Flips in Edge-Labelled Pseudo-Triangulations

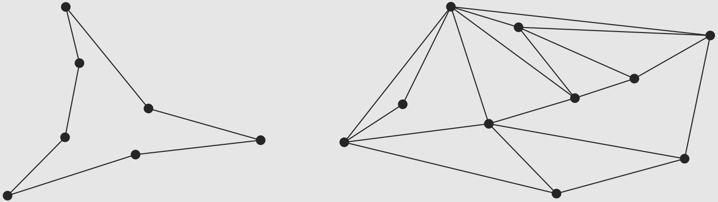
Prosenjit Bose   Sander Verdonschot

Carleton University

10 August 2015

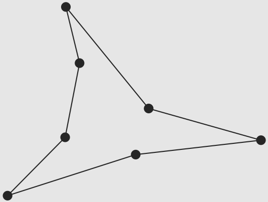
Pseudo-triangulations

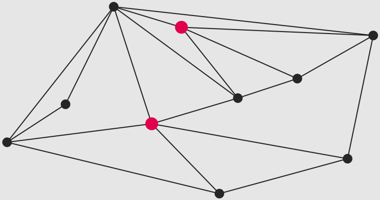
- All faces are pseudo-triangles



Pseudo-triangulations

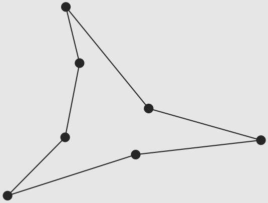
- All faces are pseudo-triangles
- Pointed: all vertices are incident to a reflex angle ($> \pi$)

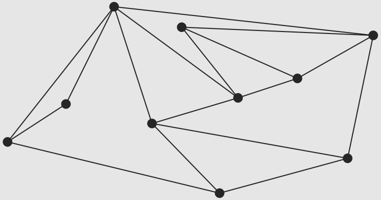




Pseudo-triangulations

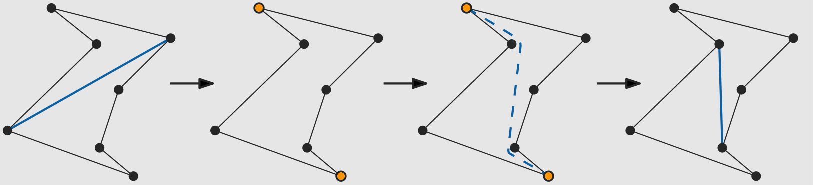
- All faces are pseudo-triangles
- Pointed: all vertices are incident to a reflex angle ($> \pi$)





Flips

- Remove edge, leaving a pseudo-quadrilateral
- Find corners opposite removed edge
- Insert connecting geodesic



Previous work

Theorem (Bereg, 2004)

Any pointed pseudo-triangulation can be transformed into any other with $O(n \log n)$ flips.

Previous work

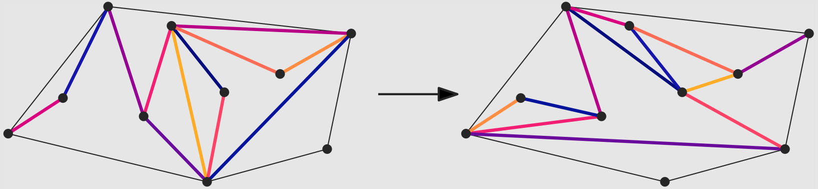
Theorem (Bereg, 2004)

Any pointed pseudo-triangulation can be transformed into any other with $O(n \log n)$ flips.

- What happens when edges are labelled?

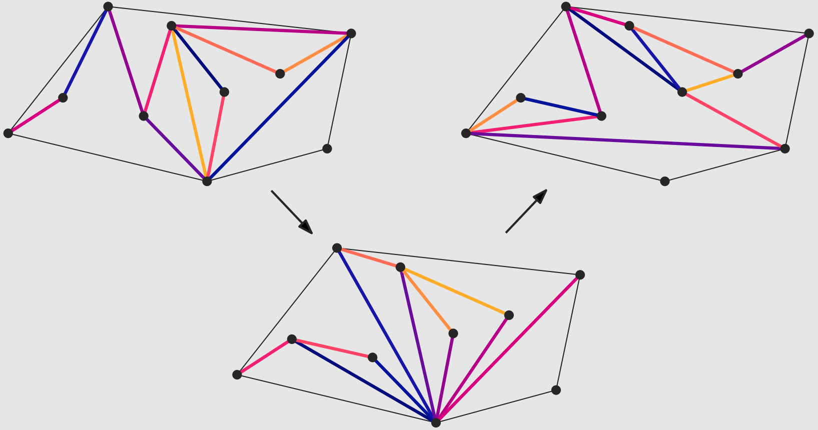
Upper bound

- Transform T_1 into T_2



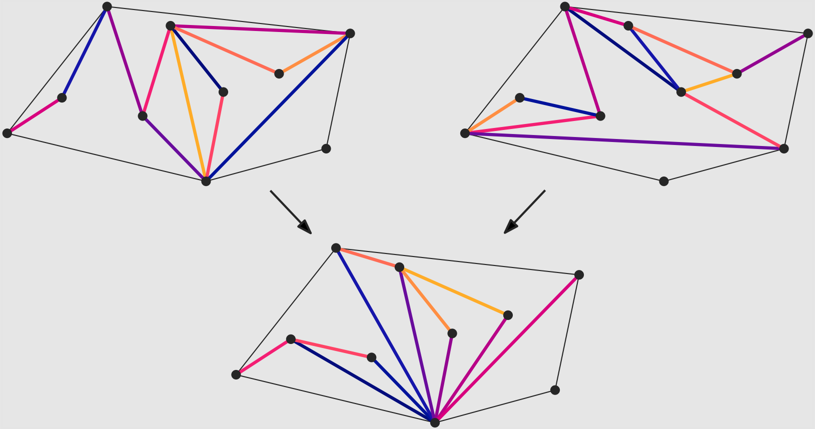
Upper bound

- Transform T_1 into T_2
- Via canonical form T_C



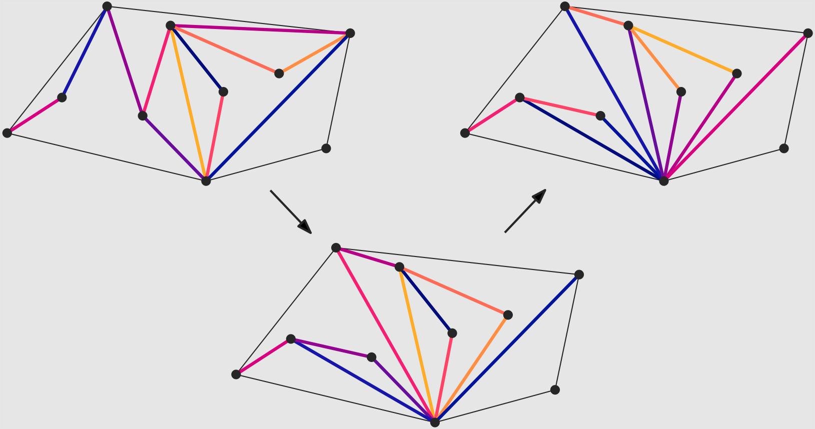
Upper bound

- Transform T_1 into T_2
- Via canonical form T_C
- We only need to show $T \mapsto T_C$



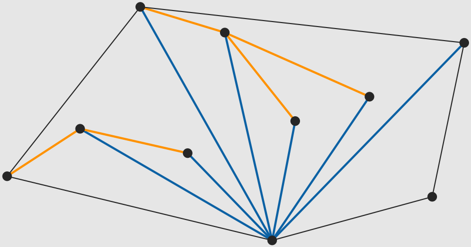
Transform into canonical

- Ignore labels
- Move labels around



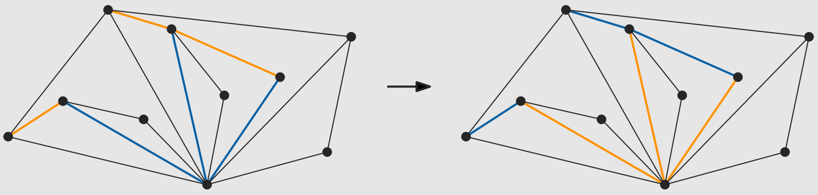
Left-shelling pseudo-triangulation

- Add vertices in clockwise order around bottom vertex
  - Connect to bottom (bottom edge)
  - Add tangent to convex hull (top edge)

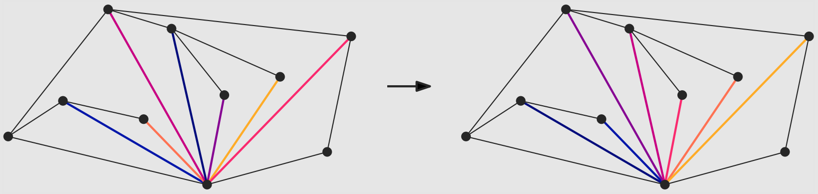


Tools

- *Sweep*: exchange labels on top and bottom pairs

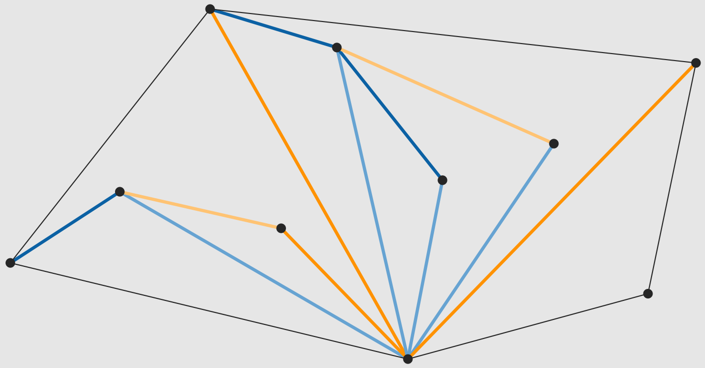


- *Shuffle*: reorder bottom labels



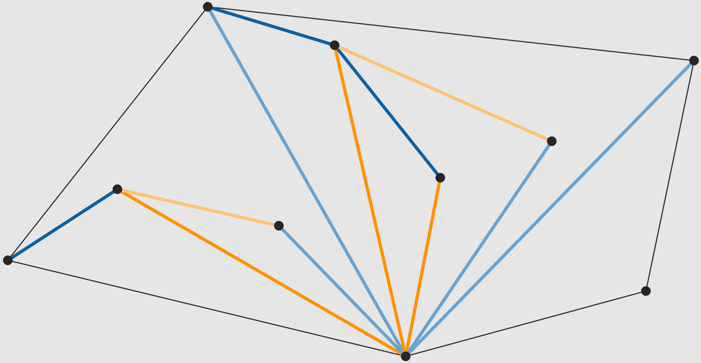
Algorithm

- Identify out-of-place top and bottom labels



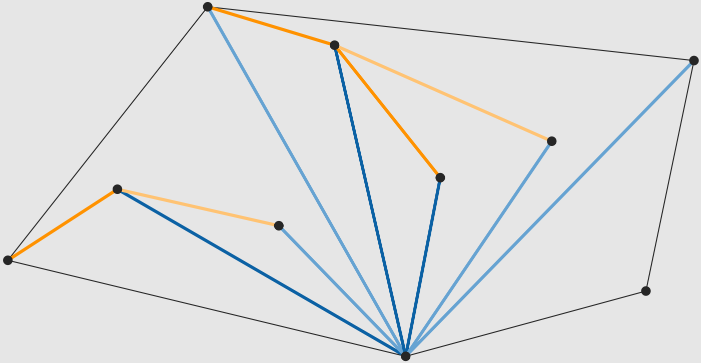
Algorithm

- Pair these up (*Shuffle*)



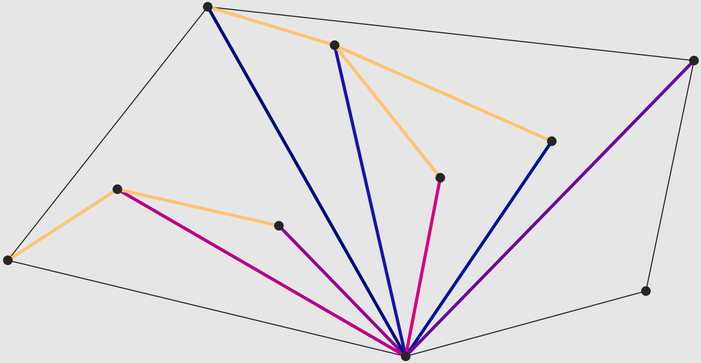
Algorithm

- Exchange them (*Sweep*)



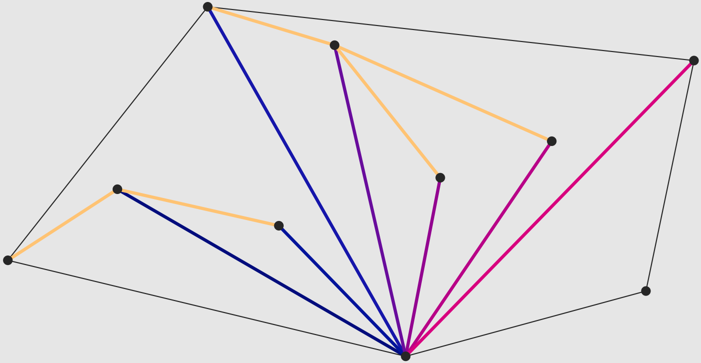
Algorithm

- Sort bottom labels (*Shuffle*)



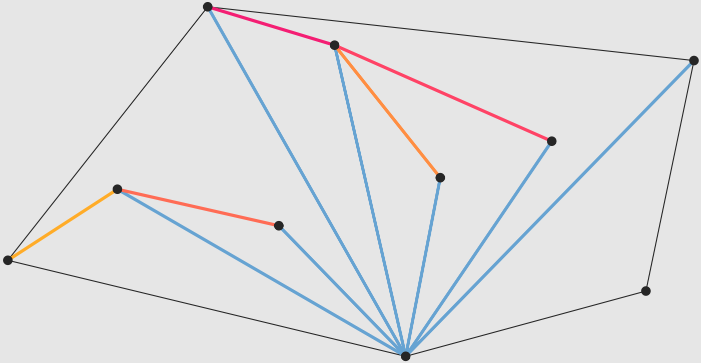
Algorithm

- Sort bottom labels (*Shuffle*)



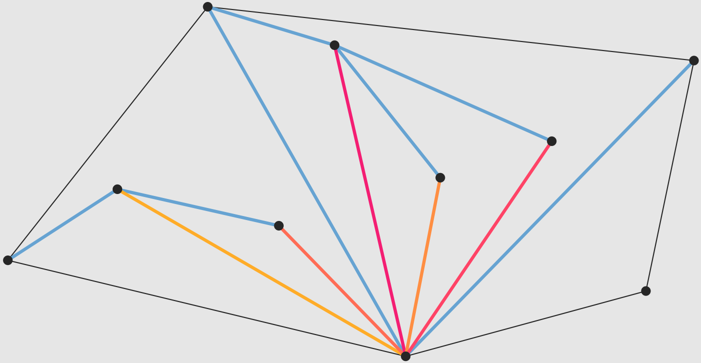
Algorithm

- Move all top labels down (*Sweep*)



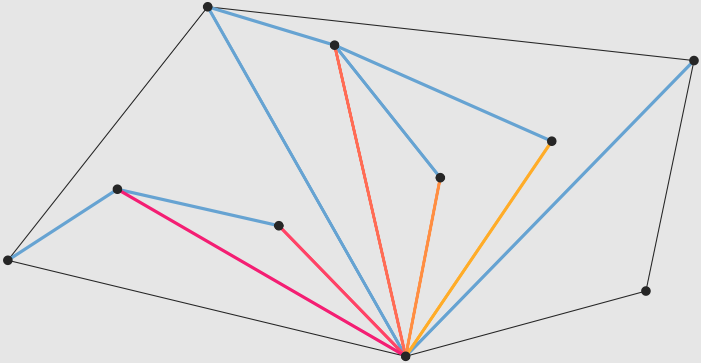
Algorithm

- Move all top labels down (*Sweep*)



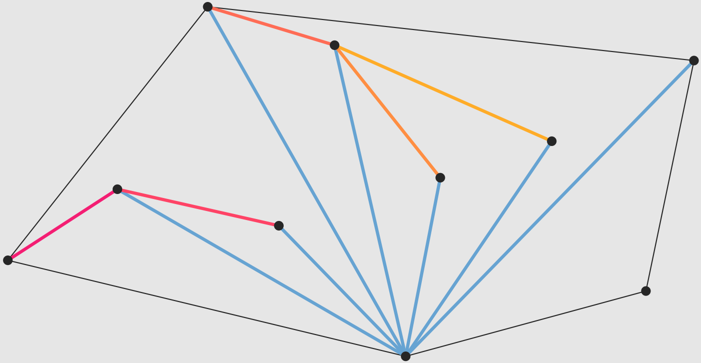
Algorithm

- Sort them (*Shuffle*)



Algorithm

- Move them back (*Sweep*)



Upper bound

Theorem

We can sort the labels of a left-shelling pseudo-triangulation with $O(1)$ shuffles and sweeps.

Upper bound

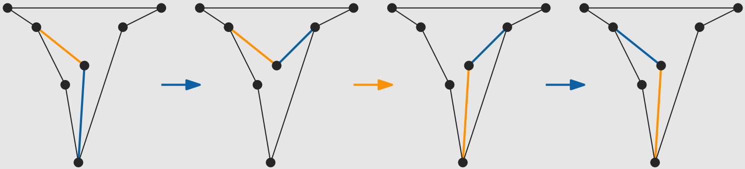
Theorem

We can sort the labels of a left-shelling pseudo-triangulation with $O(1)$ shuffles and sweeps.

- How do we shuffle and sweep?

Sweep

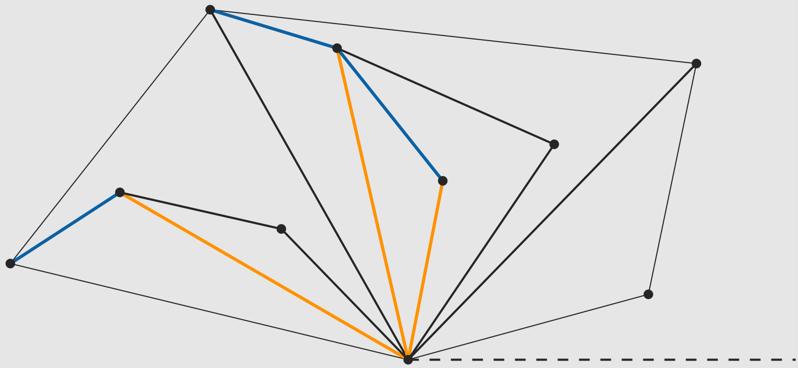
- Easy for degree-2 vertices:



- Idea: make every vertex degree-2 at some point

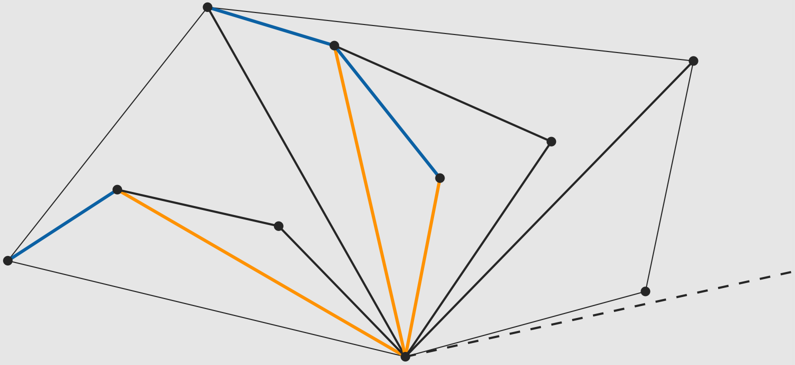
Sweep

- Shoot a ray from v_{bottom} to the right



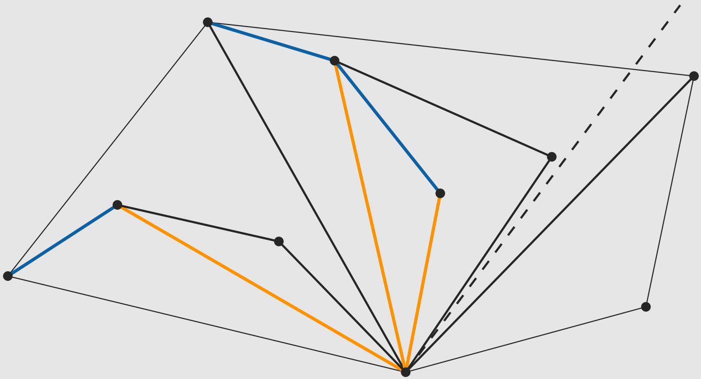
Sweep

- Sweep it counter-clockwise through the point set



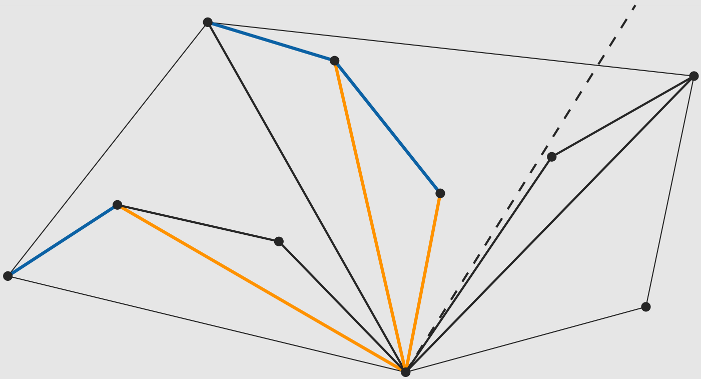
Sweep

- When it passes a vertex:
    - Swap the top and bottom edge, if necessary
    - Flip the top edge



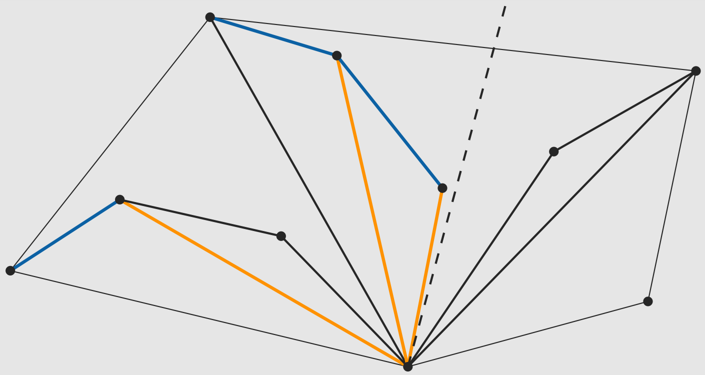
Sweep

- When it passes a vertex:
  - Swap the top and bottom edge, if necessary
  - Flip the top edge



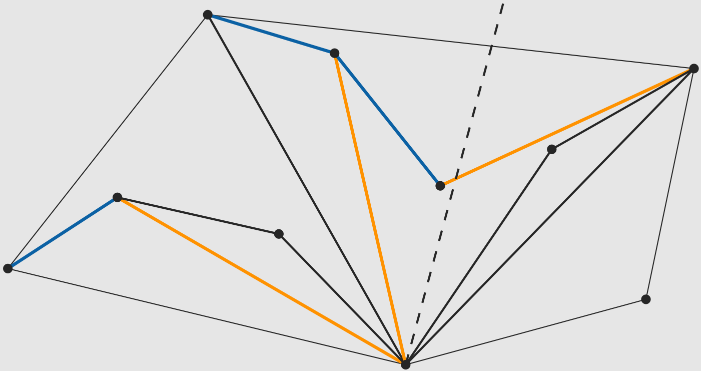
Sweep

- When it passes a vertex:
  - Swap the top and bottom edge, if necessary
  - Flip the top edge



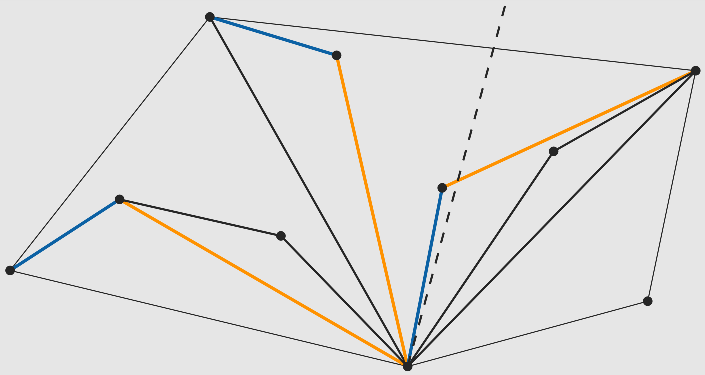
Sweep

- When it passes a vertex:
    - Swap the top and bottom edge, if necessary
    - Flip the top edge



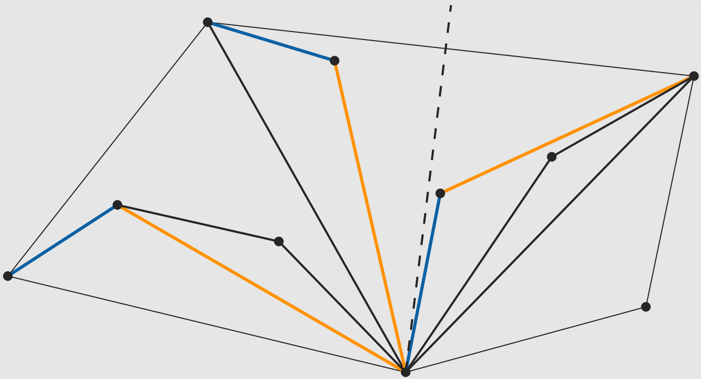
Sweep

- When it passes a vertex:
    - Swap the top and bottom edge, if necessary
    - Flip the top edge



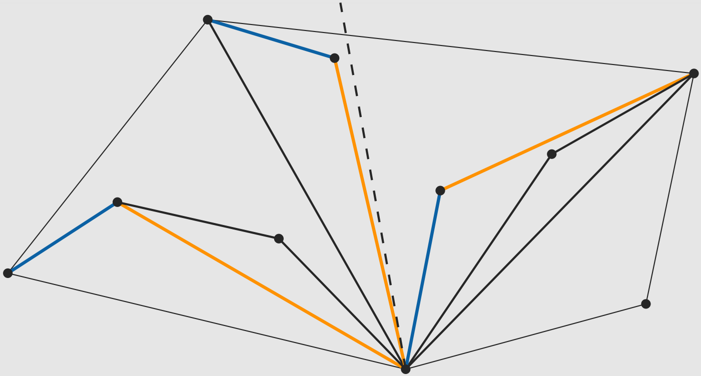
Sweep

- When it passes a vertex:
    - Swap the top and bottom edge, if necessary
    - Flip the top edge



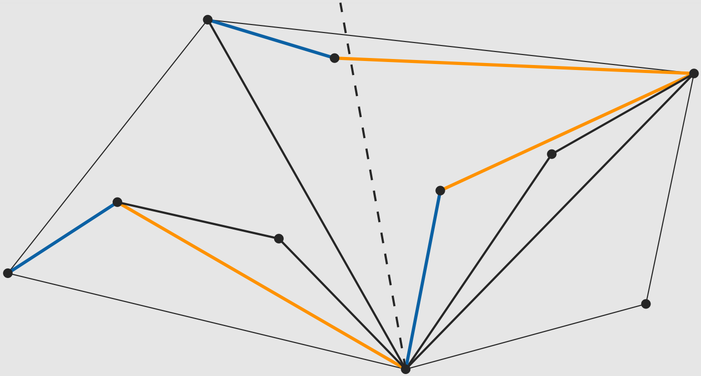
Sweep

- When it passes a vertex:
    - Swap the top and bottom edge, if necessary
    - Flip the top edge



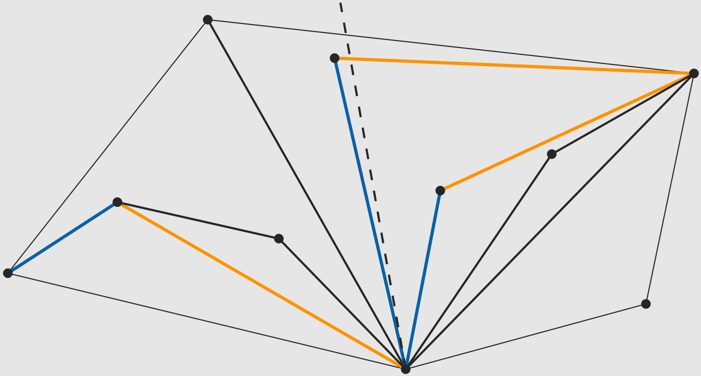
Sweep

- When it passes a vertex:
  - Swap the top and bottom edge, if necessary
  - Flip the top edge



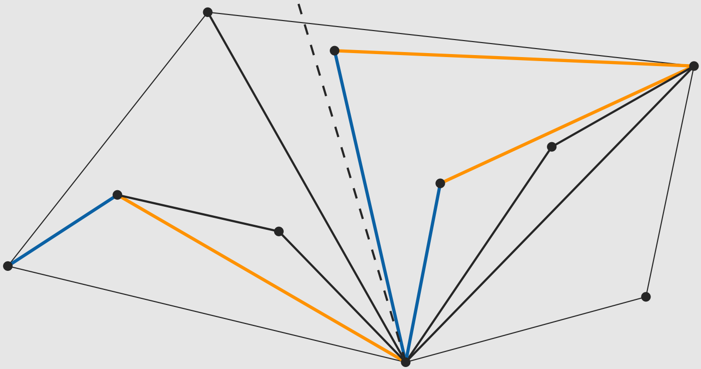
Sweep

- When it passes a vertex:
  - Swap the top and bottom edge, if necessary
  - Flip the top edge



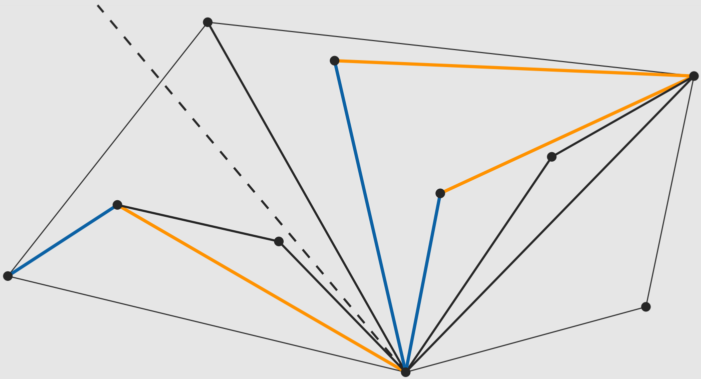
Sweep

- When it passes a vertex:
  - Swap the top and bottom edge, if necessary
  - Flip the top edge



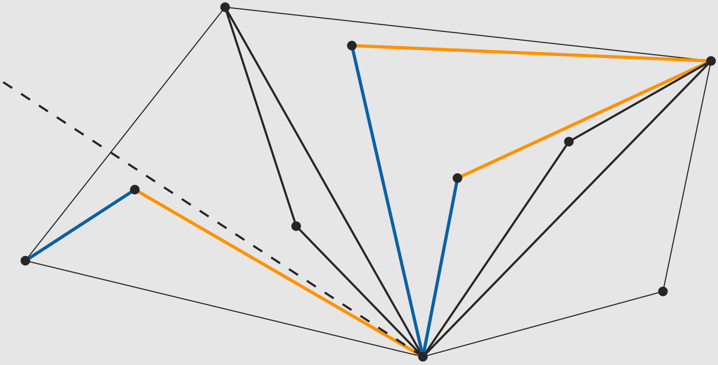
Sweep

- When it passes a vertex:
    - Swap the top and bottom edge, if necessary
    - Flip the top edge



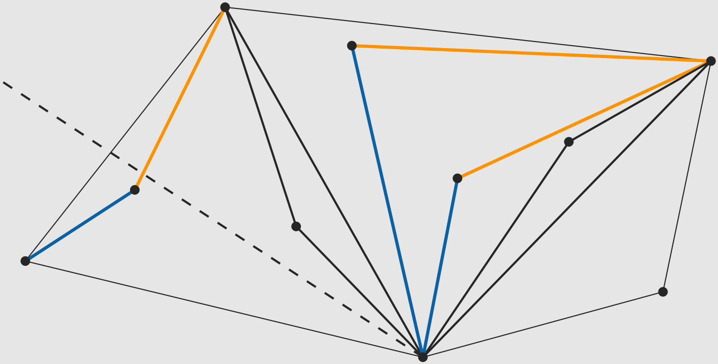
Sweep

- When it passes a vertex:
  - Swap the top and bottom edge, if necessary
  - Flip the top edge



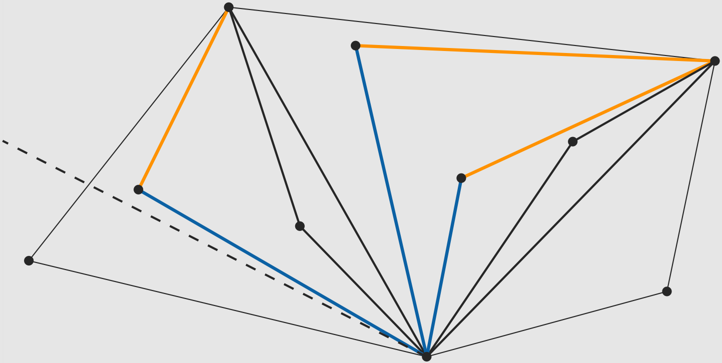
Sweep

- When it passes a vertex:
  - Swap the top and bottom edge, if necessary
  - Flip the top edge



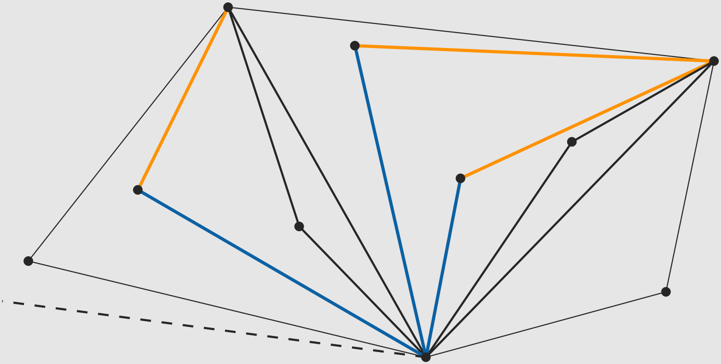
Sweep

- When it passes a vertex:
    - Swap the top and bottom edge, if necessary
    - Flip the top edge



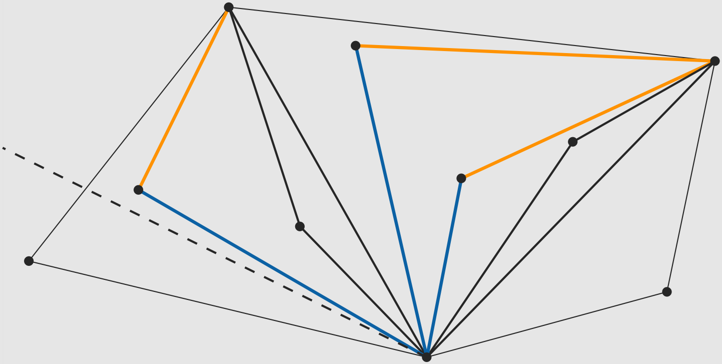
Sweep

- When it passes a vertex:
  - Swap the top and bottom edge, if necessary
  - Flip the top edge



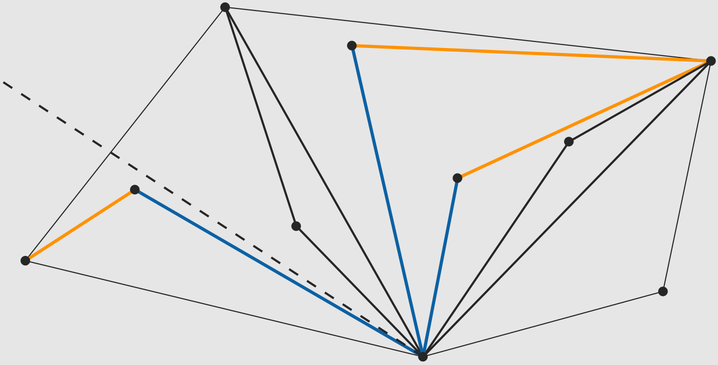
Sweep

- When it passes a vertex:
  - Swap the top and bottom edge, if necessary
  - Flip the top edge



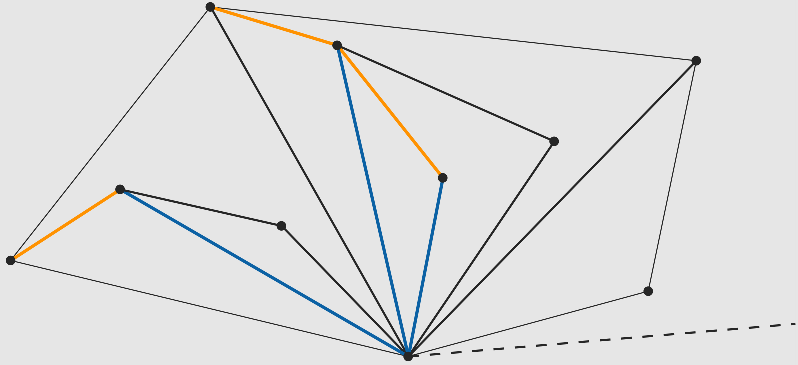
Sweep

- When it passes a vertex:
  - Swap the top and bottom edge, if necessary
  - Flip the top edge



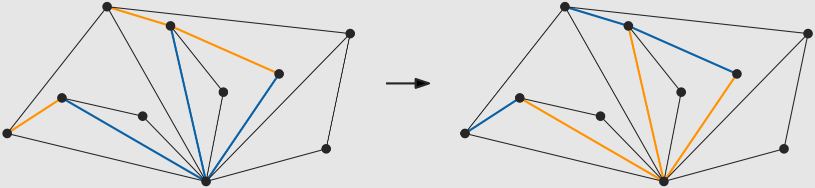
Sweep

- When it passes a vertex:
    - Swap the top and bottom edge, if necessary
    - Flip the top edge

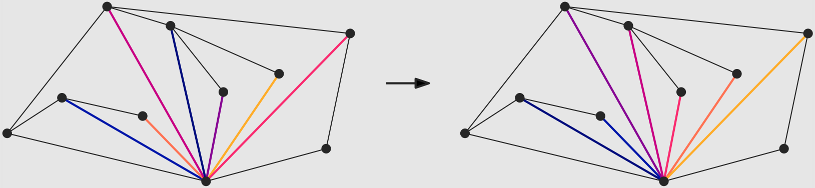


Tools

- *Sweep*: exchange labels on top and bottom pairs — $O(n)$

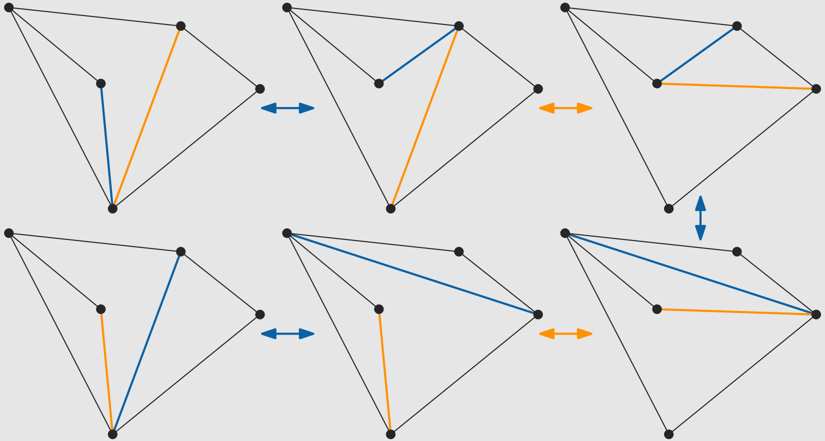


- *Shuffle*: reorder bottom labels



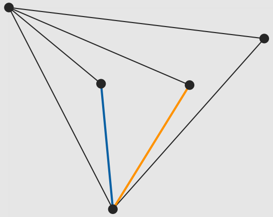
Shuffle

- Swap consecutive bottom edges
  - Easy if third pseudo-triangle is a triangle



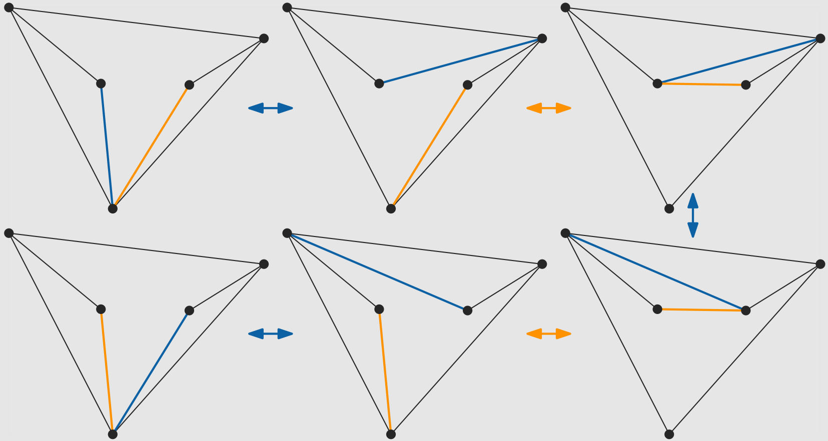
Shuffle

- Swap consecutive bottom edges
  - Easy if third pseudo-triangle is a triangle
  - Otherwise, flip top edge first



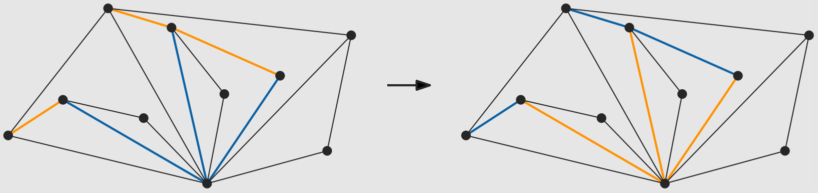
Shuffle

- Swap consecutive bottom edges
  - Easy if third pseudo-triangle is a triangle
  - Otherwise, flip top edge first
- We can do insertion sort!

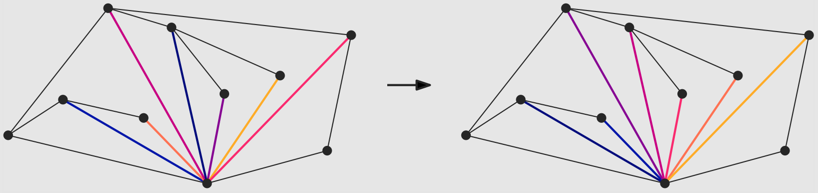


Tools

- *Sweep*: exchange labels on top and bottom pairs — $O(n)$



- *Shuffle*: reorder bottom labels — $O(n^2)$



Upper bound

Theorem

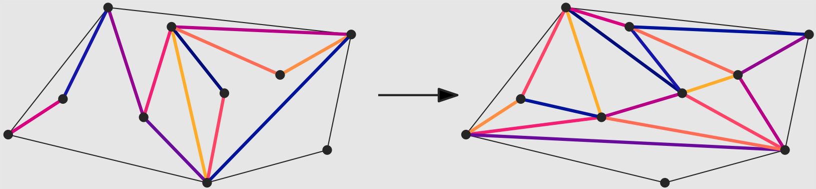
We can sort the labels of a left-shelling pseudo-triangulation with $O(1)$ shuffles and sweeps.

Theorem

We can transform any edge-labelled pointed pseudo-triangulation into any other with $O(n^2)$ flips.

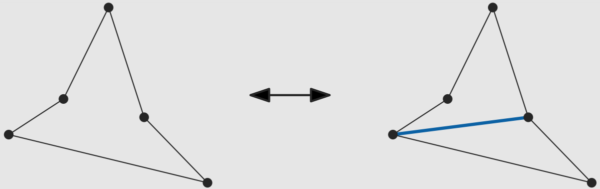
All pseudo-triangulations

- Number of edges differs



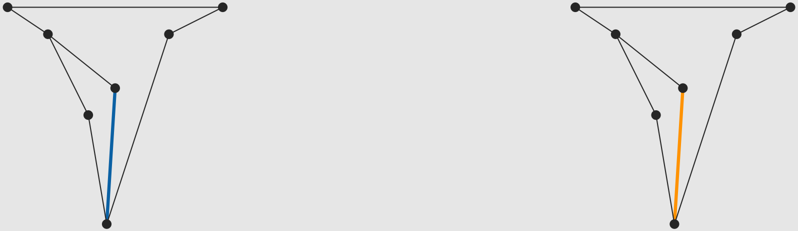
All pseudo-triangulations

- Number of edges differs
- We need *edge-insertion* and *edge-deletion* flips
- Edge label is drawn from / placed into pool of *free labels*



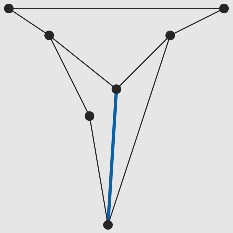
All pseudo-triangulations

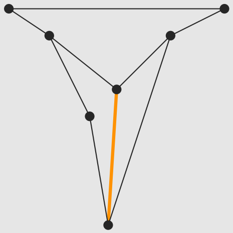
- Swap edges incident to degree-2 vertices



All pseudo-triangulations

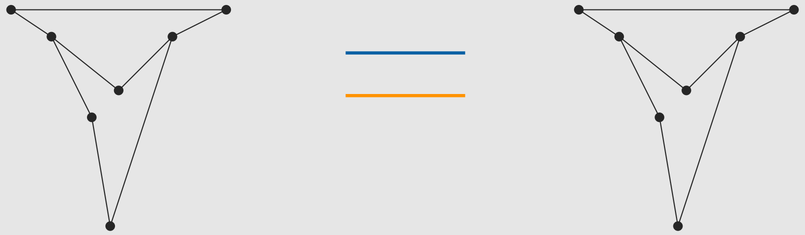
- Swap edges incident to degree-2 vertices





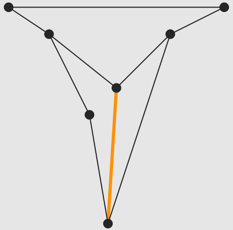
All pseudo-triangulations

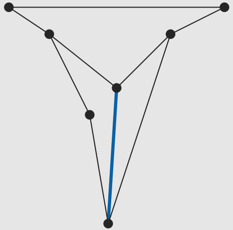
- Swap edges incident to degree-2 vertices



All pseudo-triangulations

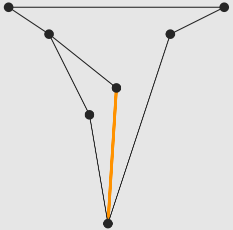
- Swap edges incident to degree-2 vertices

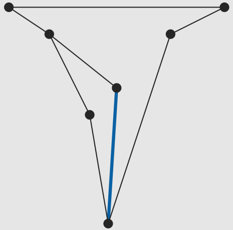




All pseudo-triangulations

- Swap edges incident to degree-2 vertices





Upper bound

Theorem

Using insertion and deletion flips, we can shuffle with $O(n \log n)$ flips.

Theorem

We can transform any edge-labelled pseudo-triangulation into any other with $O(n \log n)$ flips.

Lower bound

- Triangulation of convex polygon
  = pointed pseudo-triangulation

Theorem

There are pairs of edge-labelled pointed pseudo-triangulations such that we need at least $\Omega(n \log n)$ *flips to transform one into the other.*

Summary

- Pointed pseudo-triangulations (only exchanging flips)

$$\Omega(n \log n) \text{ and } O(n^2)$$

- All pseudo-triangulations (all three flip types)

$$\Theta(n \log n)$$