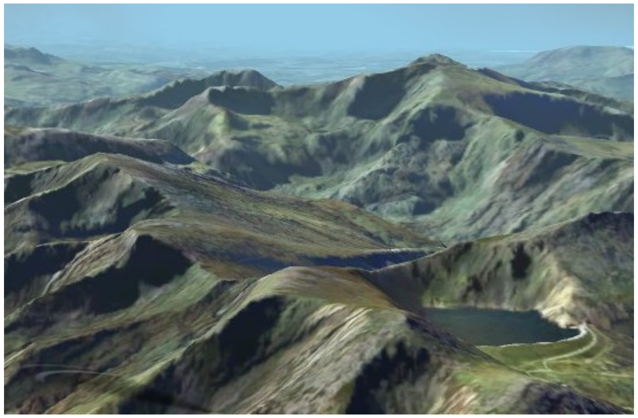
6.886 Delaunay Triangulation

Slides Credit: UFL COT5520, CIS4930 Spring 18
with minor modifications
Reference: Computational Geometry, Algorithms and Applications, 3rd Edition
yiqiuw@mit.edu

Motivation: Terrains

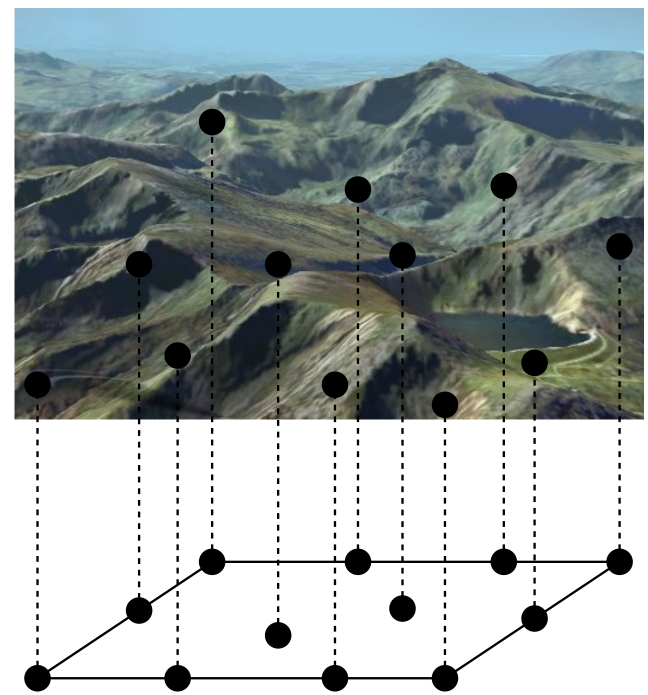
- a terrain is the graph of a function $f : A \subset \mathbb{R}^2 \to \mathbb{R}$
- we know only height values for a set of measurement points
- how can we interpolate the height at other points?
- using a triangulation



2

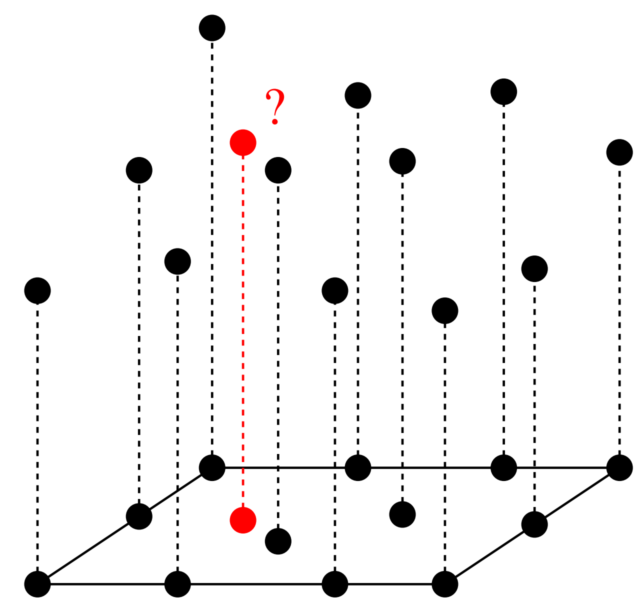
Motivation: Terrains

- a terrain is the graph of a function $f : A \subset \mathbb{R}^2 \to \mathbb{R}$
- we know only height values for a set of measurement points
- how can we interpolate the height at other points?
- using a triangulation



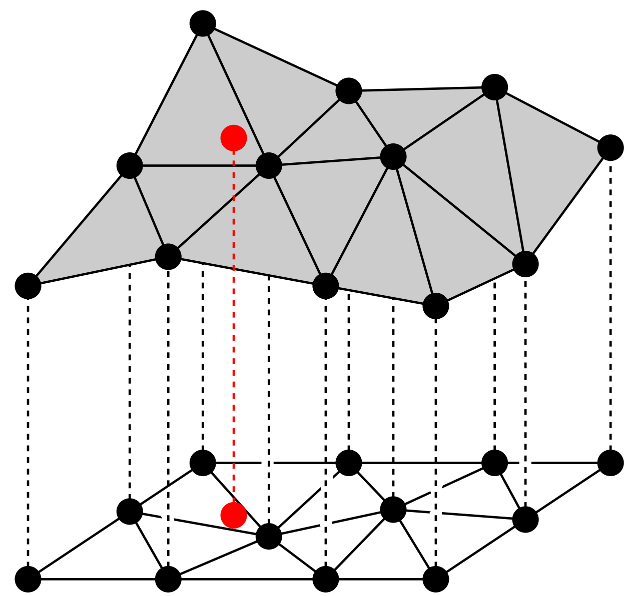
Motivation: Terrains

- a terrain is the graph of a function $f : A \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$
- we know only height values for a set of measurement points
- how can we interpolate the height at other points?
- using a triangulation



Motivation: Terrains

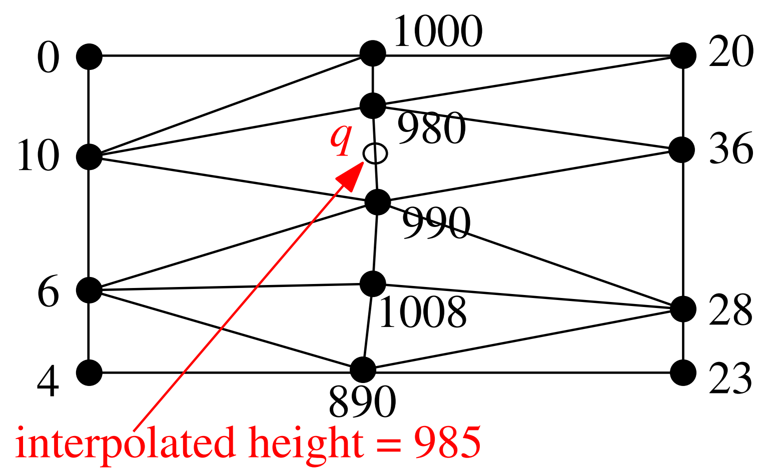
- a terrain is the graph of a function $f : A \subset \mathbb{R}^2 \to \mathbb{R}$
- we know only height values for a set of measurement points
- how can we interpolate the height at other points?
- using a triangulation

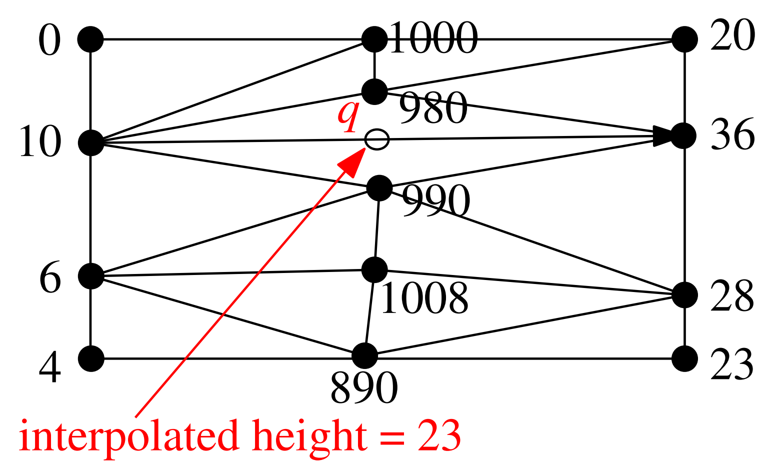


5

Motivation: Terrains

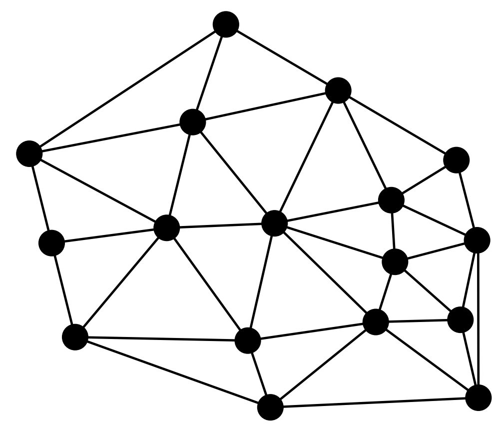
- a terrain is the graph of a function $f : A \subset \mathbb{R}^2 \rightarrow \mathbb{R}$
- we know only height values for a set of measurement points
- how can we interpolate the height at other points?
- using a triangulation
  – but which?





Triangulation

Let $P = \{p_1, \ldots, p_n\}$ be a point set. A triangulation of P is a maximal planar subdivision with vertex set P.



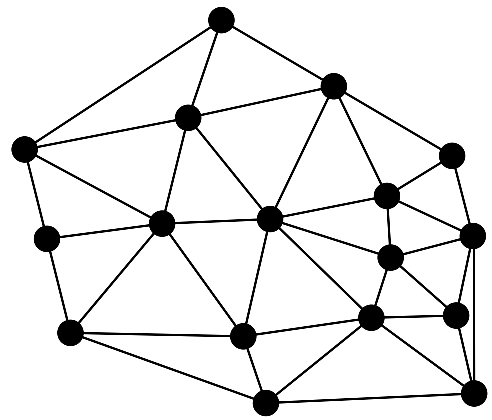
Triangulation

Let $P = \{p_1, \ldots, p_n\}$ be a point set. A triangulation of P is a maximal planar subdivision with vertex set P.

Complexity:

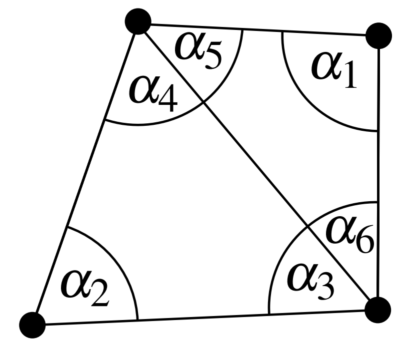
- $2n - 2 - k$ triangles
- $3n - 3 - k$ edges

where k is the number of points in P on the convex hull of P.



Angle Vector of a Triangulation

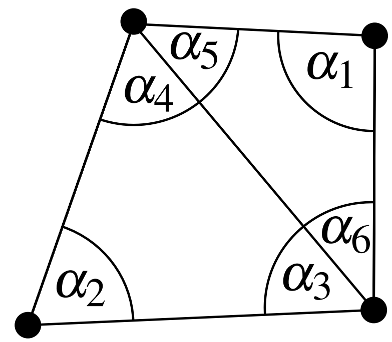
- Let $\mathcal{T}$ be a triangulation of P with m triangles and $3m$ vertices. Its angle vector is $A(\mathcal{T}) = (\alpha_1, \ldots, \alpha_{3m})$ where $\alpha_1, \ldots, \alpha_{3m}$ are the angles of $\mathcal{T}$ sorted by increasing value.
- Let $\mathcal{T}'$ be another triangulation of P. We define $A(\mathcal{T}) > A(\mathcal{T}')$ if $A(\mathcal{T})$ is lexicographically larger than $A(\mathcal{T}')$.
- $\mathcal{T}$ is angle optimal if $A(\mathcal{T}) \geq A(\mathcal{T}')$ for all triangulations $\mathcal{T}'$ of P.



$A(\mathcal{T}) = (\alpha_1, \ldots, \alpha_6)$

Angle Vector of a Triangulation

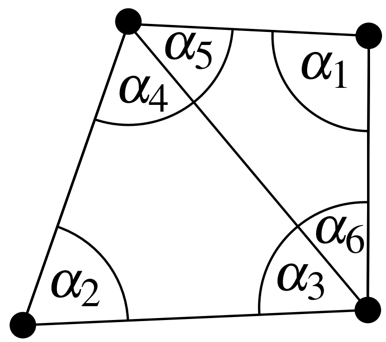
- Let $\mathcal{T}$ be a triangulation of P with m triangles and $3m$ vertices. Its angle vector is $A(\mathcal{T}) = (\alpha_1, \ldots, \alpha_{3m})$ where $\alpha_1, \ldots, \alpha_{3m}$ are the angles of $\mathcal{T}$ sorted by increasing value.
- Let $\mathcal{T}'$ be another triangulation of P. We define $A(\mathcal{T}) > A(\mathcal{T}')$ if $A(\mathcal{T})$ is lexicographically larger than $A(\mathcal{T}')$.
- $\mathcal{T}$ is angle optimal if $A(\mathcal{T}) \geq A(\mathcal{T}')$ for all triangulations $\mathcal{T}'$ of P.



$A(\mathcal{T}) = (\alpha_1, \ldots, \alpha_6)$

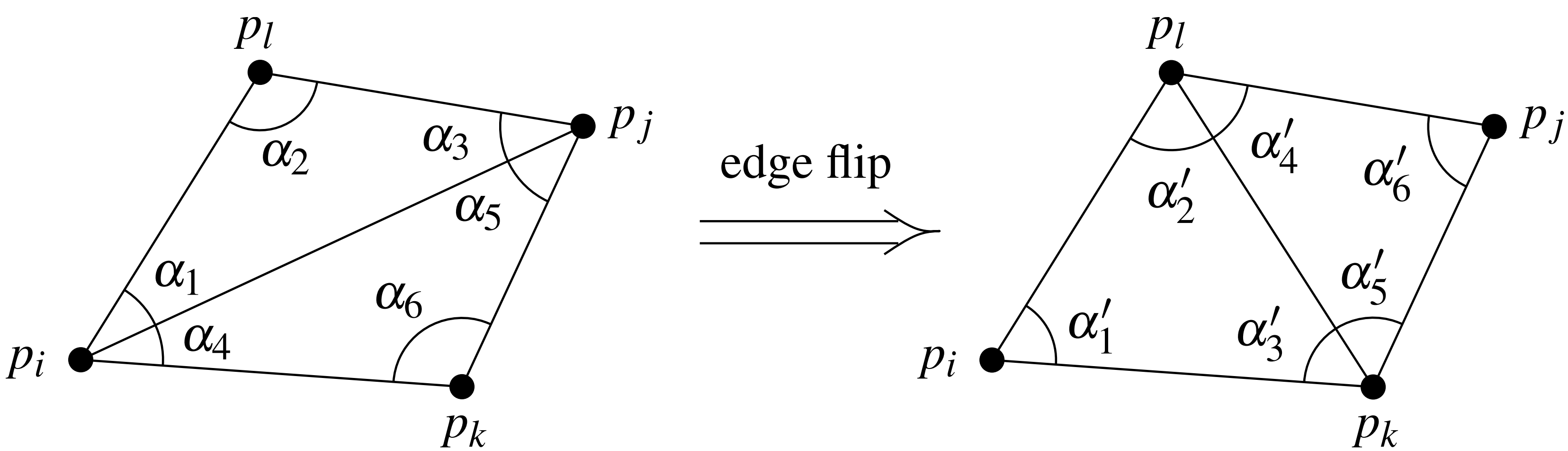
Angle Vector of a Triangulation

- Let $\mathcal{T}$ be a triangulation of P with m triangles and $3m$ vertices. Its angle vector is $A(\mathcal{T}) = (\alpha_1, \ldots, \alpha_{3m})$ where $\alpha_1, \ldots, \alpha_{3m}$ are the angles of $\mathcal{T}$ sorted by increasing value.
- Let $\mathcal{T}'$ be another triangulation of P. We define $A(\mathcal{T}) > A(\mathcal{T}')$ if $A(\mathcal{T})$ is lexicographically larger than $A(\mathcal{T}')$.
- $\mathcal{T}$ is angle optimal if $A(\mathcal{T}) \geq A(\mathcal{T}')$ for all triangulations $\mathcal{T}'$ of P.



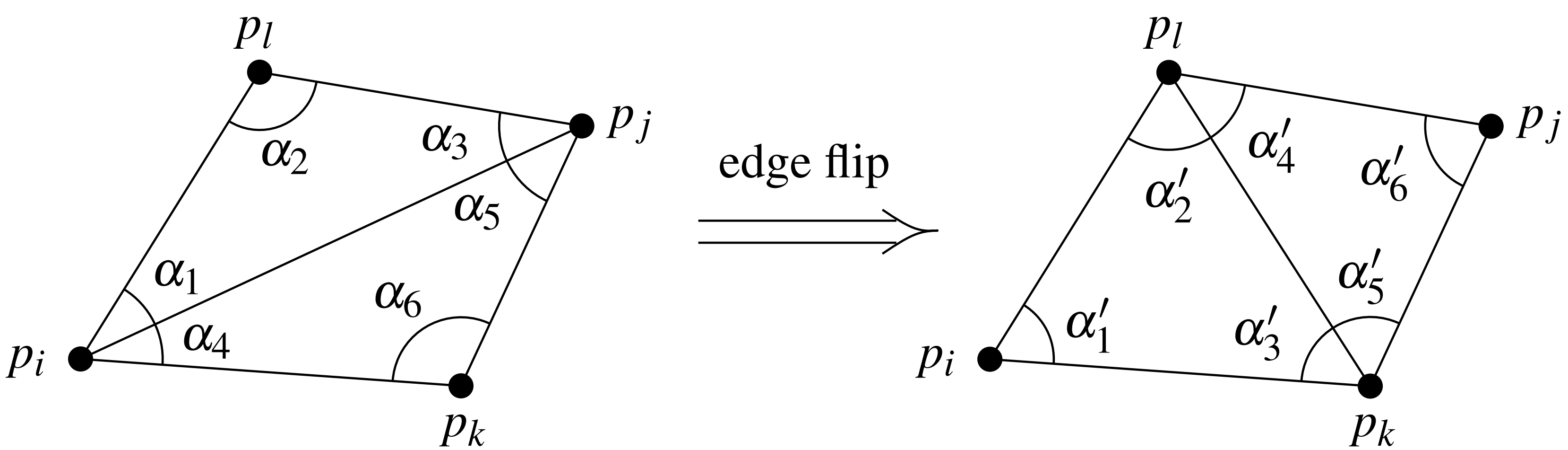
$A(\mathcal{T}) = (\alpha_1, \ldots, \alpha_6)$

Edge Flipping



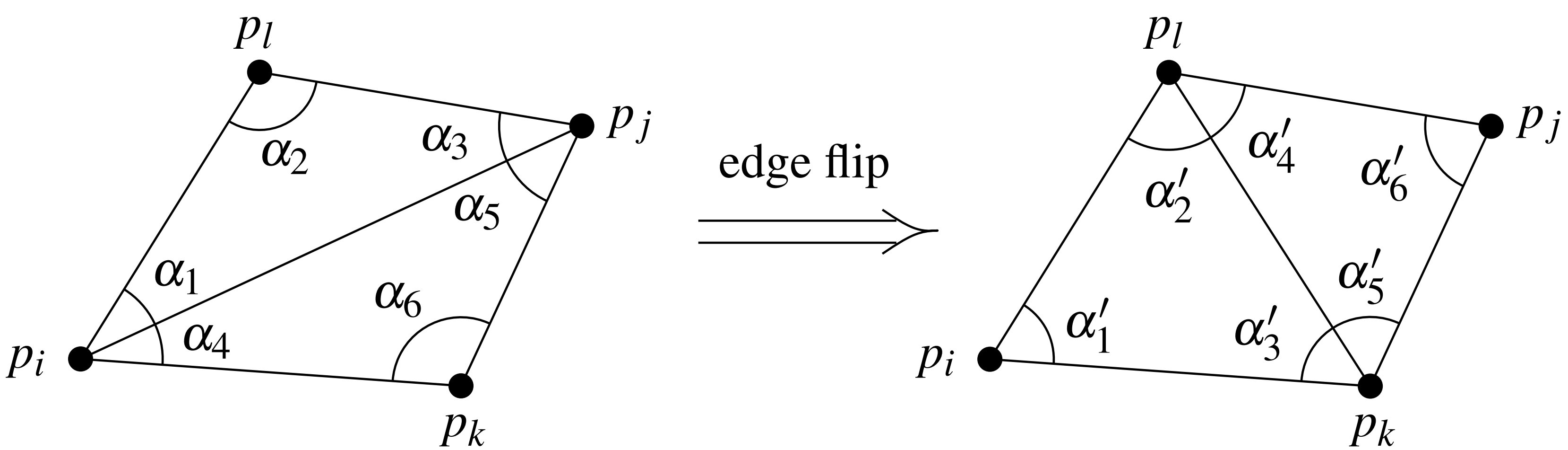
- Change in angle vector: $\alpha_1, \ldots, \alpha_6$ are replaced by $\alpha'_1, \ldots, \alpha'_6$.
- The edge $e = \overline{p_i p_j}$ is illegal if $\min_{1 \leq i \leq 6} \alpha_i < \min_{1 \leq i \leq 6} \alpha'_i$.
- Flipping an illegal edge increases the angle vector.

Edge Flipping



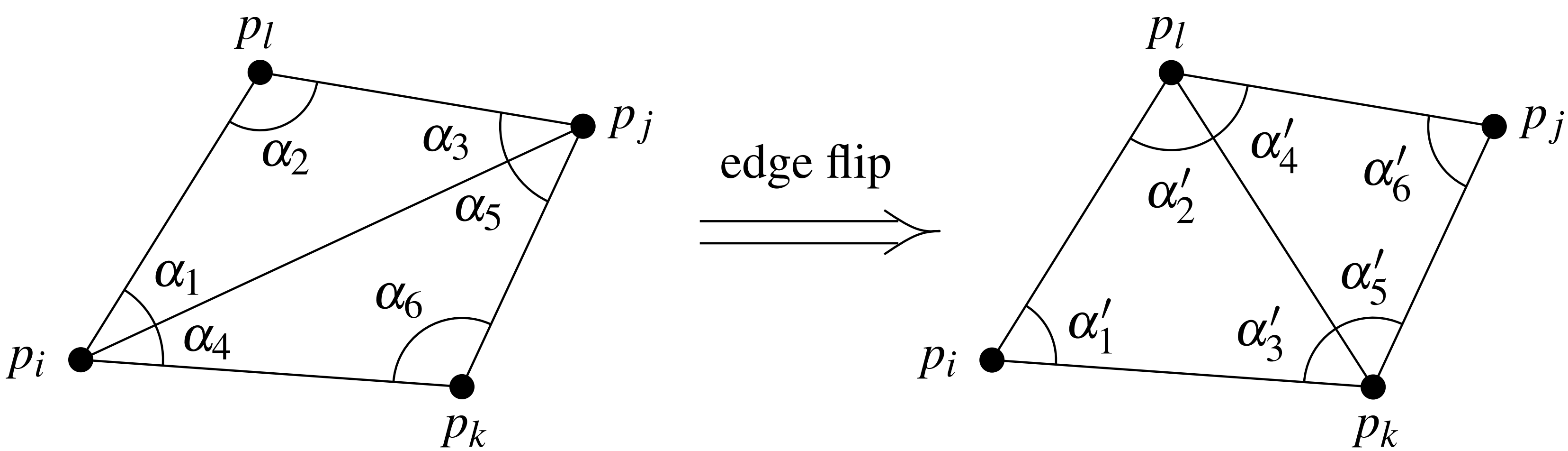
- Change in angle vector:
  $\alpha_1, \dots, \alpha_6$ are replaced by $\alpha'_1, \dots, \alpha'_6$.
- The edge $e = \overline{p_i p_j}$ is illegal if $\min_{1 \le i \le 6} \alpha_i < \min_{1 \le i \le 6} \alpha'_i$.
- Flipping an illegal edge increases the angle vector.

Edge Flipping



- Change in angle vector:
  $\alpha_1, \ldots, \alpha_6$ are replaced by $\alpha'_1, \ldots, \alpha'_6$.
- The edge $e = \overline{p_i p_j}$ is illegal if $\min_{1 \le i \le 6} \alpha_i < \min_{1 \le i \le 6} \alpha'_i$.
- Flipping an illegal edge increases the angle vector.

Edge Flipping



- Change in angle vector:
  $\alpha_1, \ldots, \alpha_6$ are replaced by $\alpha'_1, \ldots, \alpha'_6$.
- The edge $e = \overline{p_i p_j}$ is illegal if $\min_{1 \leq i \leq 6} \alpha_i < \min_{1 \leq i \leq 6} \alpha'_i$.
- Flipping an illegal edge increases the angle vector.

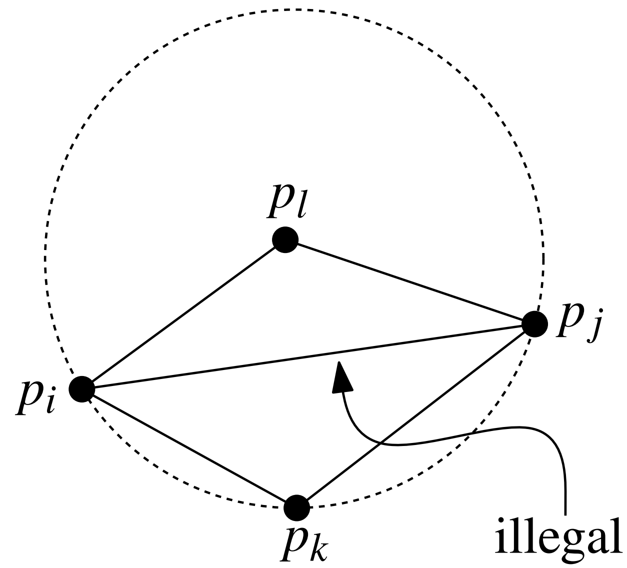
Characterisation of Illegal Edges

How do we determine if an edge is illegal?

Characterisation of Illegal Edges

How do we determine if an edge is illegal?

Lemma: The edge $\overline{p_i p_j}$ is illegal if and only if p_l lies in the interior of the circle C.

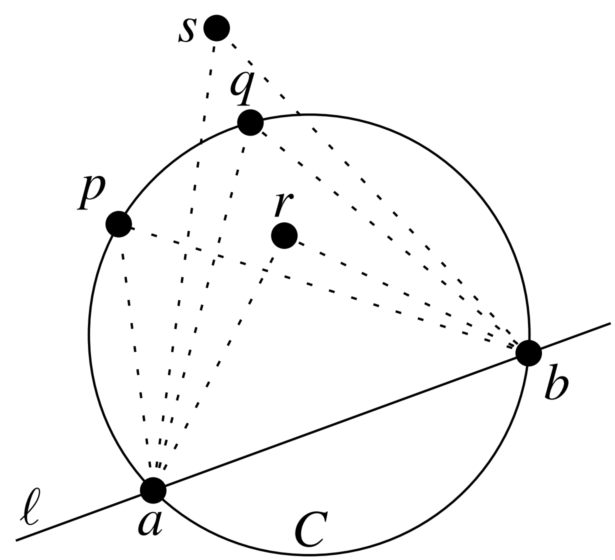


Thales Theorem

Theorem: Let C be a circle, ℓ a line intersecting C in points a and b, and p,q,r,s points lying on the same side of ℓ. Suppose that p,q lie on C, r lies inside C, and s lies outside C. Then

$$\measuredangle arb > \measuredangle apb = \measuredangle aqb > \measuredangle asb,$$

where $\measuredangle abc$ denotes the smaller angle defined by three points a,b,c.

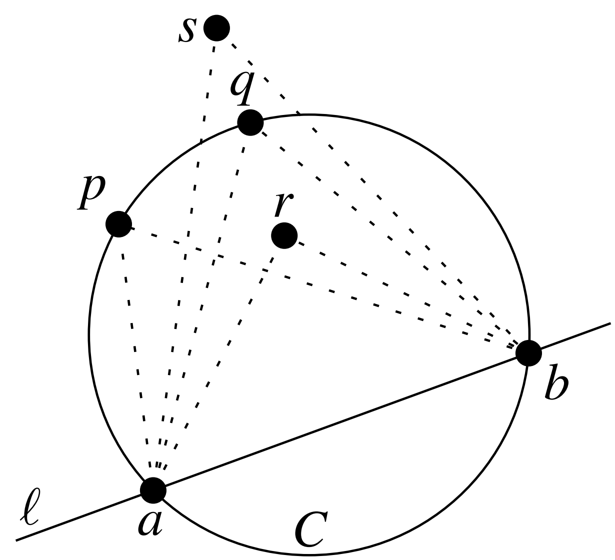


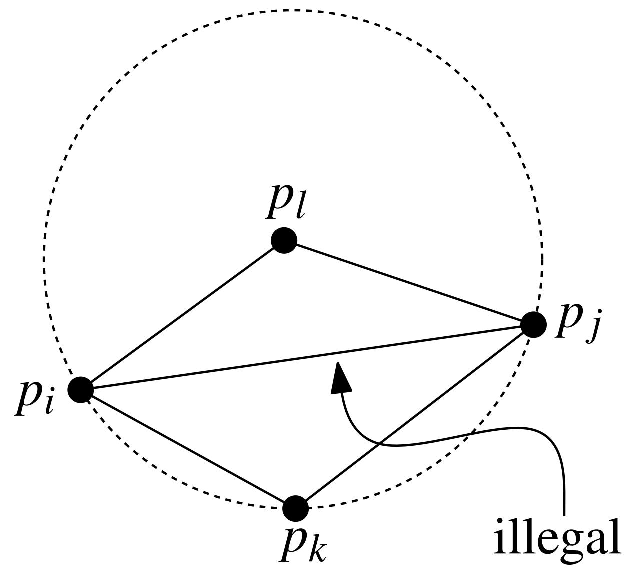
Thales Theorem

Theorem: Let C be a circle, ℓ a line intersecting C in points a and b, and p,q,r,s points lying on the same side of ℓ. Suppose that p,q lie on C, r lies inside C, and s lies outside C. Then

$$\measuredangle arb > \measuredangle apb = \measuredangle aqb > \measuredangle asb,$$

where $\measuredangle abc$ denotes the smaller angle defined by three points a,b,c.



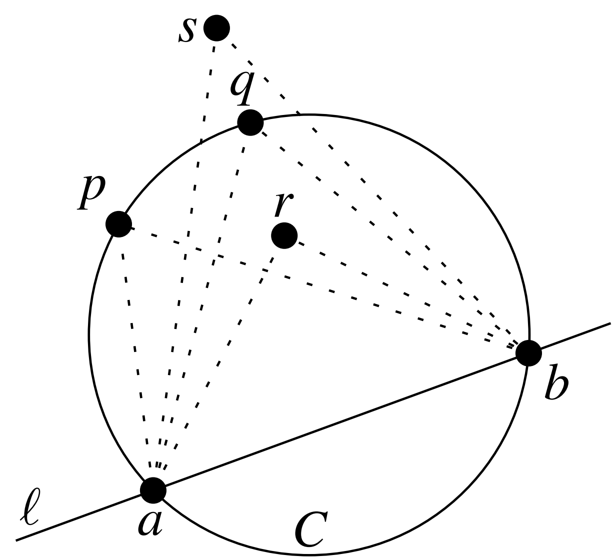


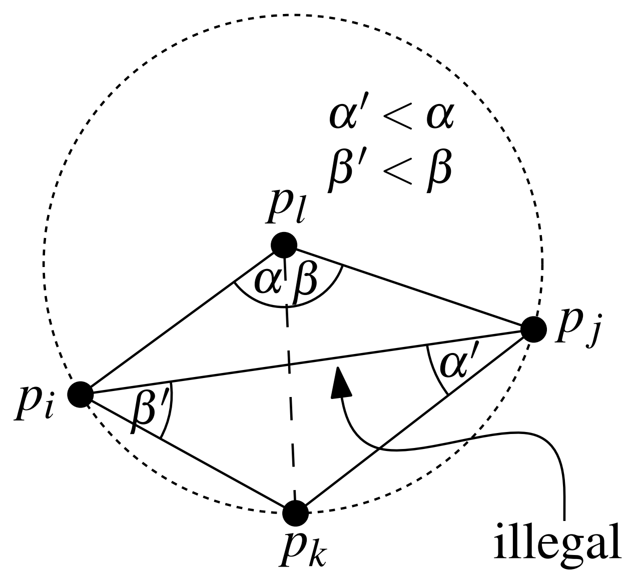
Thales Theorem

Theorem: Let C be a circle, ℓ a line intersecting C in points a and b, and p,q,r,s points lying on the same side of ℓ. Suppose that p,q lie on C, r lies inside C, and s lies outside C. Then

$$\measuredangle arb > \measuredangle apb = \measuredangle aqb > \measuredangle asb,$$

where $\measuredangle abc$ denotes the smaller angle defined by three points a,b,c.





Legal Triangulations

A legal triangulation is a triangulation that does not contain any illegal edge.

Legal Triangulations

A legal triangulation is a triangulation that does not contain any illegal edge.

Algorithm LegalTriangulation($\mathcal{T}$)
Input. A triangulation $\mathcal{T}$ of a point set P.
Output. A legal triangulation of P.
1. **while** $\mathcal{T}$ contains an illegal edge $\overline{p_i p_j}$
2. **do** ($*$ Flip $\overline{p_i p_j}$ $*$)
3. Let $p_i p_j p_k$ and $p_i p_j p_l$ be the two triangles adjacent to $\overline{p_i p_j}$.
4. Remove $\overline{p_i p_j}$ from $\mathcal{T}$, and add $\overline{p_k p_l}$ instead.
5. **return** $\mathcal{T}$

Legal Triangulations

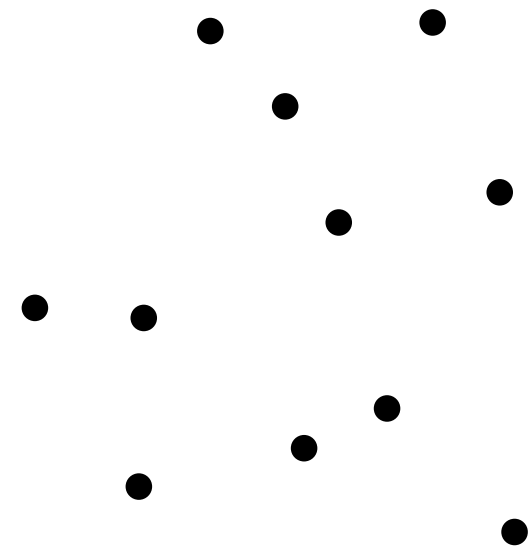
A legal triangulation is a triangulation that does not contain any illegal edge.

Algorithm LegalTriangulation($\mathcal{T}$)
Input. A triangulation $\mathcal{T}$ of a point set P.
Output. A legal triangulation of P.
1. **while** $\mathcal{T}$ contains an illegal edge $\overline{p_i p_j}$
2. **do** (∗ Flip $\overline{p_i p_j}$ ∗)
3. Let $p_i p_j p_k$ and $p_i p_j p_l$ be the two triangles adjacent to $\overline{p_i p_j}$.
4. Remove $\overline{p_i p_j}$ from $\mathcal{T}$, and add $\overline{p_k p_l}$ instead.
5. **return** $\mathcal{T}$

Question: Why does this algorithm terminate?

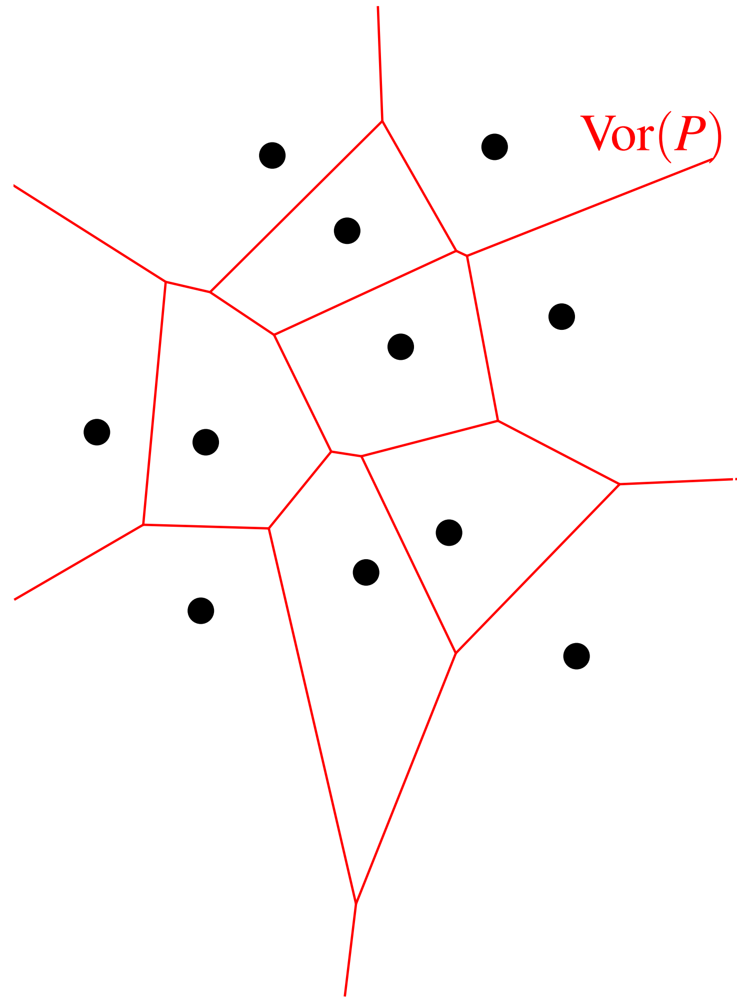
Voronoi Diagram and Delaunay Graph

- Let P be a set of n points in the plane.
- The Voronoi diagram $\text{Vor}(P)$ is the subdivision of the plane into Voronoi cells $\mathcal{V}(p)$ for all $p \in P$.
- Let $\mathcal{G}$ be the *dual graph* of $\text{Vor}(P)$.
- The Delaunay graph $\mathcal{DG}(P)$ is the *straight line embedding* of $\mathcal{G}$.
- **Question:** How can we compute the Delaunay graph?



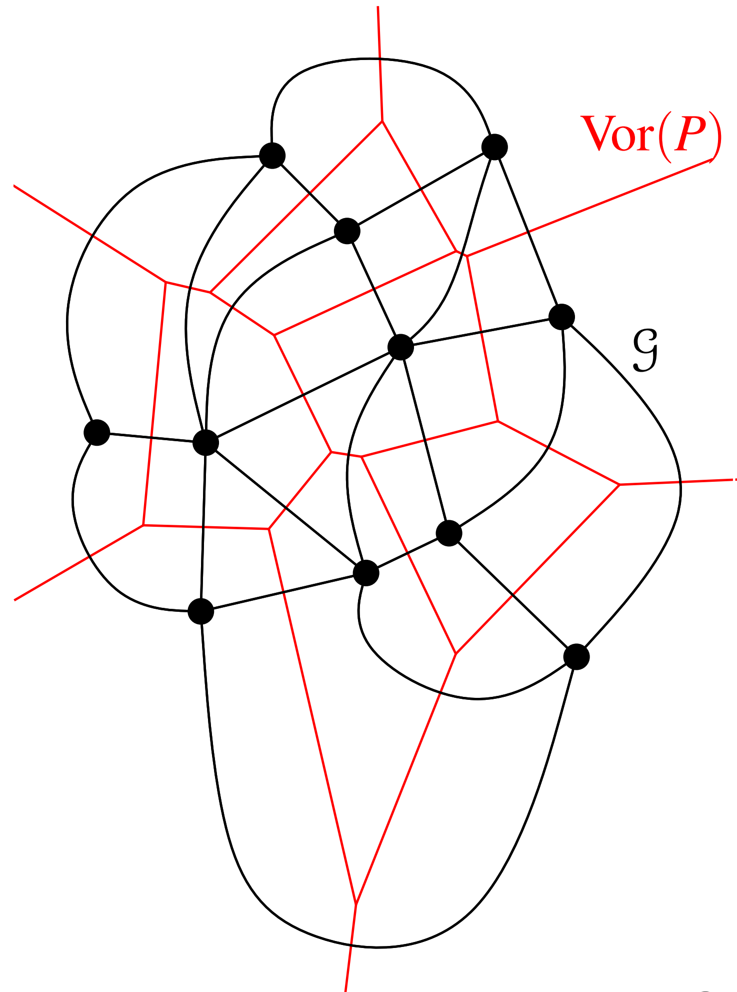
Voronoi Diagram and Delaunay Graph

- Let P be a set of n points in the plane.
- The Voronoi diagram $\text{Vor}(P)$ is the subdivision of the plane into Voronoi cells $\mathcal{V}(p)$ for all $p \in P$.
- Let $\mathcal{G}$ be the *dual graph* of $\text{Vor}(P)$.
- The Delaunay graph $\mathcal{DG}(P)$ is the *straight line embedding* of $\mathcal{G}$.
- **Question:** How can we compute the Delaunay graph?



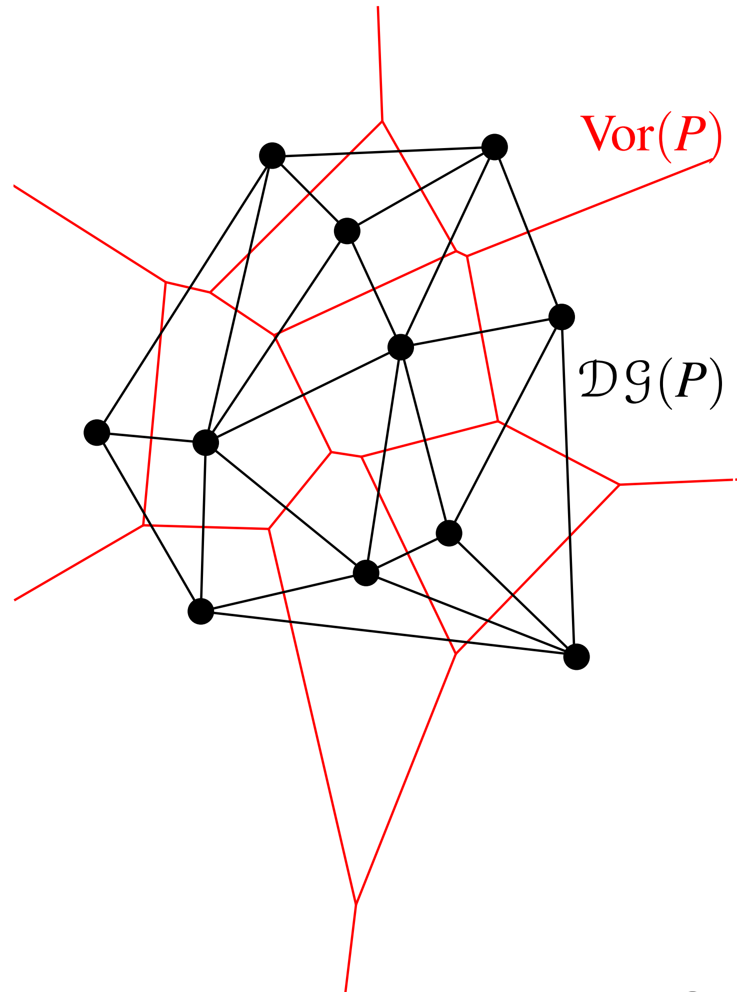
Voronoi Diagram and Delaunay Graph

- Let P be a set of n points in the plane.
- The Voronoi diagram $\mathrm{Vor}(P)$ is the subdivision of the plane into Voronoi cells $\mathcal{V}(p)$ for all $p \in P$.
- Let $\mathcal{G}$ be the *dual graph* of $\mathrm{Vor}(P)$.
- The Delaunay graph $\mathcal{DG}(P)$ is the *straight line embedding* of $\mathcal{G}$.
- **Question:** How can we compute the Delaunay graph?



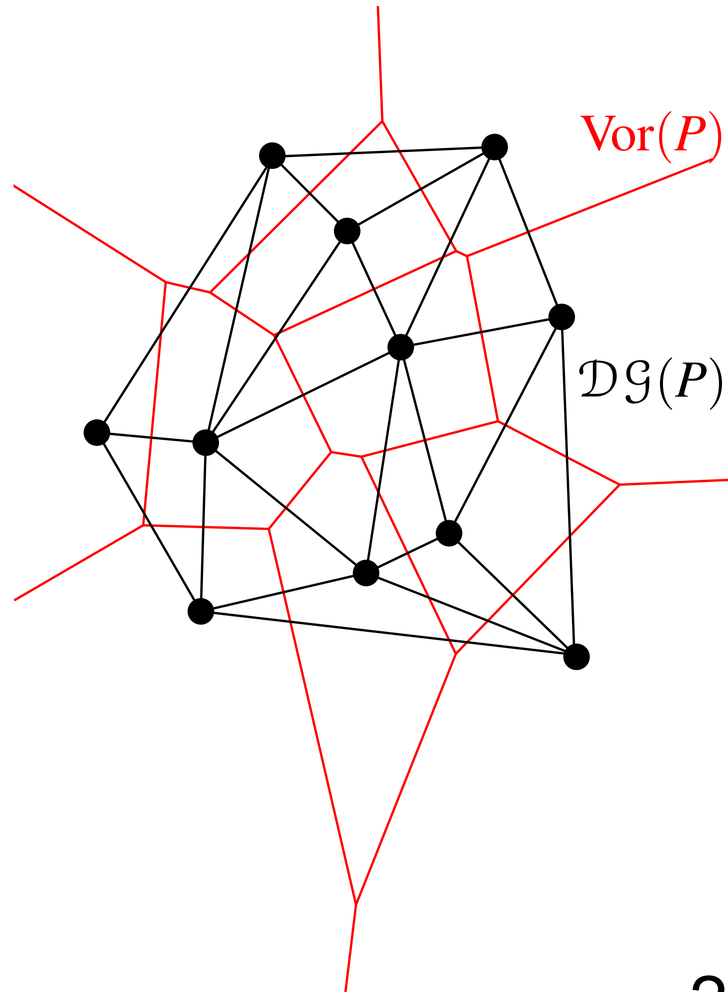
Voronoi Diagram and Delaunay Graph

- Let P be a set of n points in the plane.
- The Voronoi diagram $\text{Vor}(P)$ is the subdivision of the plane into Voronoi cells $\mathcal{V}(p)$ for all $p \in P$.
- Let $\mathcal{G}$ be the *dual graph* of $\text{Vor}(P)$.
- The Delaunay graph $\mathcal{DG}(P)$ is the *straight line embedding* of $\mathcal{G}$.
- **Question:** How can we compute the Delaunay graph?



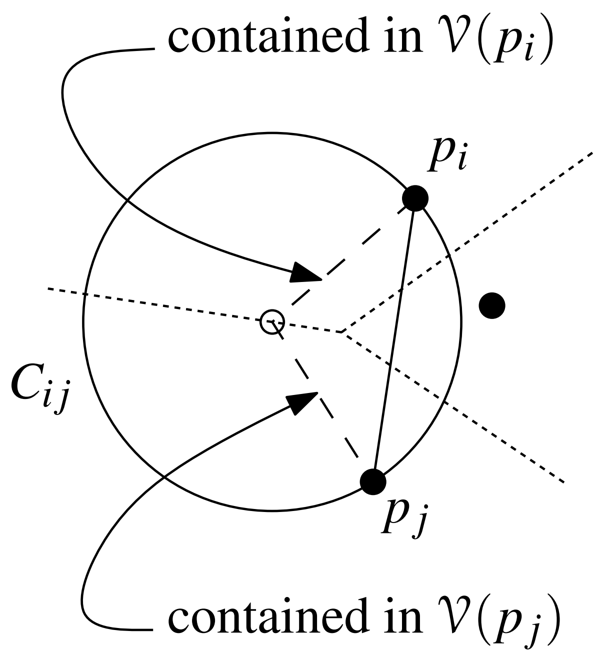
Voronoi Diagram and Delaunay Graph

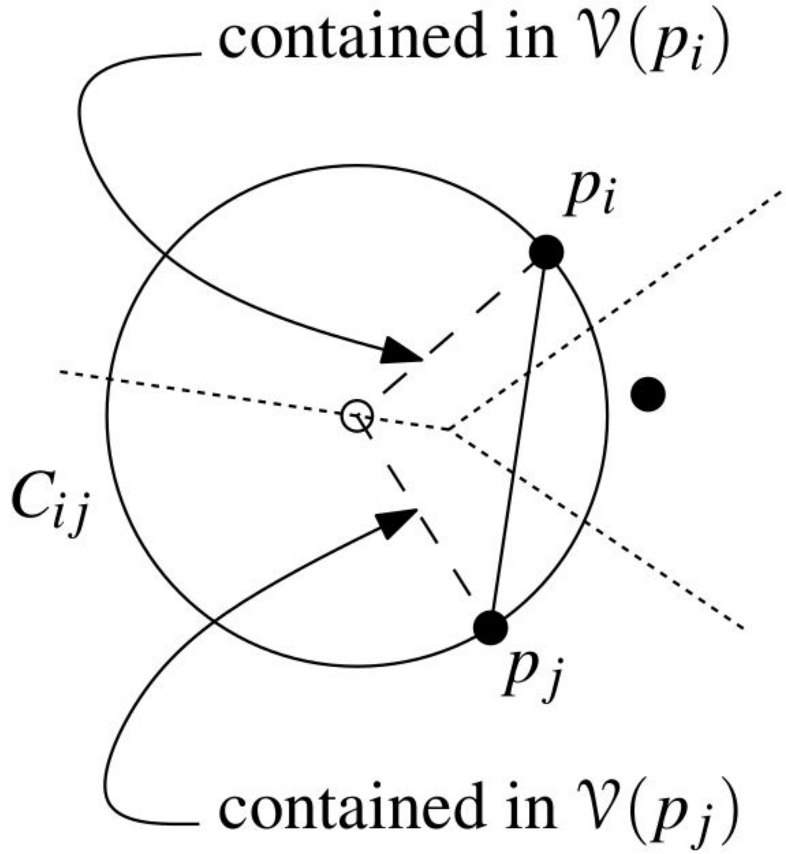
- Let P be a set of n points in the plane.
- The Voronoi diagram $\text{Vor}(P)$ is the subdivision of the plane into Voronoi cells $\mathcal{V}(p)$ for all $p \in P$.
- Let $\mathcal{G}$ be the *dual graph* of $\text{Vor}(P)$.
- The Delaunay graph $\mathcal{DG}(P)$ is the *straight line embedding* of $\mathcal{G}$.
- **Question:** How can we compute the Delaunay graph?



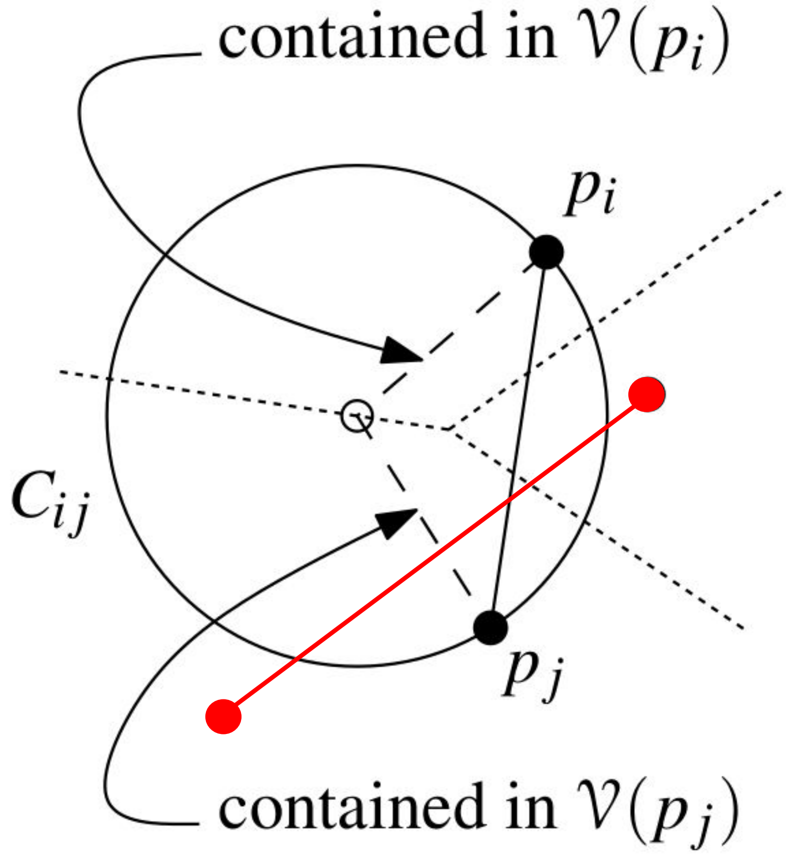
Planarity of the Delaunay Graph

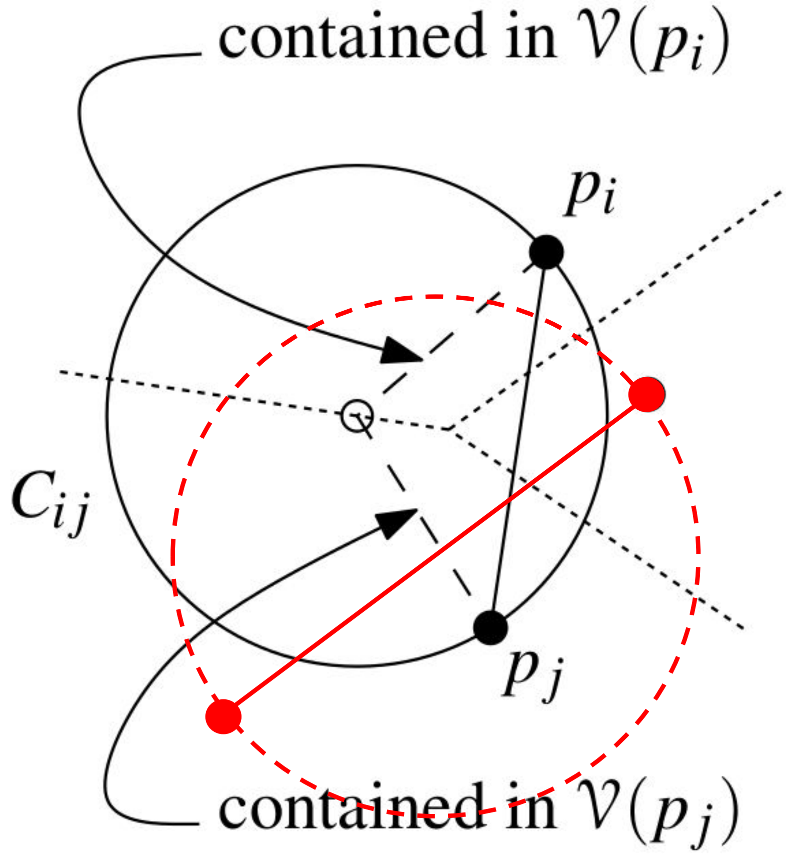
Theorem: The Delaunay graph of a planar point set is a plane graph.

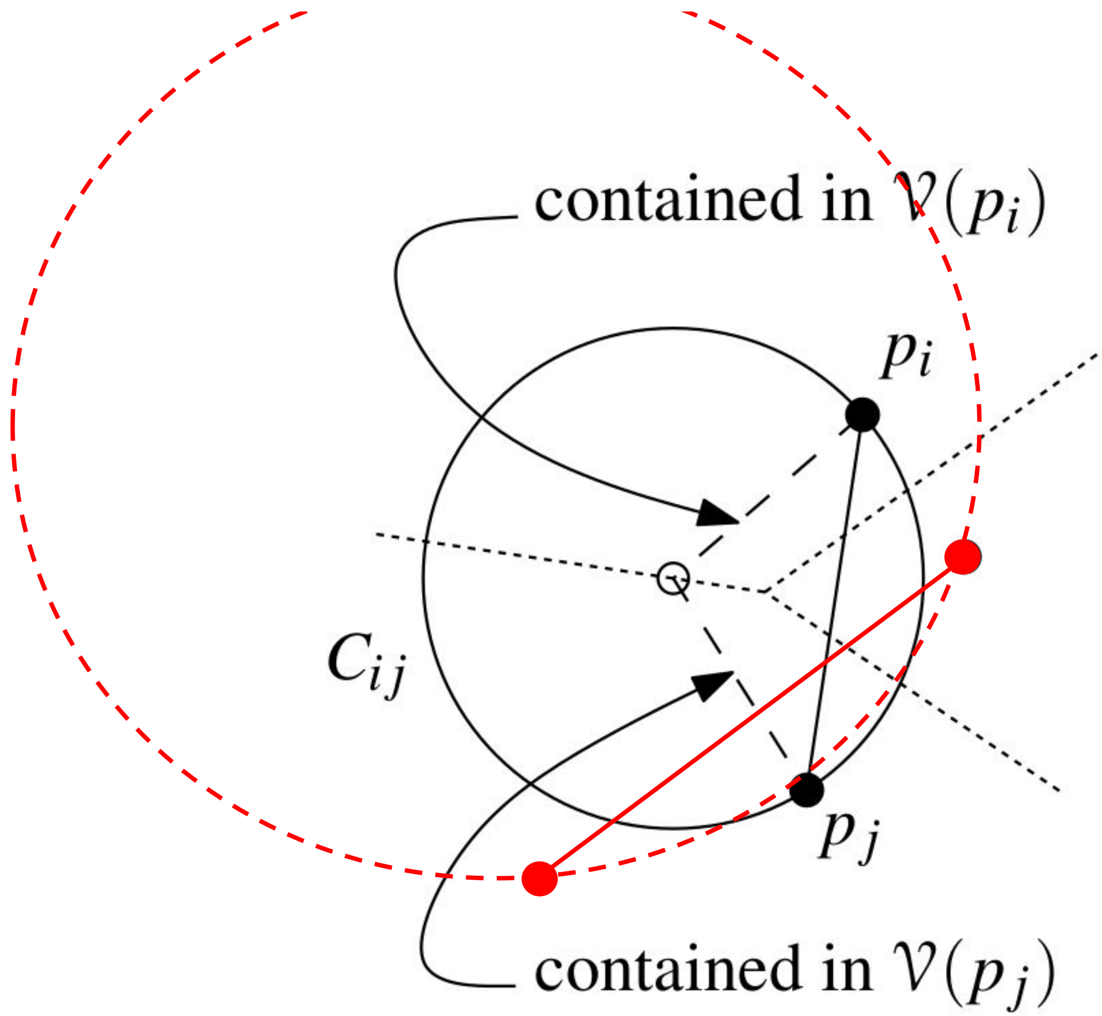




contained in $\mathcal{V}(p_i)$
p_i
C_{ij}
p_j
contained in $\mathcal{V}(p_j)$


contained in $\mathcal{V}(p_i)$
p_i
C_{ij}
p_j
contained in $\mathcal{V}(p_j)$

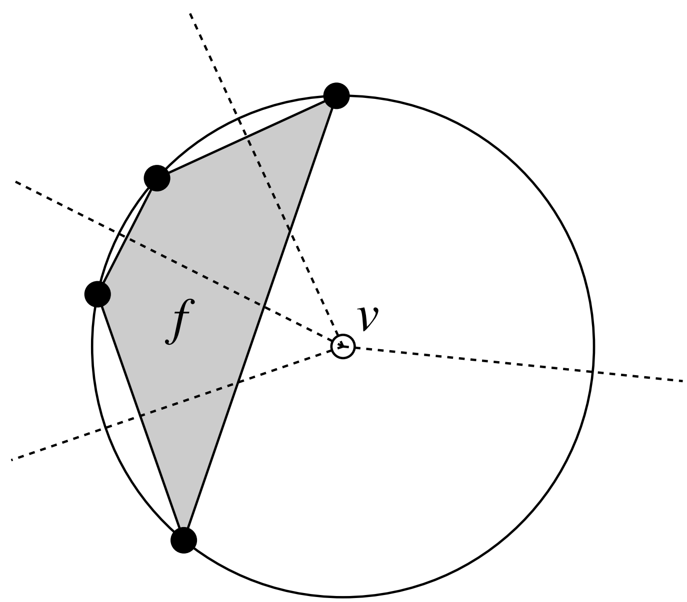

contained in $\mathcal{V}(p_i)$
p_i
C_{ij}
p_j
contained in $\mathcal{V}(p_j)$



contained in $\mathcal{V}(p_i)$
p_i
C_{ij}
p_j
contained in $\mathcal{V}(p_j)$

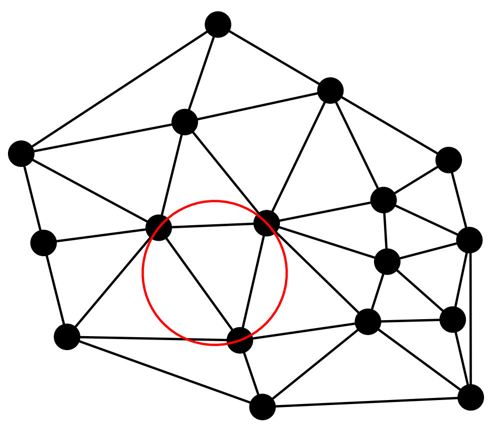
Delaunay Triangulation

If the point set P is in *general position* then the Delaunay graph is a triangulation.



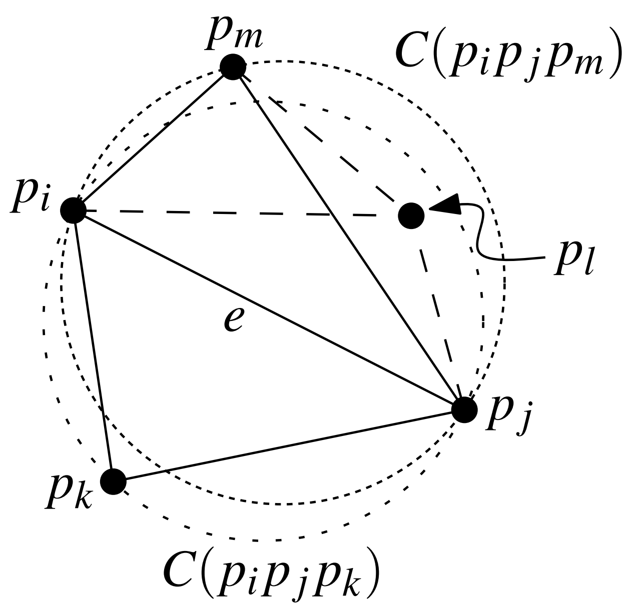
Empty Circle Property

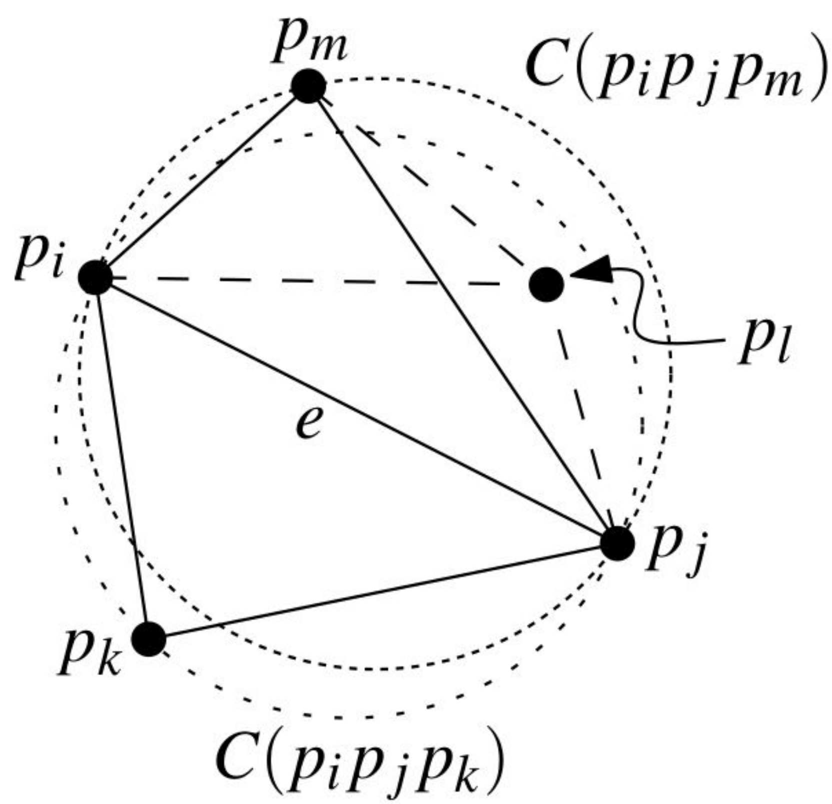
Theorem: Let P be a set of points in the plane, and let $\mathcal{T}$ be a triangulation of P. Then $\mathcal{T}$ is a Delaunay triangulation of P if and only if the circumcircle of any triangle of $\mathcal{T}$ does not contain a point of P in its interior.



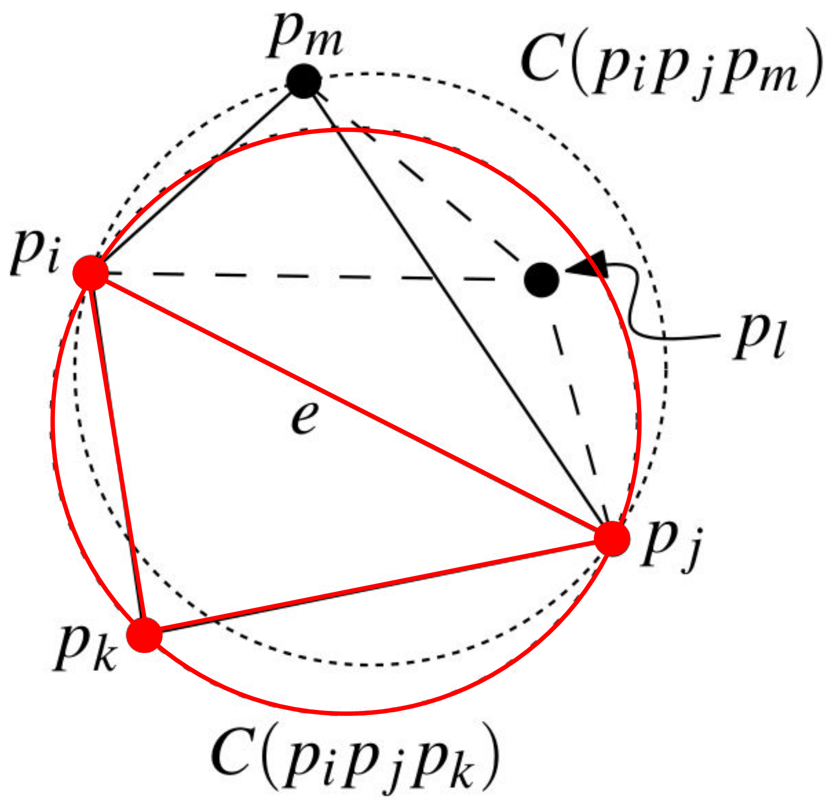
Delaunay Triangulations and Legal Triangulations

Theorem: Let P be a set of points in the plane. A triangulation $\mathcal{T}$ of P is legal if and only if $\mathcal{T}$ is a Delaunay triangulation.

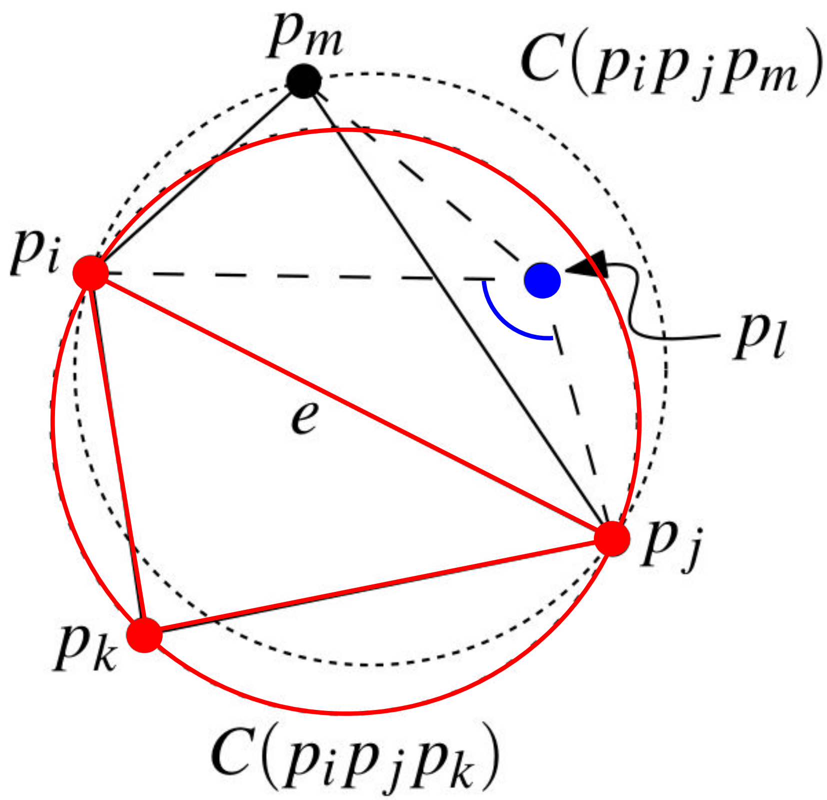


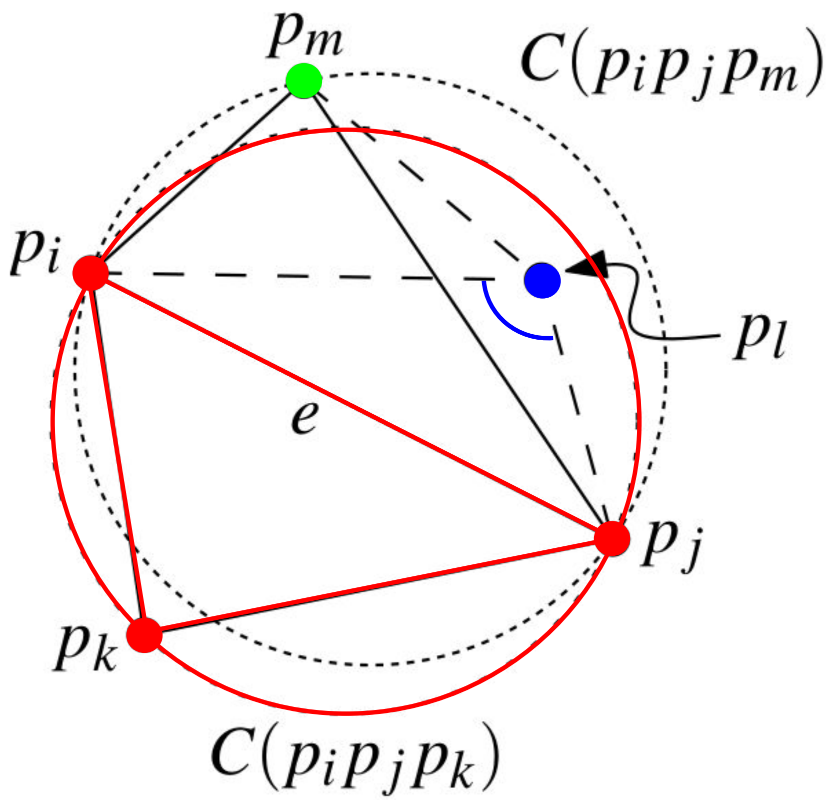


p_m
$C(p_i p_j p_m)$
p_i
p_l
e
p_j
p_k
$C(p_i p_j p_k)$

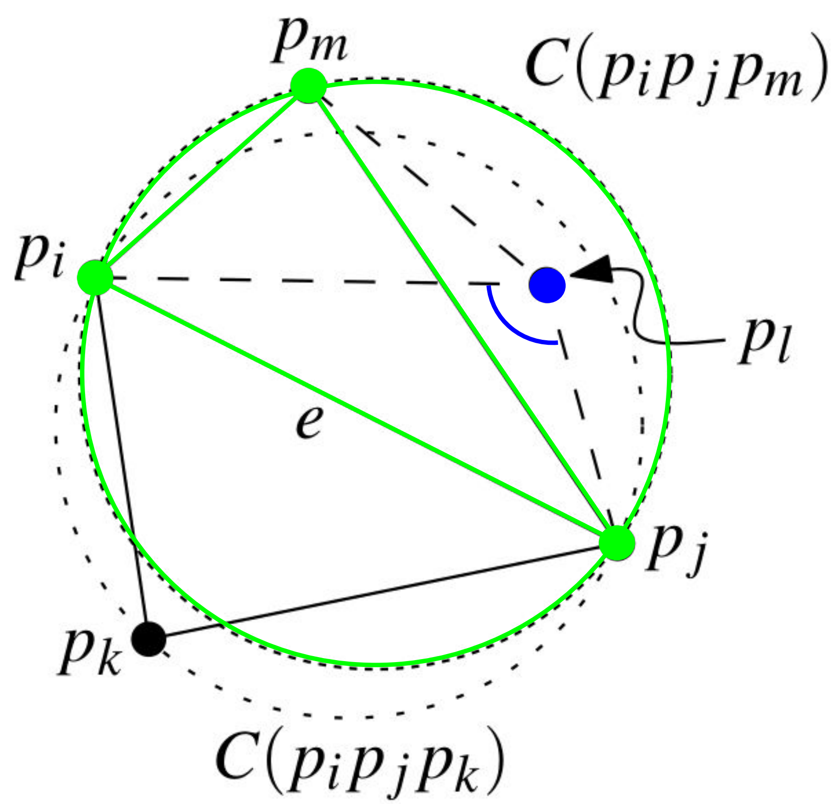


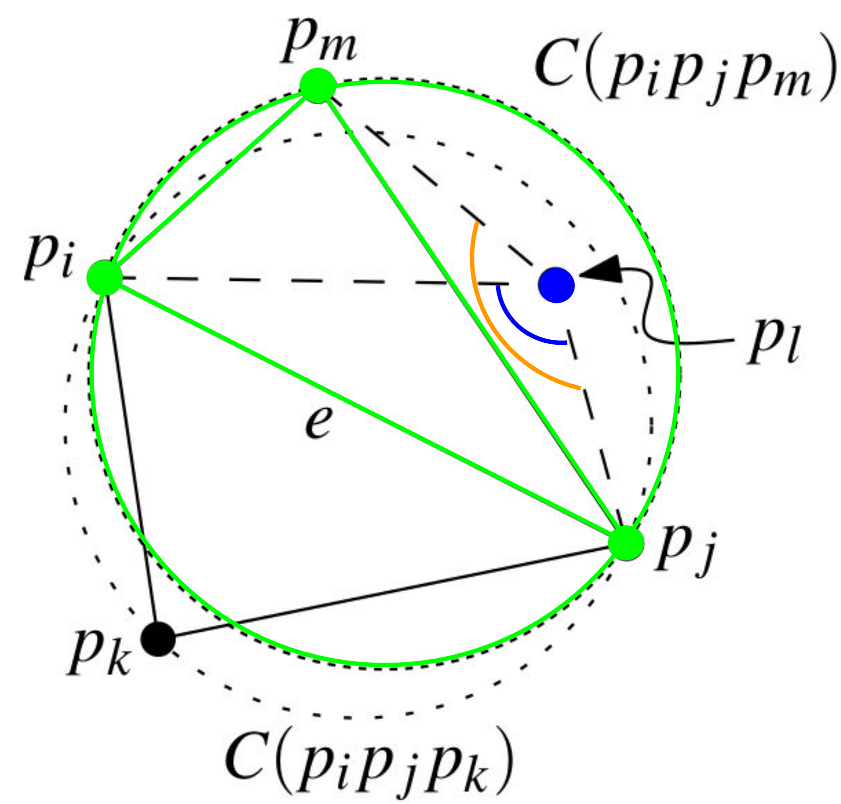
p_m
$C(p_i p_j p_m)$
p_i
p_l
e
p_j
p_k
$C(p_i p_j p_k)$


p_m
$C(p_i p_j p_m)$
p_i
p_l
e
p_j
p_k
$C(p_i p_j p_k)$



p_m
$C(p_i p_j p_m)$
p_i
p_l
e
p_j
p_k
$C(p_i p_j p_k)$


p_m
$C(p_i p_j p_m)$
p_i
p_l
e
p_j
p_k
$C(p_i p_j p_k)$



p_m
$C(p_i p_j p_m)$
p_i
p_l
e
p_j
p_k
$C(p_i p_j p_k)$

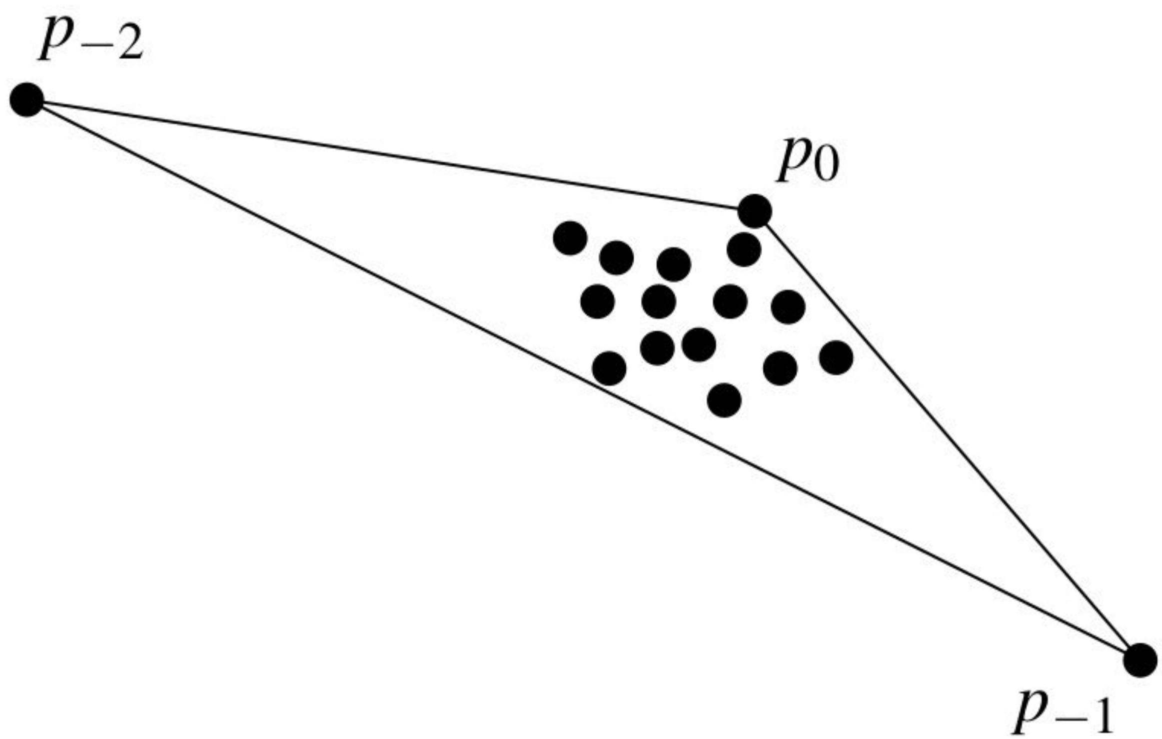
Angle Optimality and Delaunay Triangulations

Theorem: Let P be a set of points in the plane. Any angle-optimal triangulation of P is a Delaunay triangulation of P. Furthermore, any Delaunay triangulation of P maximizes the minimum angle over all triangulations of P.

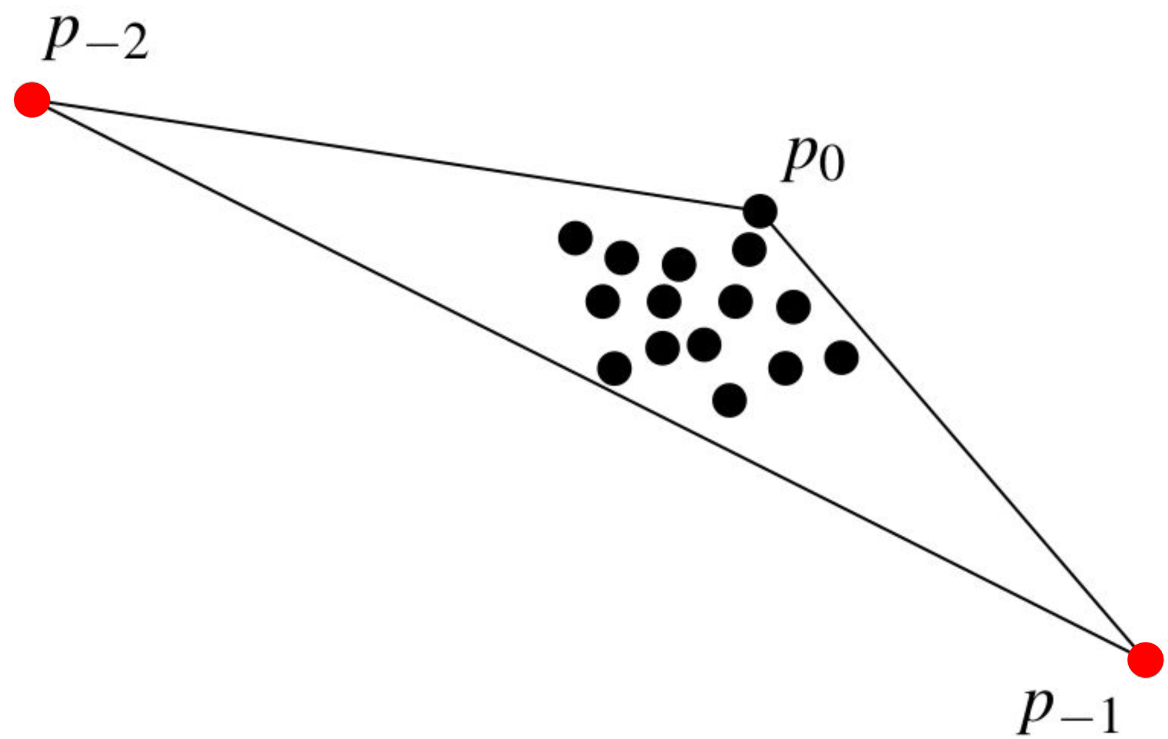
Randomized Incremental Construction

Algorithm DELAUNAYTRIANGULATION(P)
Input. A set P of $n+1$ points in the plane.
Output. A Delaunay triangulation of P.

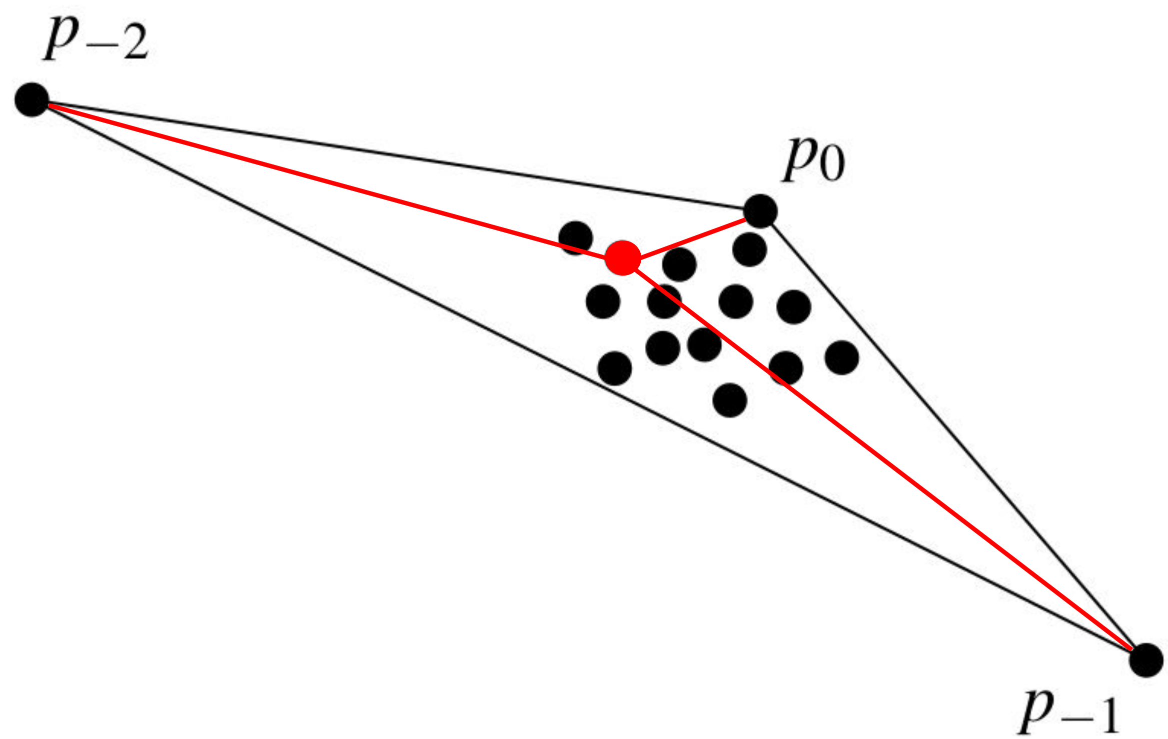
1. Initialize $\mathcal{T}$ as the triangulation consisting of an outer triangle $p_0 p_{-1} p_{-2}$ containing points of P, where p_0 is the lexicographically highest point of P.
2. Compute a random permutation $p_1, p_2, \ldots, p_n$ of $P \setminus \{p_0\}$.
3. **for** $r \leftarrow 1$ **to** n
4. **do**
5. LOCATE($p_r, \mathcal{T}$)
6. INSERT($p_r, \mathcal{T}$)
7. Discard p_{-1} and p_{-2} with all their incident edges from $\mathcal{T}$.
8. **return** $\mathcal{T}$



p_{-2}
p_0
p_{-1}



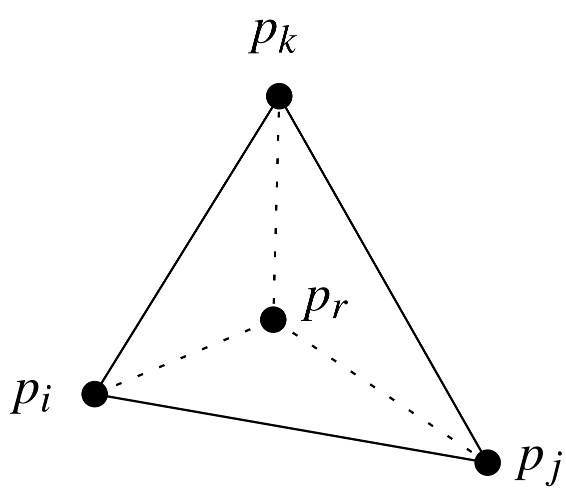
p_{-2}
p_0
p_{-1}



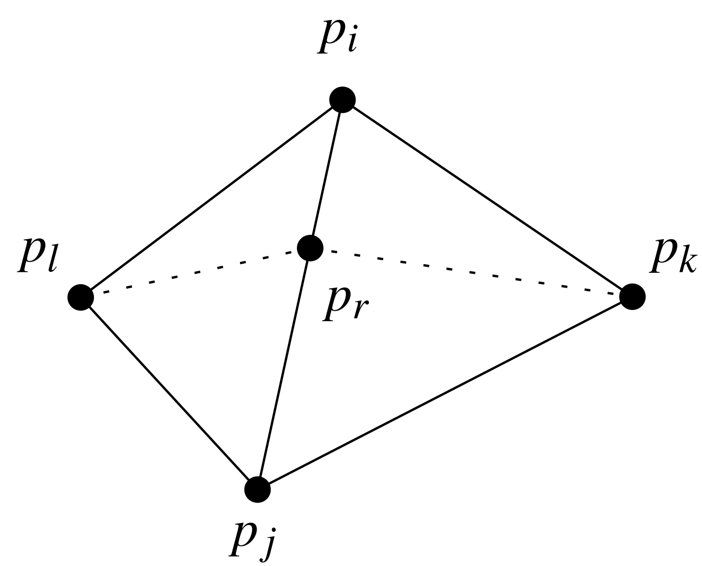
p_{-2}
p_0
p_{-1}

Randomized Incremental Construction

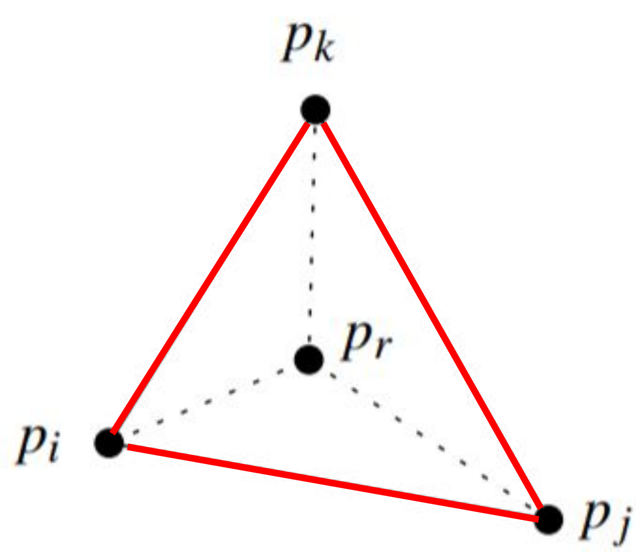
p_r lies in the interior of a triangle



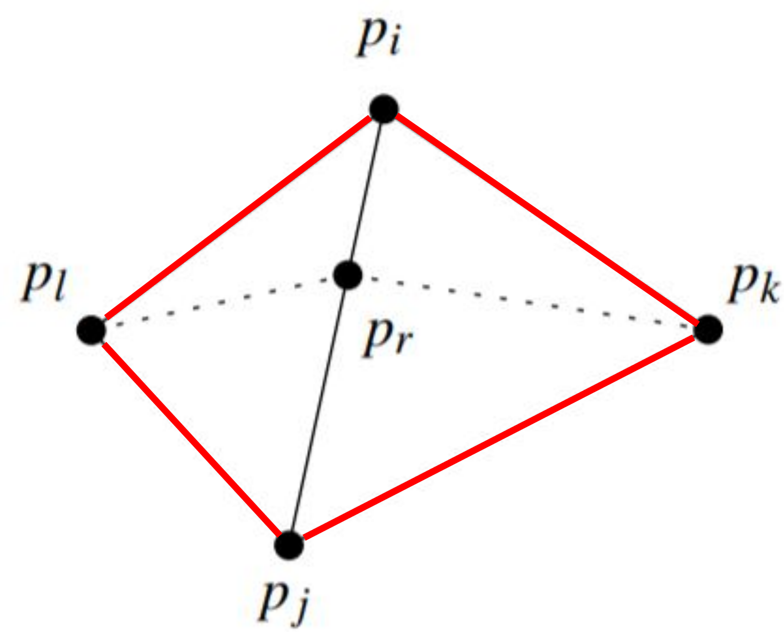
p_r falls on an edge



p_r lies in the interior of a triangle



p_r falls on an edge



Randomized Incremental Construction

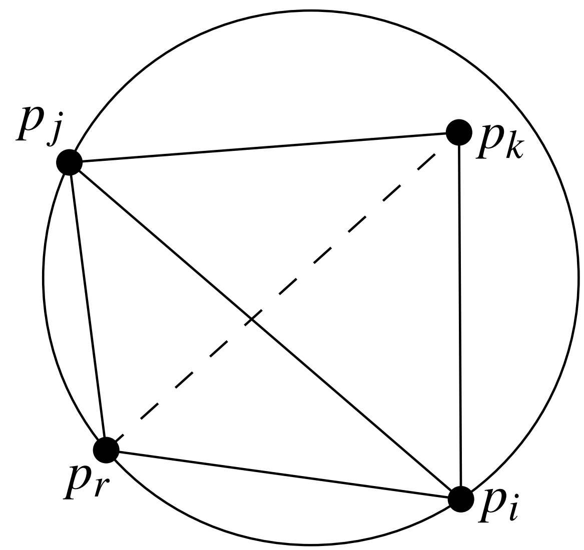
INSERT($p_r, \mathcal{T}$)

1. **if** p_r lies in the interior of the triangle $p_ip_jp_k$
2. **then** Add edges from p_r to the three vertices of $p_ip_jp_k$, thereby splitting $p_ip_jp_k$ into three triangles.
3. LEGALIZEEDGE($p_r, \overline{p_ip_j}, \mathcal{T}$)
4. LEGALIZEEDGE($p_r, \overline{p_jp_k}, \mathcal{T}$)
5. LEGALIZEEDGE($p_r, \overline{p_kp_i}, \mathcal{T}$)
6. **else** ($*$ p_r lies on an edge of $p_ip_jp_k$, say the edge $\overline{p_ip_j}$ $*$)
7. Add edges from p_r to p_k and to the third vertex p_l of the other triangle that is incident to $\overline{p_ip_j}$, thereby splitting the two triangles incident to $\overline{p_ip_j}$ into four triangles.
8. LEGALIZEEDGE($p_r, \overline{p_ip_l}, \mathcal{T}$)
9. LEGALIZEEDGE($p_r, \overline{p_lp_j}, \mathcal{T}$)
10. LEGALIZEEDGE($p_r, \overline{p_jp_k}, \mathcal{T}$)
11. LEGALIZEEDGE($p_r, \overline{p_kp_i}, \mathcal{T}$)

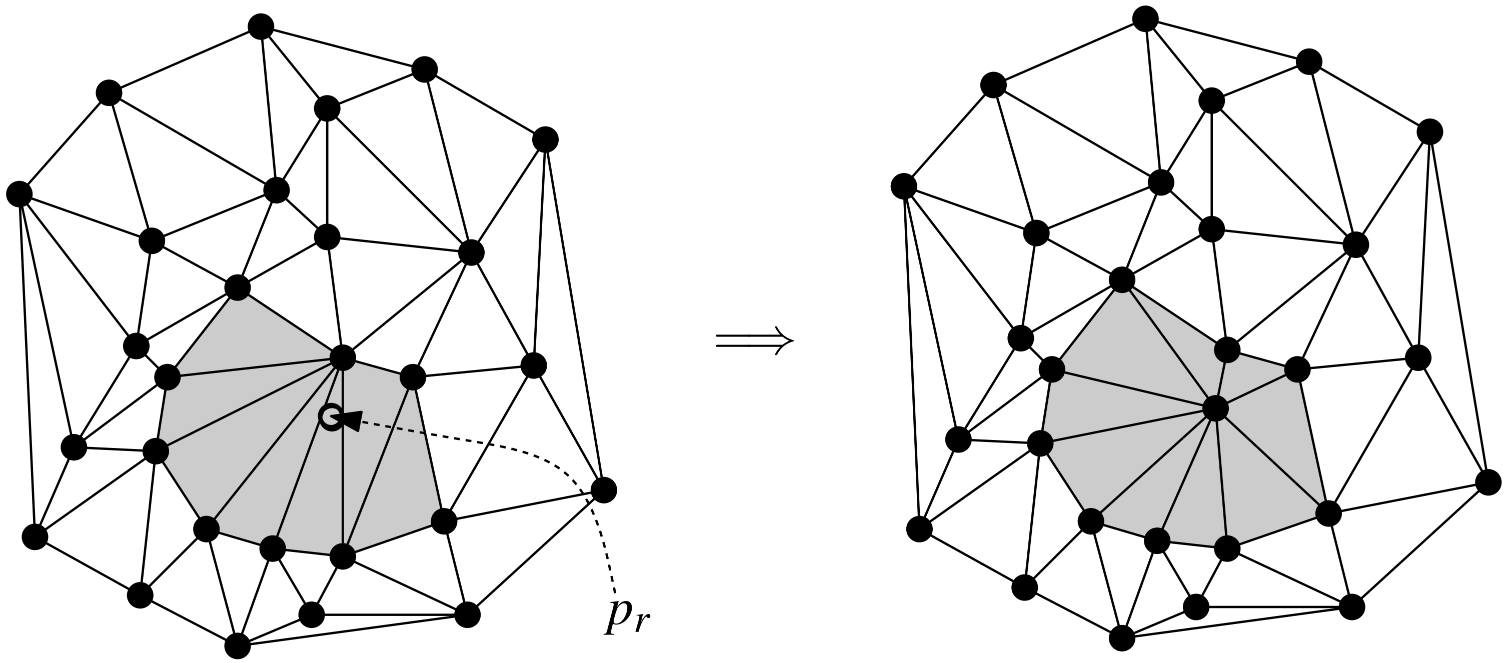
Randomized Incremental Construction

$\textsc{LegalizeEdge}(p_r, \overline{p_i p_j}, \mathcal{T})$

1. (∗ The point being inserted is p_r, and $\overline{p_i p_j}$ is the edge of $\mathcal{T}$ that may need to be flipped. ∗)
2. **if** $\overline{p_i p_j}$ is illegal
3. **then** Let $p_i p_j p_k$ be the triangle adjacent to $p_r p_i p_j$ along $\overline{p_i p_j}$.
4. (∗ Flip $\overline{p_i p_j}$: ∗) Replace $\overline{p_i p_j}$ with $\overline{p_r p_k}$.
5. $\textsc{LegalizeEdge}(p_r, \overline{p_i p_k}, \mathcal{T})$
6. $\textsc{LegalizeEdge}(p_r, \overline{p_k p_j}, \mathcal{T})$

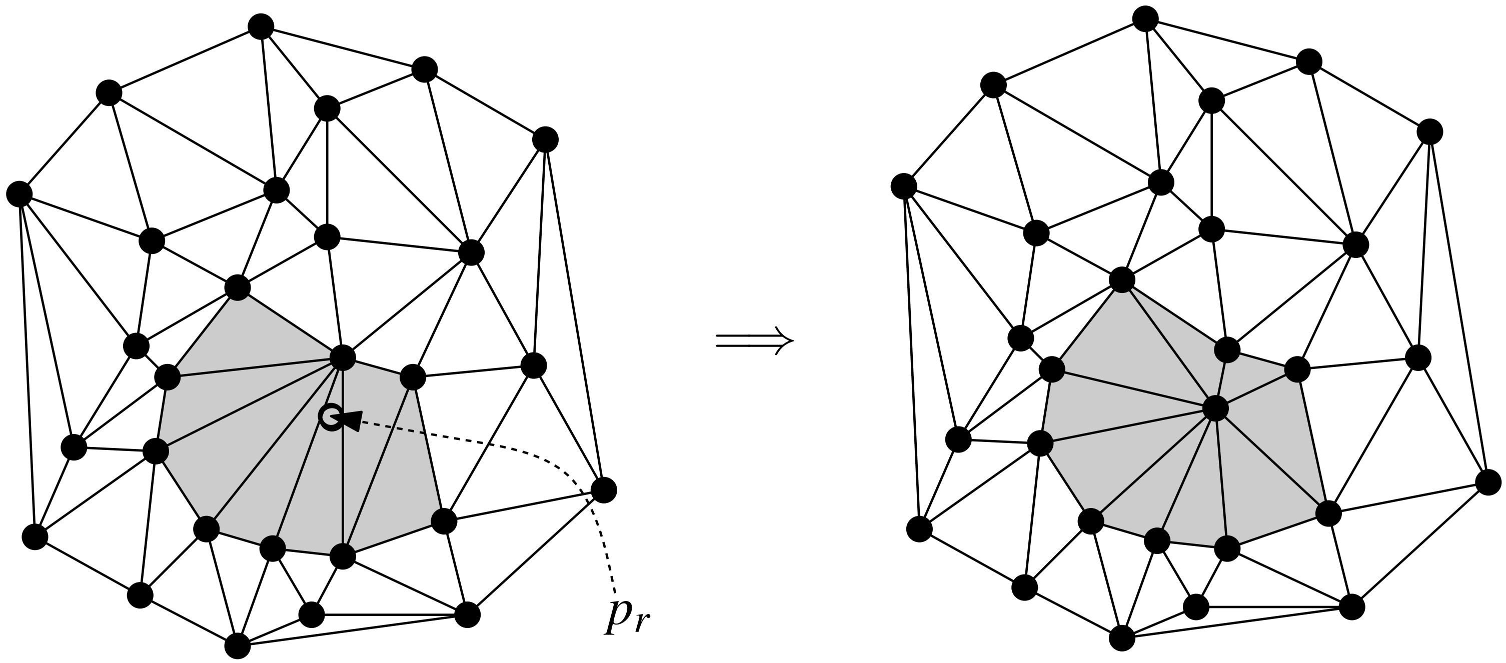


Randomized Incremental Construction



All edges created are incident to p_r.

Randomized Incremental Construction



All edges created are incident to p_r.

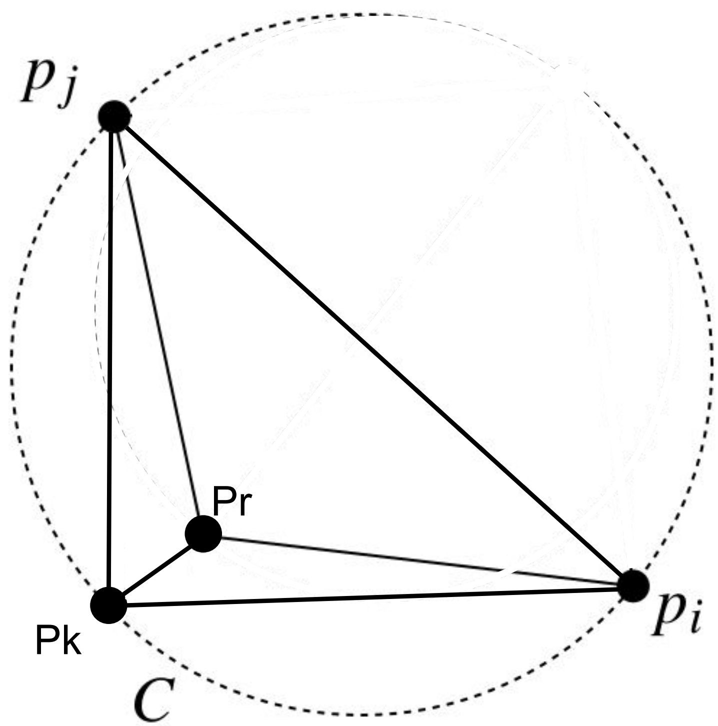
Correctness: Are new edges legal?

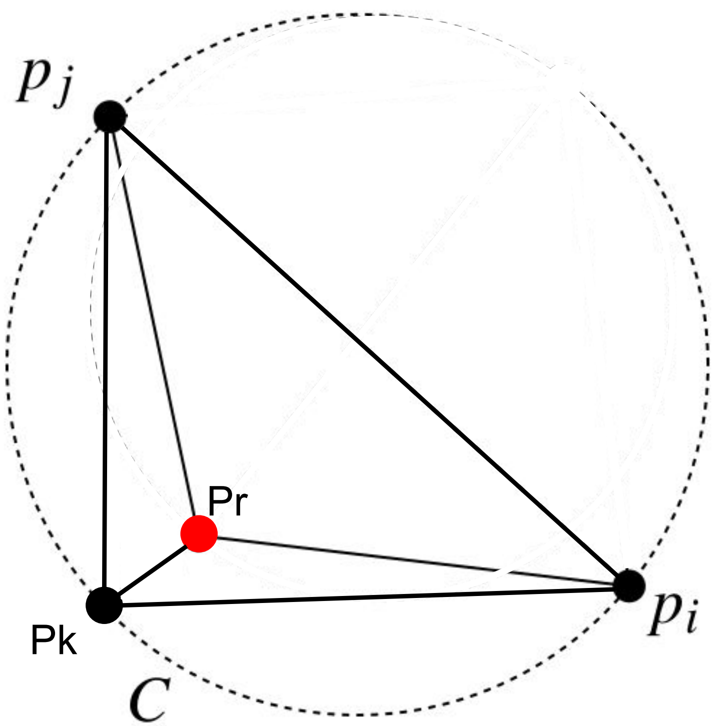
Correctness

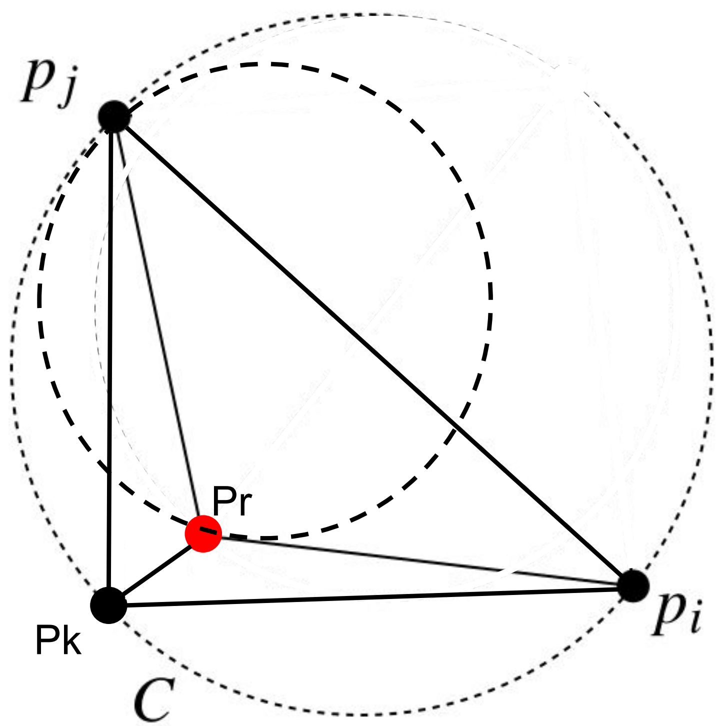
- Newly added edges
- Edges due to flipping

Correctness

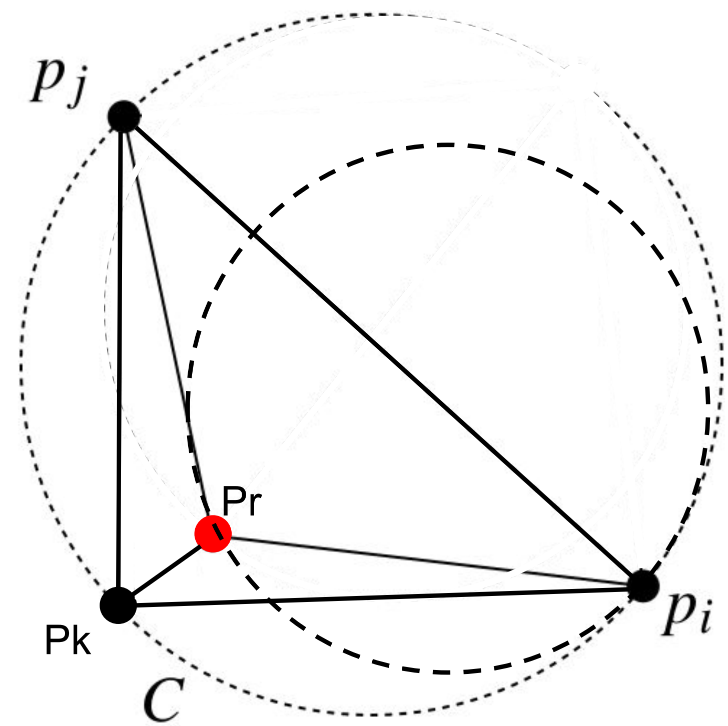
- **Newly added edges**
- Edges due to flipping


p_j
Pr
Pk
p_i
C

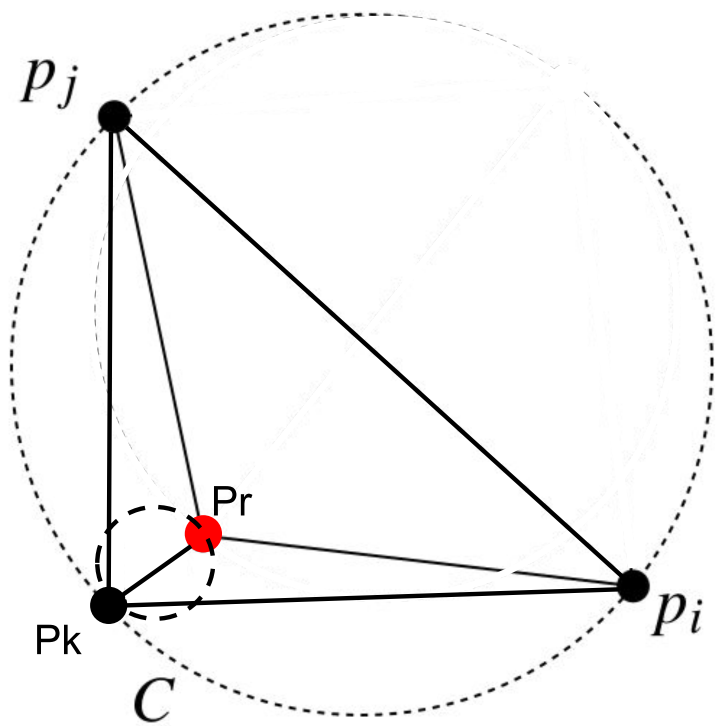

p_j
Pr
Pk
p_i
C



p_j
Pr
Pk
p_i
C

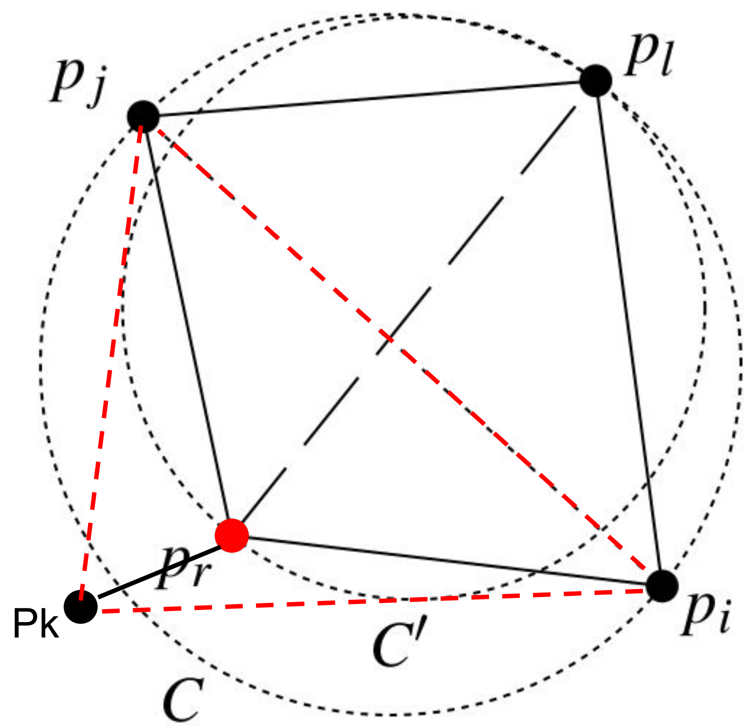


p_j
Pr
Pk
p_i
C

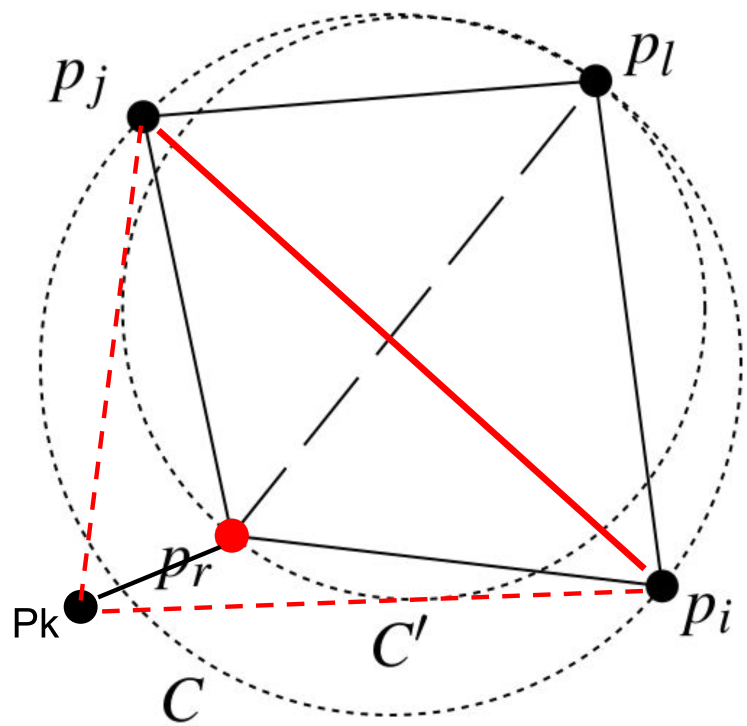

p_j
Pr
Pk
p_i
C

Correctness

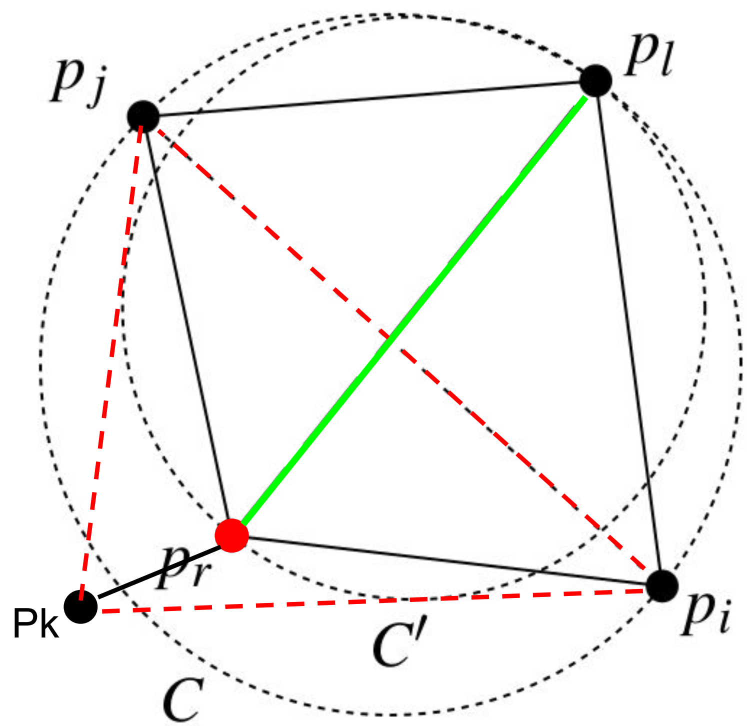
- Newly added edges
- **Edges due to flipping**



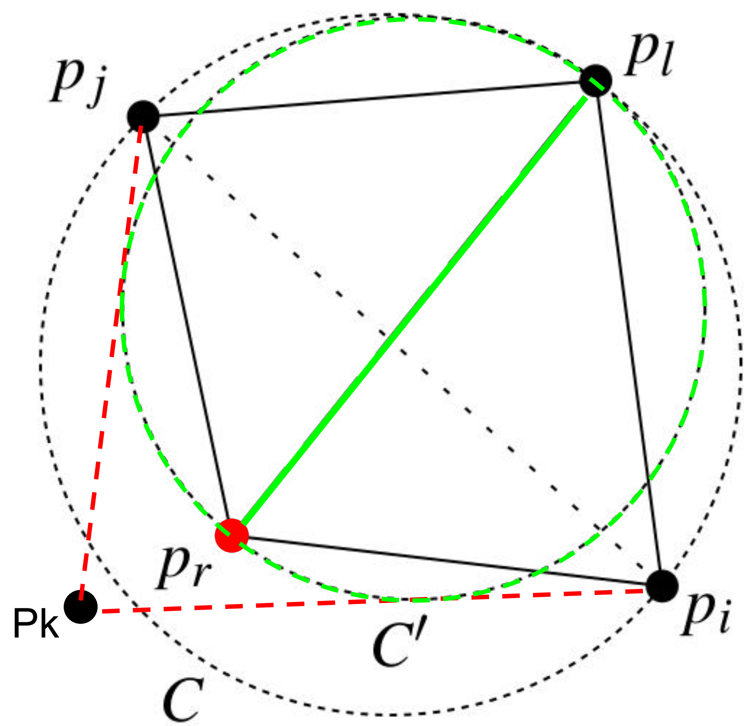
p_j
p_l
p_r
Pk
p_i
C'
C



p_j
p_l
p_r
Pk
p_i
C'
C



p_j
p_l
p_r
Pk
p_i
C'
C



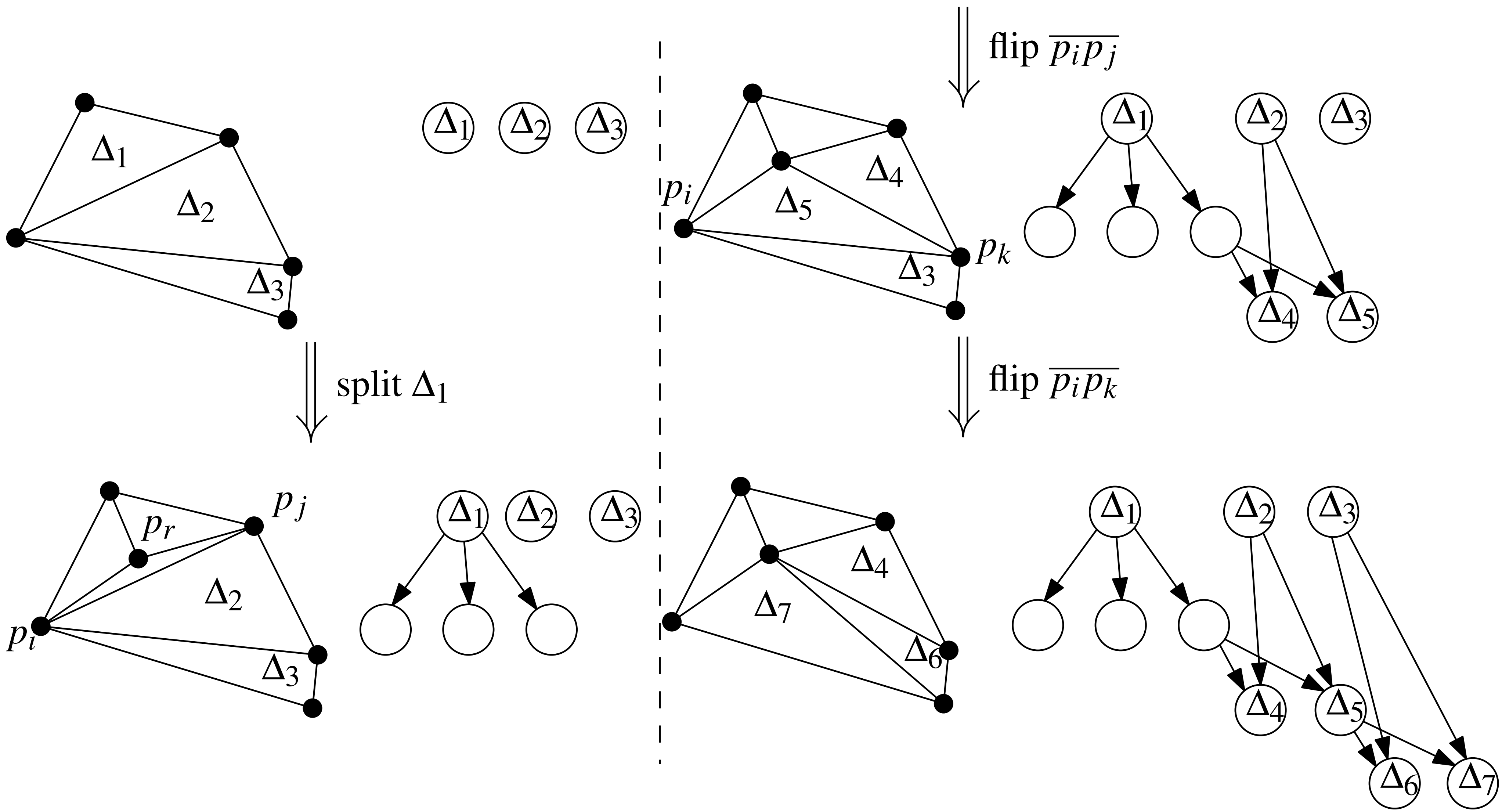
p_j
p_l
p_r
p_i
Pk
C'
C

Randomized Incremental Construction

Initializing triangulation: treat p_{-1} and p_{-2} symbolically.
No actual coordinates.
Modify tests for point location and illegal edges to work as if far away.

Point location: search data structure.
Point visits triangles of previous triangulations that contain it.

Randomized Incremental Construction



Analysis

- Expected total number of triangles created is O(n)
  - Space usage (point data structure) O(n)
  - Expected time other than point location queries O(n)
- Expected total number of triangles visited while search point location data structure is O(n log n)

Analysis

Lemma: Total number of triangles created is at most 9n + 1

- For each of p_r added, let it have k incident edges
- It creates at most 2k - 3 triangles
- Known: delaunay graph has at most 3(r+3)-6 edges, three of which is the outer triangle
- 2[3(r+3) - 9] = 6r -- expected degree of a random point is 6
- 2 * 6 - 3 = 9
- +1 for the outer triangle