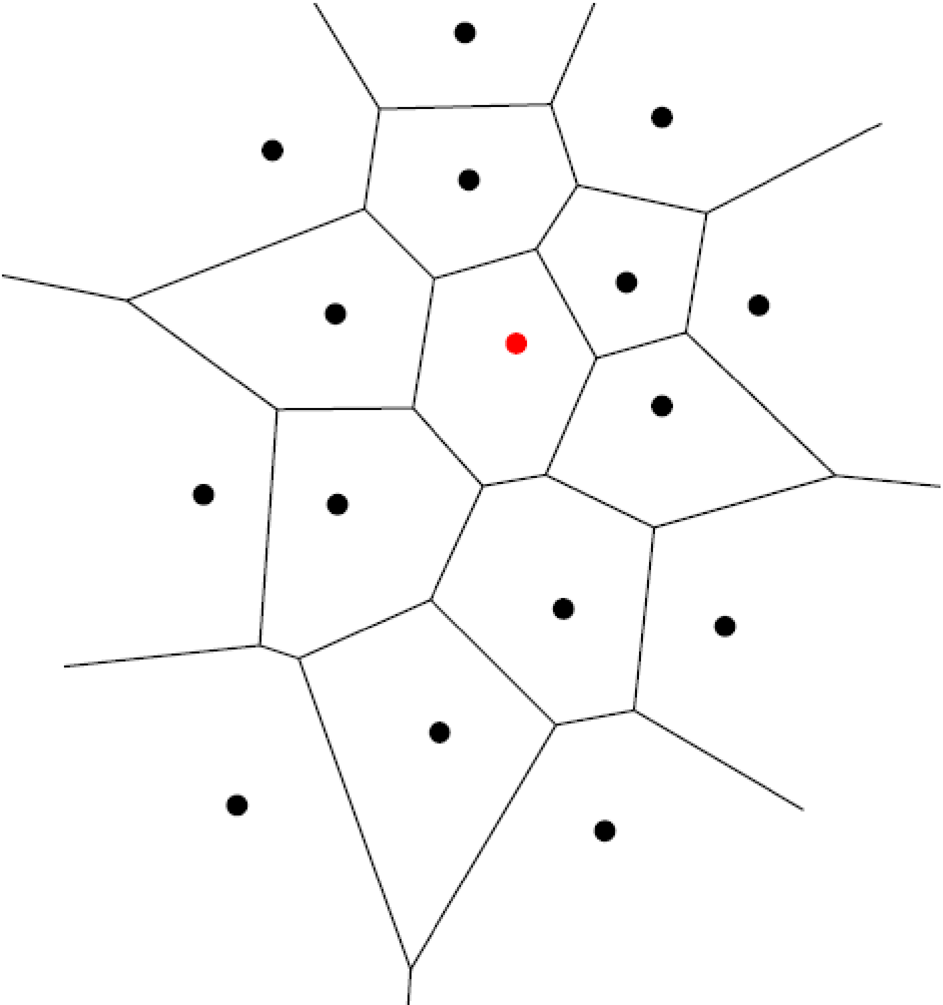
- Build its other boundaries



Incremental Algorithm –(contd.)

- Initial diagram with the point and the VD after the algorithm





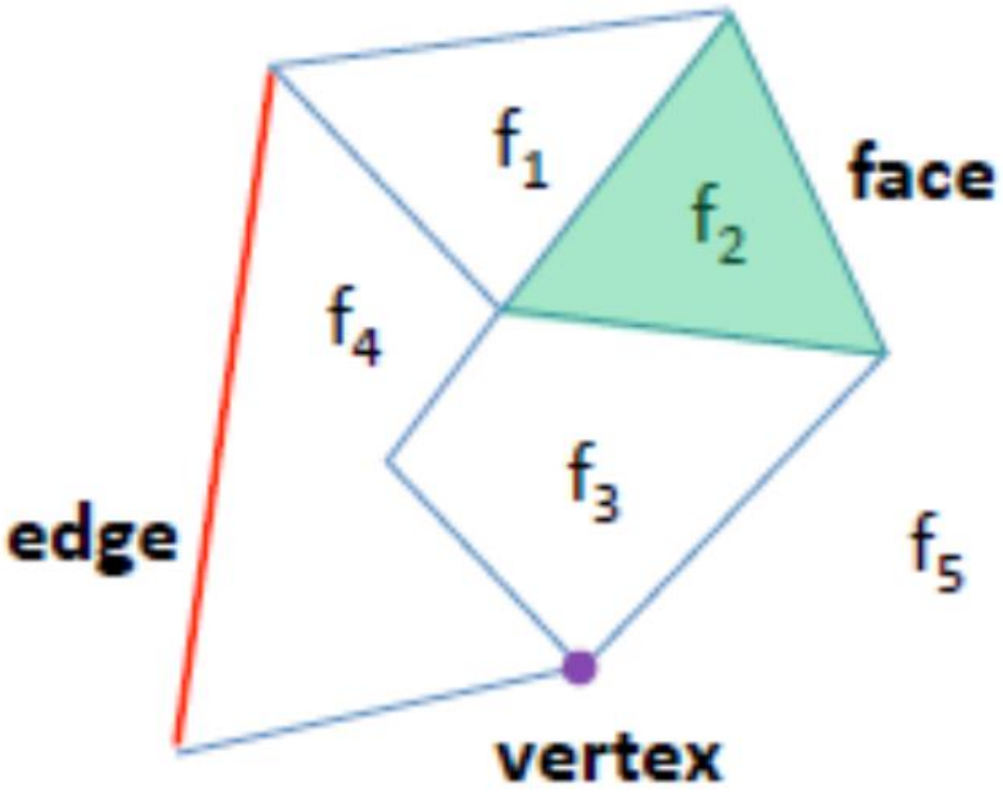
Incremental Algorithm –(contd.)

- To build the Voronoi polygon/ region of p_{i+1} , we use a data structure called Doubly Connected Edge List (DCEL)
- DCEL is proposed by Muller and Preparata
- DCEL is also known as half edge data structure

DCEL

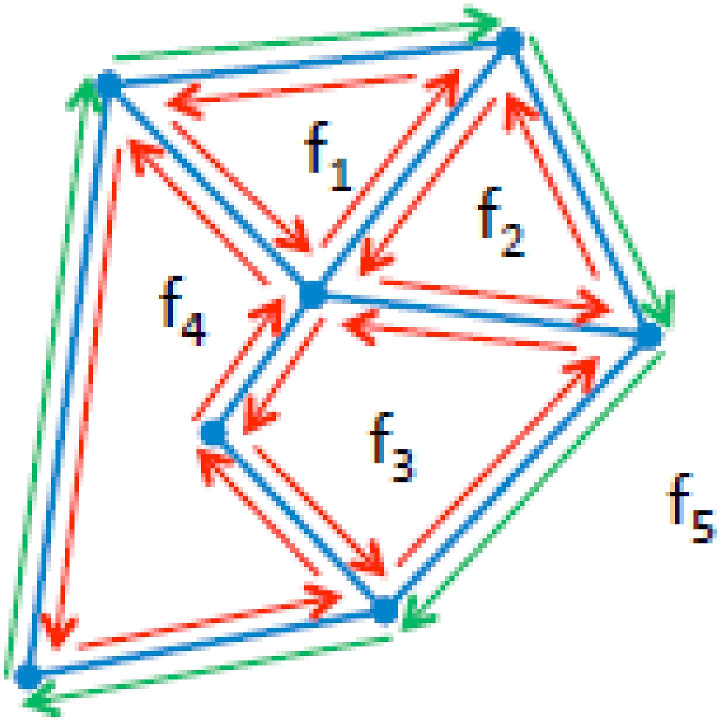
- DCEL is one of the most commonly used representations for planar subdivisions such as Voronoi diagrams.
- It is an edge-based structure which links together three sets of records:
  - **Vertex**
  - **Edge**
  - **Face**
- It facilitates traversing the faces of planar subdivision, visiting all the edges around a given vertex

DCEL



- **Record for each face, edge and vertex**

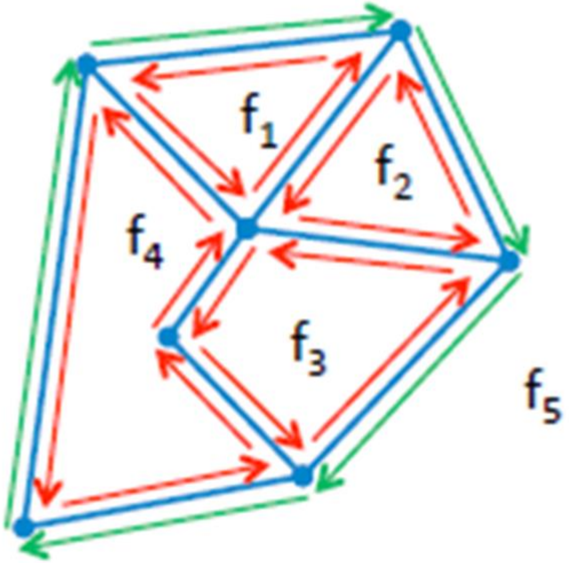
DCEL



- Edges are oriented counterclockwise inside each face
- Since each edge is shared by two faces, each edge is replaced by two half edges, one for each face

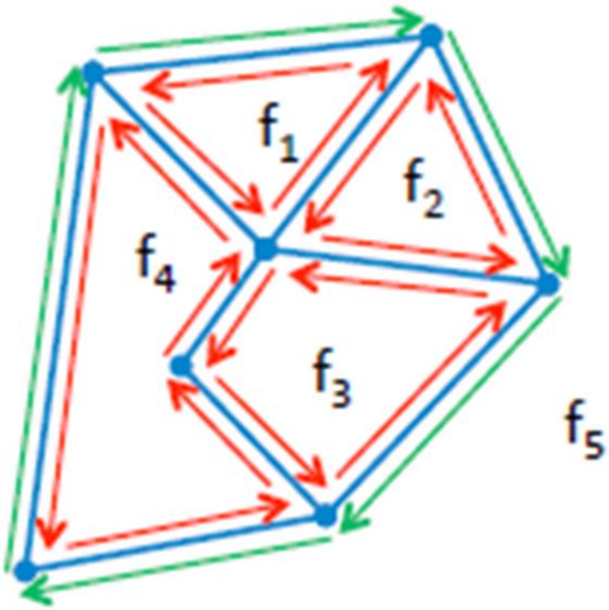
Vertex record

- **The vertex record of a vertex v stores:**
- Coordinates of v
- A pointer IncidentEdge(v)
  - – To an arbitrary half edge that has v as its origin



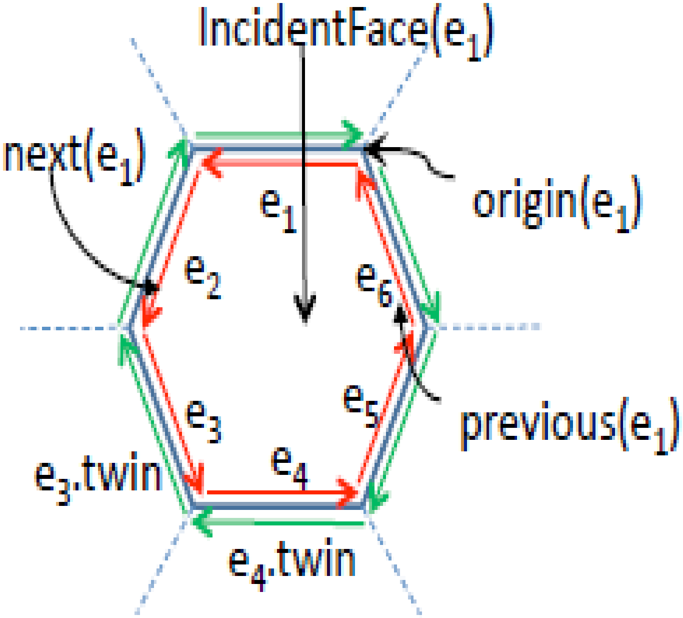
Face record

- **Face record of a face f stores:**
- A pointer to some half edge on its boundary
  - Which can be used as a starting point to traverse f in a counterclockwise order



Half-Edge Record

- **The half-edge record of a half-edge e stores pointer to** :
- Origin(e)
- Twin of e, e.twin or twin(e)
- The face to its left, IncidentFace(e)
- Next half edge on the boundary of IncidentFace(e), Next(e)
- Previous half-edge, Previous(e)



References

- https://dccg.upc.edu/people/vera/wp-content/uploads/2013/06/GeoC-Voronoi-algorithms.pdf by Professor Vera Sacristan
- de Berg, Van Krevald, Overmars, and Schwarzkpf, *Computational Geometry Algorithms and Applications,* Springer Third Edition, 1998
- F.P. Preparata & M.I. Shamos, *Computational Geometry An Introduction,* Springer International Edition, 1985

THANK YOU