Flips in Edge-Labelled Triangulations

Prosenjit Bose[1] Anna Lubiw[2] Vinayak Pathak[2]
Sander Verdonschot[1]

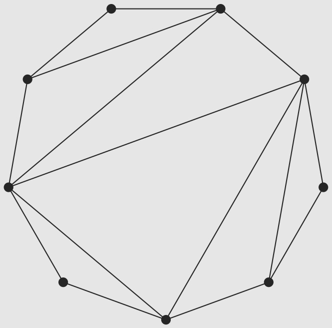
[1]Carleton University

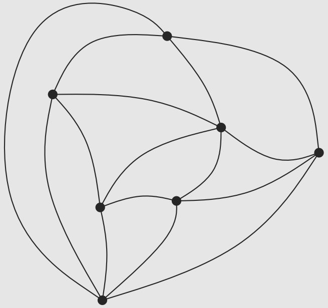
[2]University of Waterloo

February 17, 2016

Triangulations

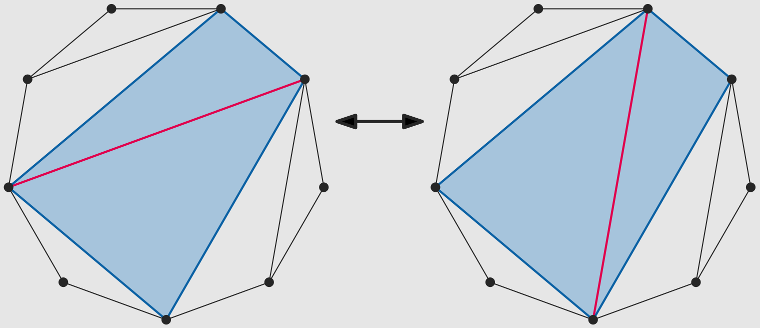
- Graphs where all faces are triangles





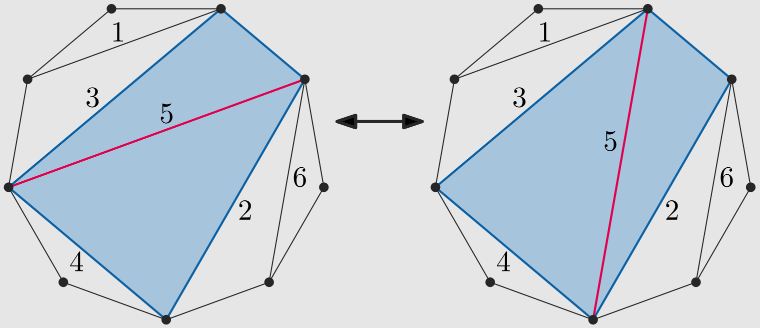
Flips

- Replace edge by other diagonal of quadrilateral



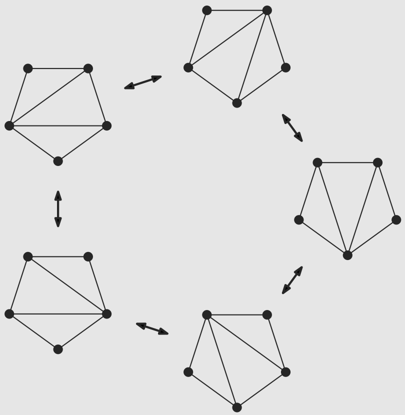
Flips

- Replace edge by other diagonal of quadrilateral
- Diagonals have unique labels



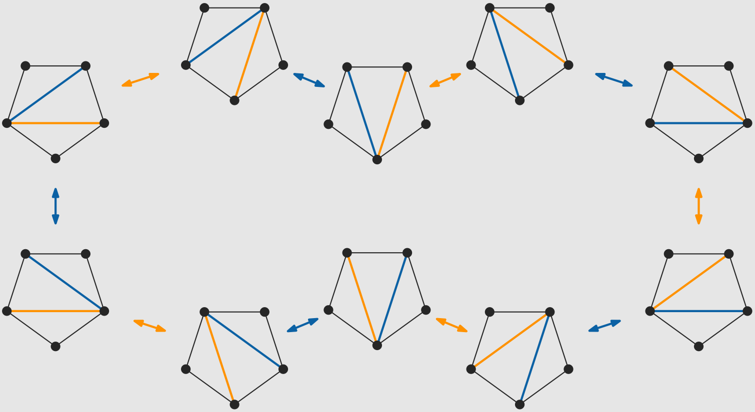
Flip graphs

- Vertex = triangulation, Edge = flip



Flip graphs

- Vertex = triangulation, Edge = flip



History

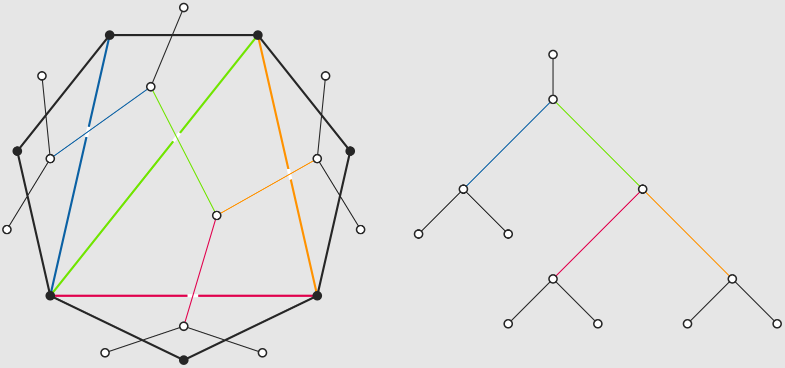
- Introduced by Wagner in 1936
  - Flip graph of combinatorial triangulations is connected
- Diameter:
  - $O(n^2)$ — Wagner, 1936

History

- Introduced by Wagner in 1936
  - Flip graph of combinatorial triangulations is connected
- Diameter:
  - $O(n^2)$ — Wagner, 1936
  - $O(n)$ — Sleator et al., 1992
  - $8n - O(1)$ — Komuro, 1997
  - $6n - O(1)$ — Mori et al., 2001
  - $5.2n - O(1)$ — Bose et al., 2014
  - $5n - O(1)$ — Cardinal et al., 2015

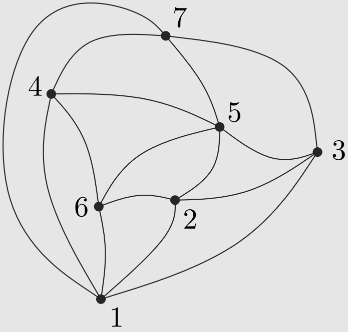
History

- Triangulation of convex polygon = binary tree
- Diameter $= 2n - 10$ — Sleator et al., 1988



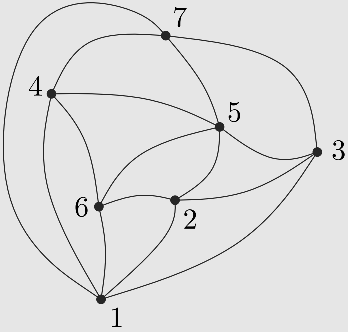
History

- What happens when the vertices are labelled?
  - Diameter is $\Theta(n \log n)$ - Sleator et al., 1992



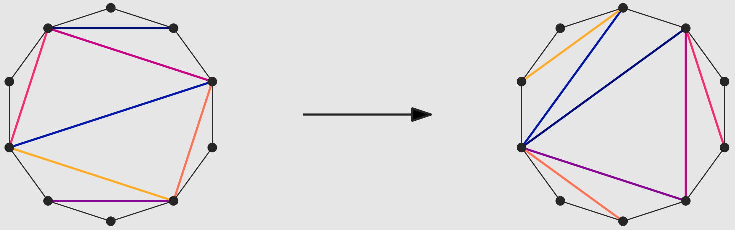
History

- What happens when the vertices are labelled?
  - Diameter is $\Theta(n \log n)$ - Sleator et al., 1992
- What happens when edges are labelled?



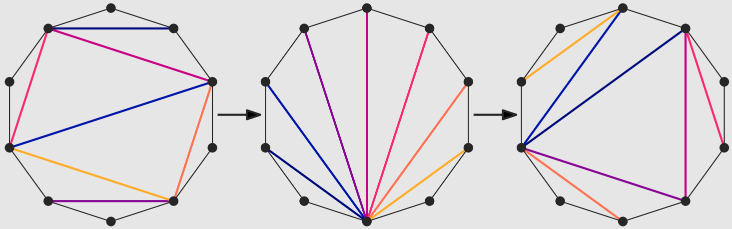
Upper bound

- Transform T_1 into T_2



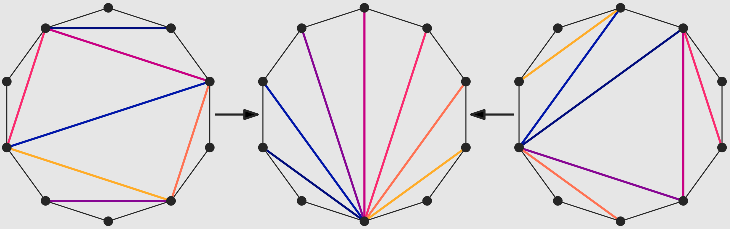
Upper bound

- Transform T_1 into T_2
- Via canonical form T_C



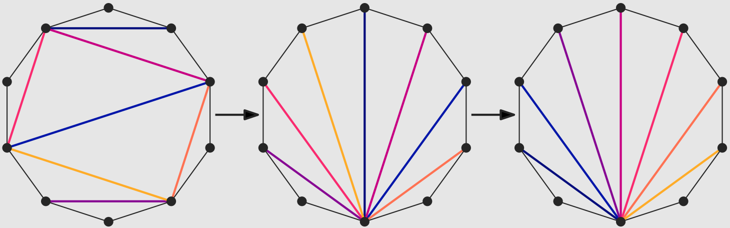
Upper bound

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



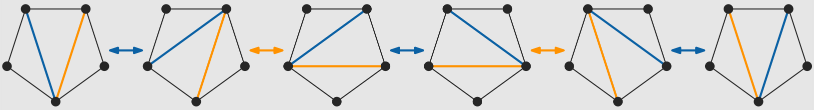
Transform into canonical

- Ignore labels
- Sort



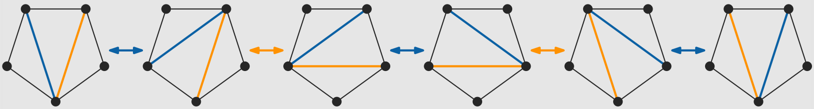
Sorting

- We can exchange adjacent diagonals



Sorting

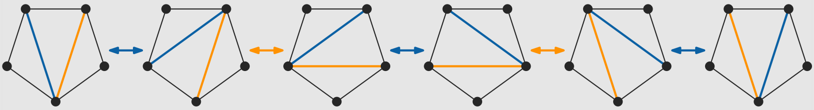
- We can exchange adjacent diagonals



- We can do insertion sort

Sorting

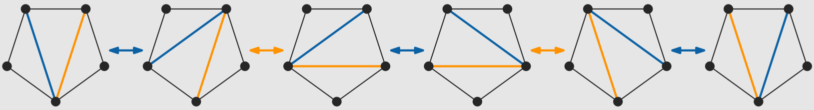
- We can exchange adjacent diagonals



- We can do insertion sort
  - Flip graph is connected!
  - Diameter is $O(n^2)$

Sorting

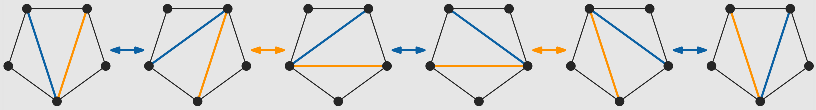
- We can exchange adjacent diagonals



- We can do insertion sort
  - Flip graph is connected!
  - Diameter is $O(n^2)$
- Can we do better?

Sorting

- We can exchange adjacent diagonals



- We can do insertion sort
  - Flip graph is connected!
  - Diameter is $O(n^2)$
- Can we do better?
  - Yes! Simulating quicksort gives us $O(n \log n)$

Lower bound

Theorem (Sleator, Tarjan, and Thurston, 1992)

Given a triangulation T of a convex polygon, the number of triangulations reachable from T by a sequence of m flips is at most $2^{O(n+m)}$, regardless of labellings.

Lower bound

Theorem (Sleator, Tarjan, and Thurston, 1992)

Given a triangulation T of a convex polygon, the number of triangulations reachable from T by a sequence of m flips is at most $2^{O(n+m)}$, regardless of labellings.

- There are over $n!$ edge-labelled triangulations:

$$
\begin{aligned}
2^{O(n+d)} &\geqslant n! \\
O(n+d) &\geqslant \log n! \\
d &\geqslant \Omega(n \log n)
\end{aligned}
$$

Lower bound

Theorem (Sleator, Tarjan, and Thurston, 1992)

Given a triangulation T of a convex polygon, the number of triangulations reachable from T by a sequence of m flips is at most $2^{O(n+m)}$, regardless of labellings.

- There are over $n!$ edge-labelled triangulations:

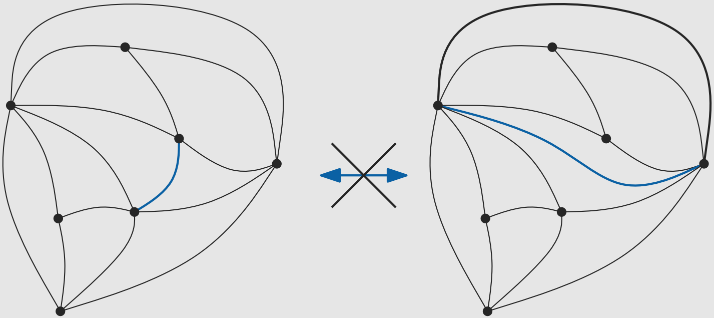
$$\begin{aligned} 2^{O(n+d)} &\geqslant n! \\ O(n+d) &\geqslant \log n! \\ d &\geqslant \Omega(n \log n) \end{aligned}$$

Theorem

The diameter of the flip graph is $\Theta(n \log n)$.

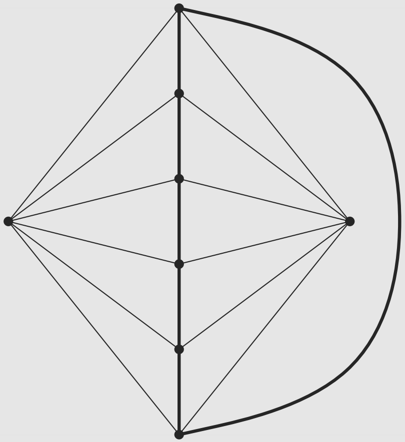
Combinatorial triangulations

- Not all flips are valid



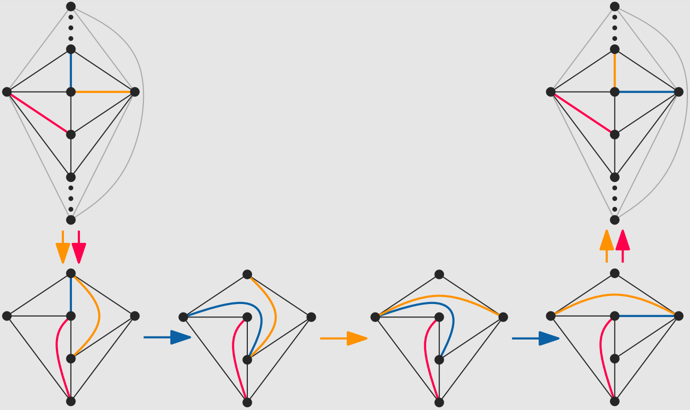
Combinatorial triangulations

- Transform to a canonical form — $O(n)$
- Sort the labels — ?



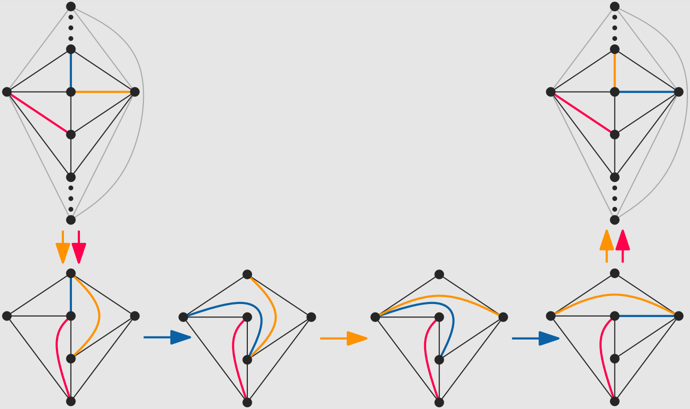
Combinatorial triangulations

- Exchange spine edge with incident non-spine edge



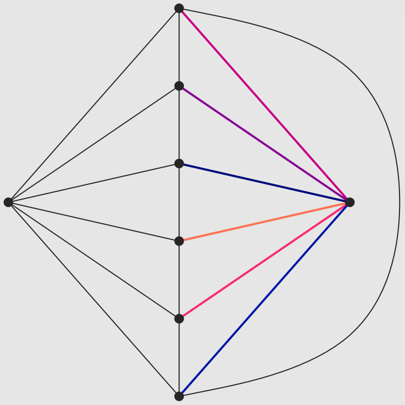
Combinatorial triangulations

- Exchange spine edge with incident non-spine edge
- Flip graph is connected!



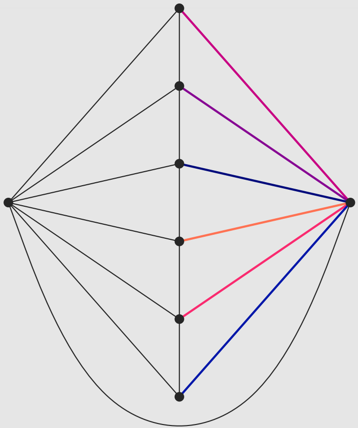
Combinatorial triangulations

- Faster: reorder all labels around inner vertex at the same time



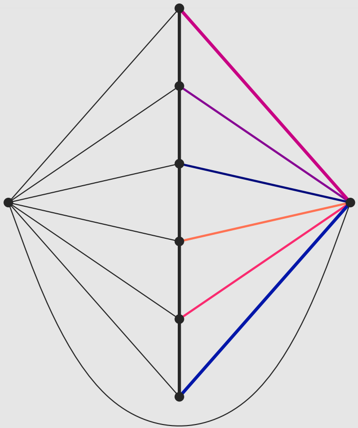
Combinatorial triangulations

- Faster: reorder all labels around inner vertex at the same time
  - Flip external edge



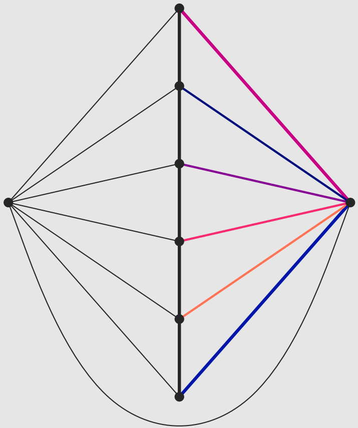
Combinatorial triangulations

- Faster: reorder all labels around inner vertex at the same time
  - Flip external edge
  - Use convex polygon result



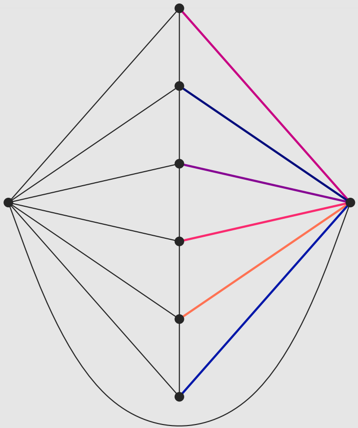
Combinatorial triangulations

- Faster: reorder all labels around inner vertex at the same time
  - Flip external edge
  - Use convex polygon result



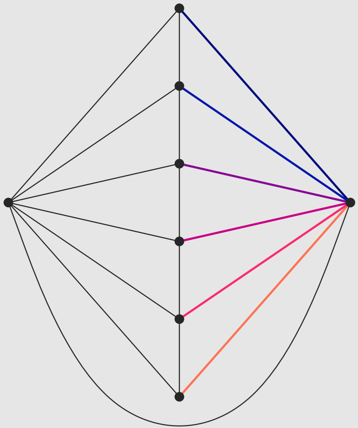
Combinatorial triangulations

- Faster: reorder all labels around inner vertex at the same time
  - Flip external edge
  - Use convex polygon result
  - Swap boundary edges in



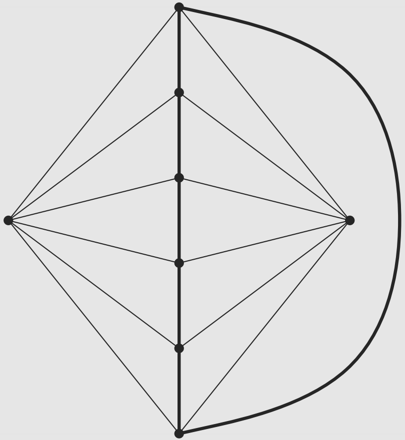
Combinatorial triangulations

- Faster: reorder all labels around inner vertex at the same time
  - Flip external edge — $O(1)$
  - Use convex polygon result — $O(n \log n)$
  - Swap boundary edges in — $O(n)$



Combinatorial triangulations

- Transform to a canonical form — $O(n)$
- Sort the labels — $O(n \log n)$



Combinatorial triangulations

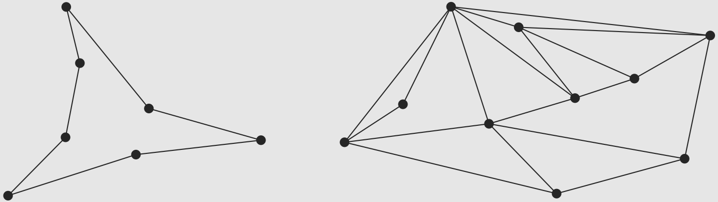
- Transform to a canonical form — $O(n)$
- Sort the labels — $O(n \log n)$

Theorem

The diameter of the flip graph is $\Theta(n \log n)$.

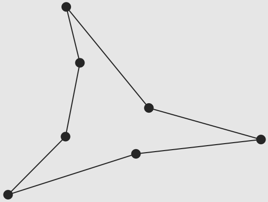
Pseudo-triangulations

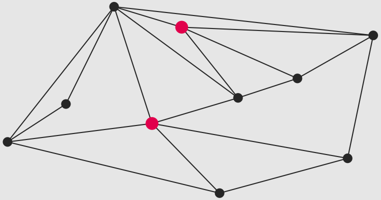
- All faces are pseudo-triangles



Pseudo-triangulations

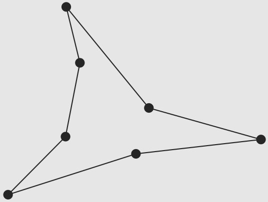
- All faces are pseudo-triangles
- Pointed: all vertices are incident to a reflex angle ($> \pi$)

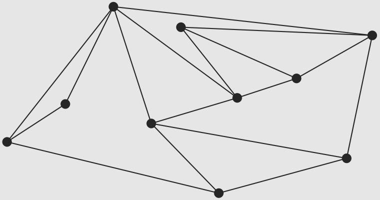




Pseudo-triangulations

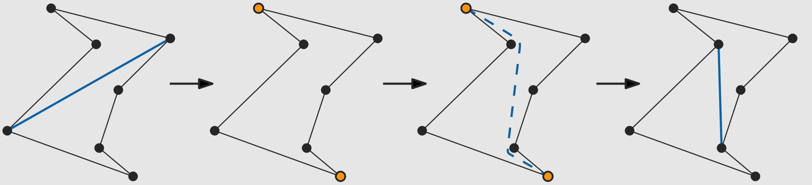
- All faces are pseudo-triangles
- Pointed: all vertices are incident to a reflex angle ($> \pi$)





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

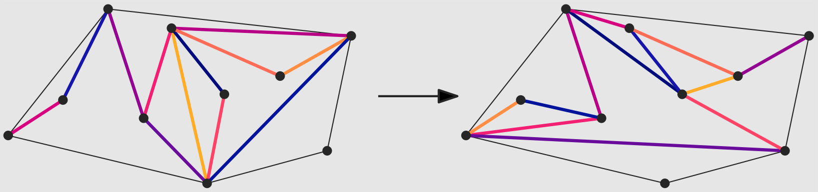
Theorem (Bereg, 2004)

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

- What happens when edges are labelled?

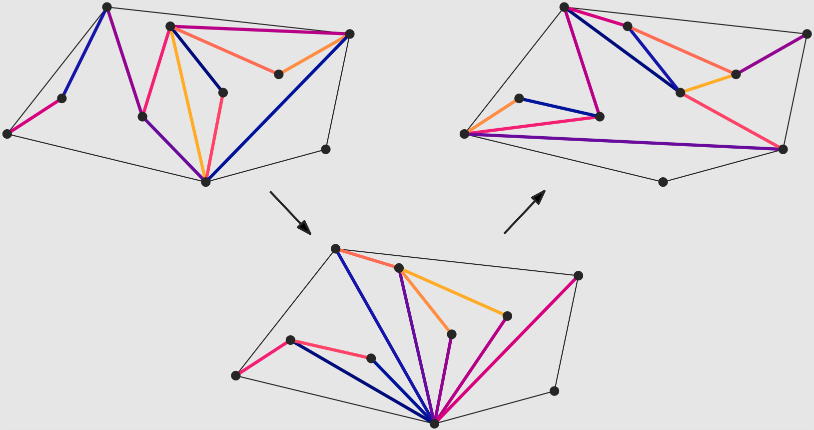
Upper bound

- Transform T_1 into T_2



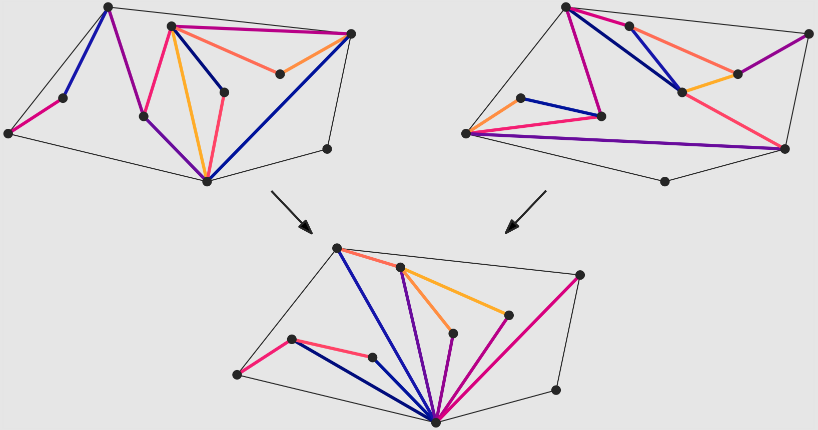
Upper bound

- Transform T_1 into T_2
- Via canonical form T_C



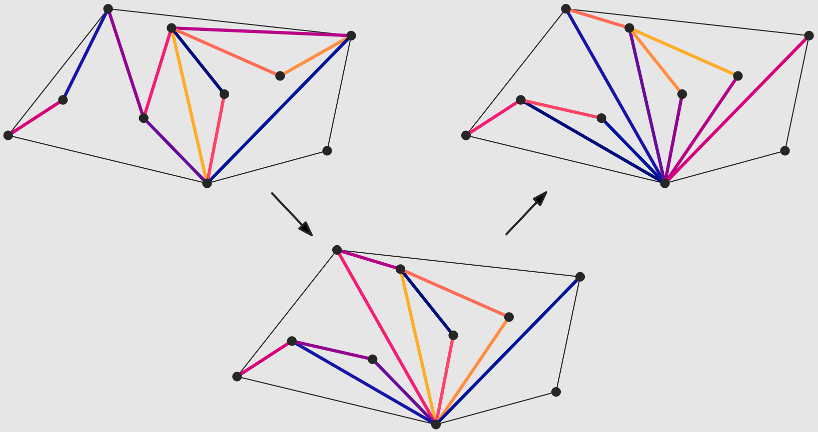
Upper bound

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



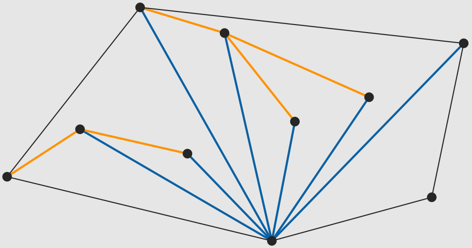
Transform into canonical

- Ignore labels
- Move labels around



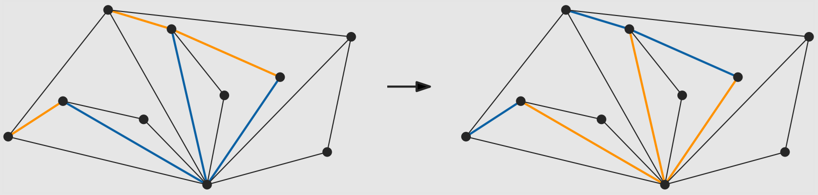
Left-shelling pseudo-triangulation

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

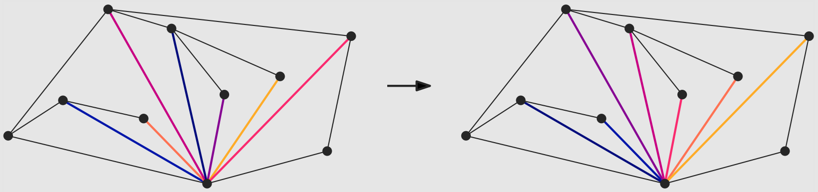


Tools

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

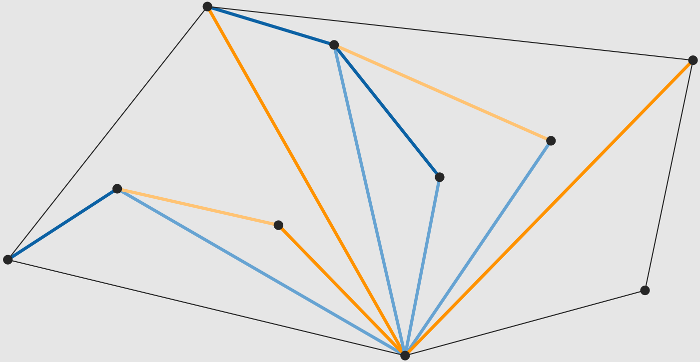


- *Shuffle*: reorder bottom labels



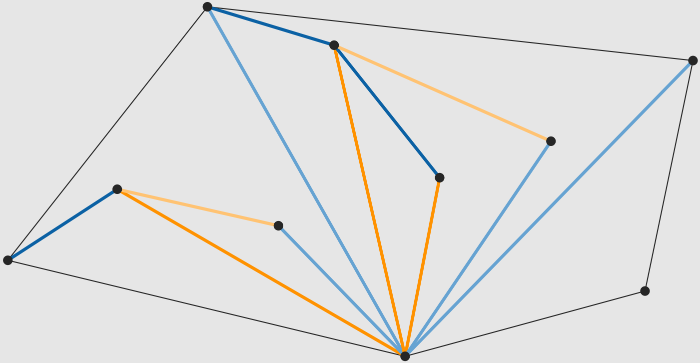
Algorithm

- Identify out-of-place top and bottom labels



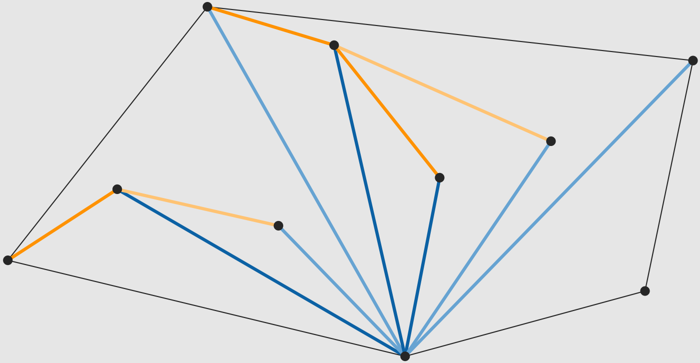
Algorithm

- Pair these up (*Shuffle*)



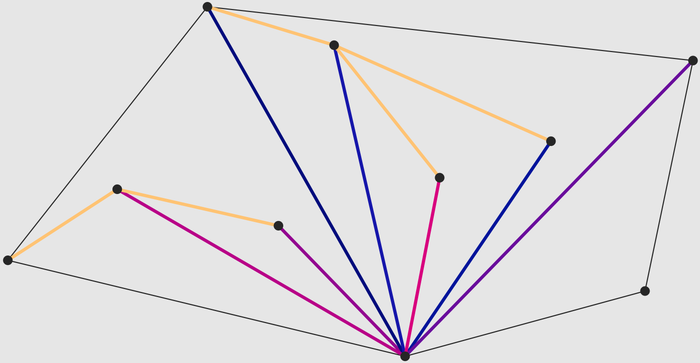
Algorithm

- Exchange them (*Sweep*)



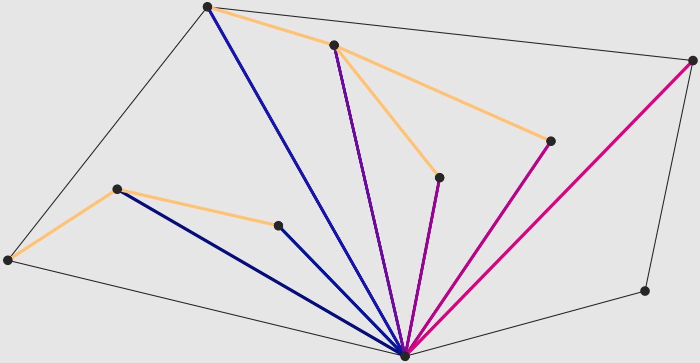
Algorithm

- Sort bottom labels (*Shuffle*)



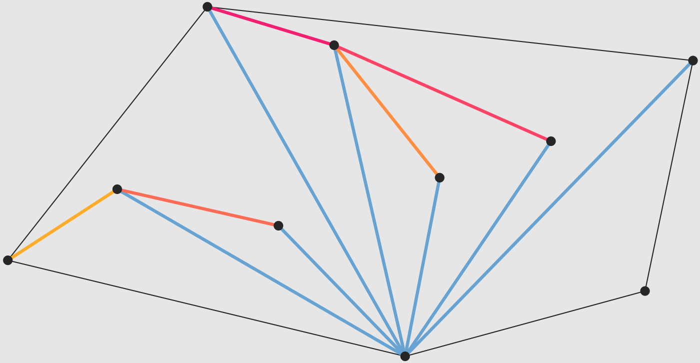
Algorithm

- Sort bottom labels (*Shuffle*)



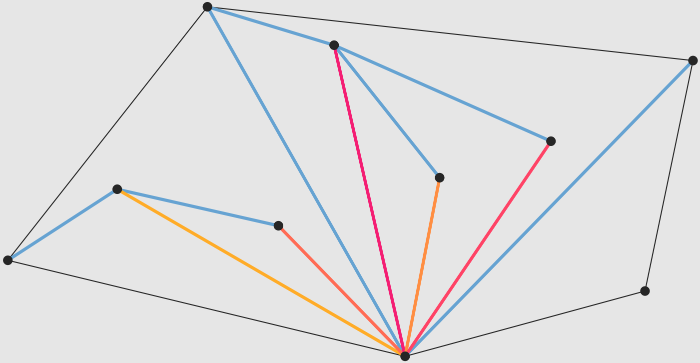
Algorithm

- Move all top labels down (*Sweep*)



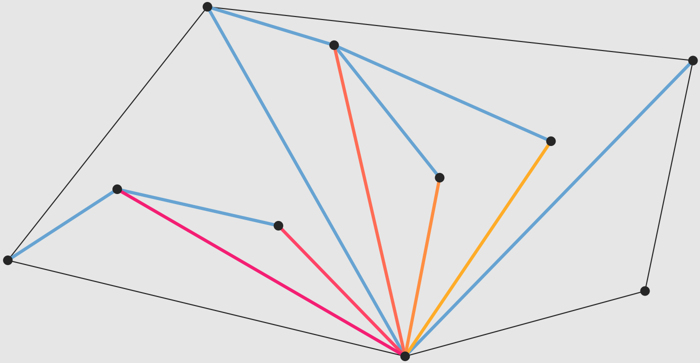
Algorithm

- Move all top labels down (*Sweep*)



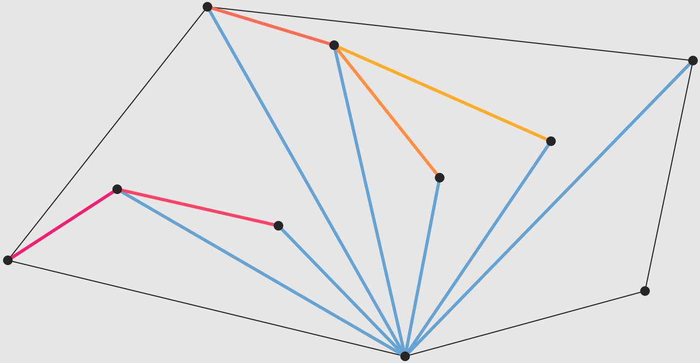
Algorithm

- 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

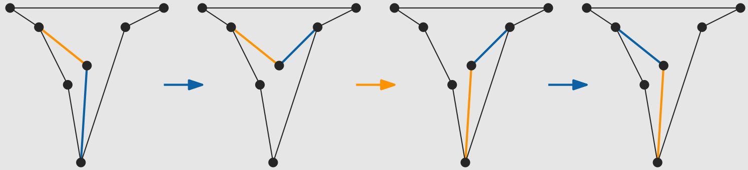
Theorem

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

- How do we shuffle and sweep?

Sweep

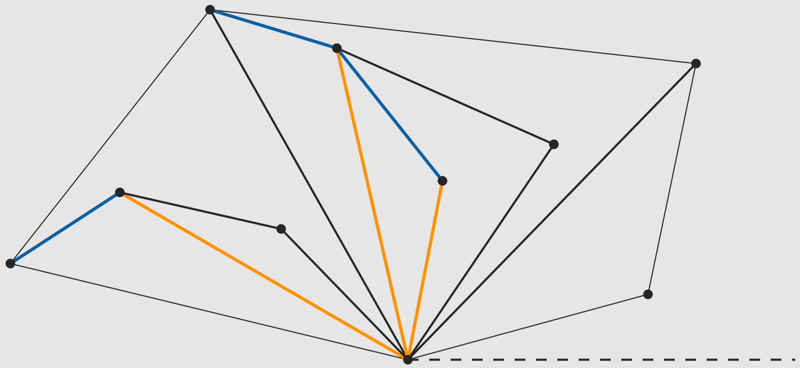
- Easy for degree-2 vertices:



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

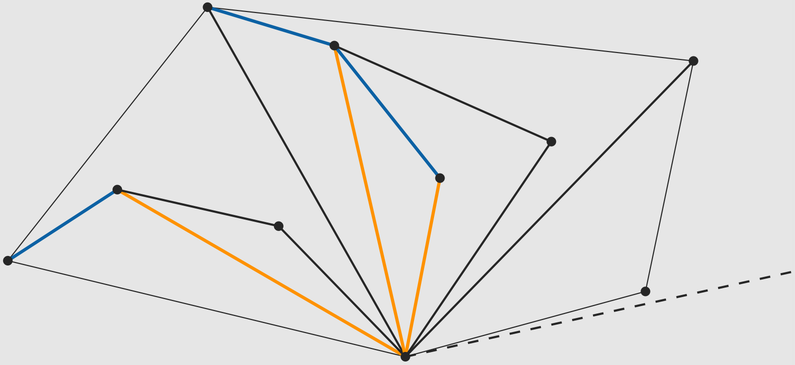
Sweep

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



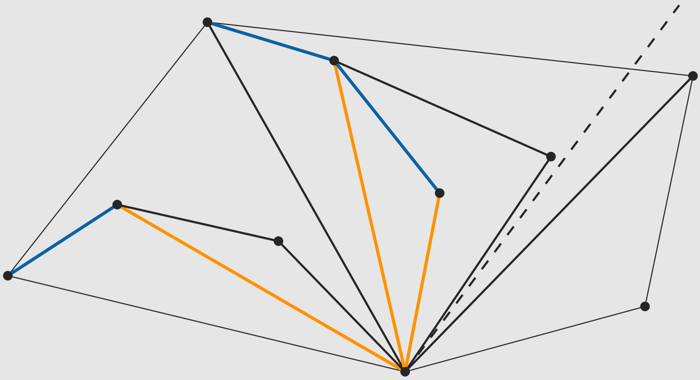
Sweep

- Sweep it counter-clockwise through the point set



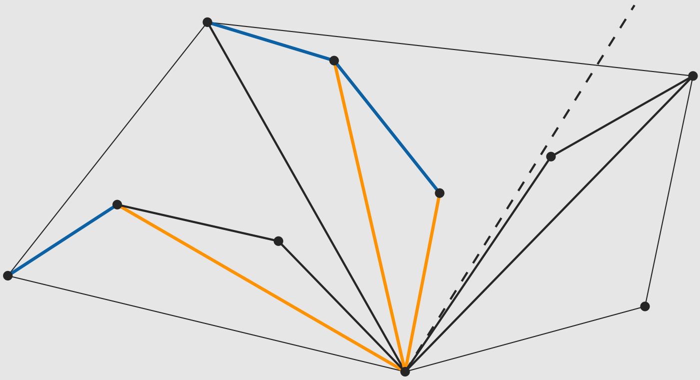
Sweep

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



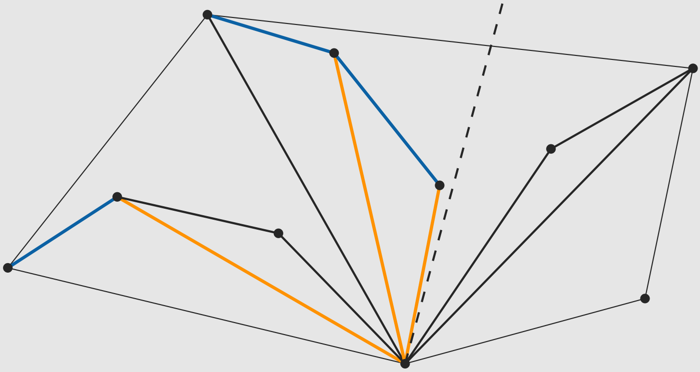
Sweep

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



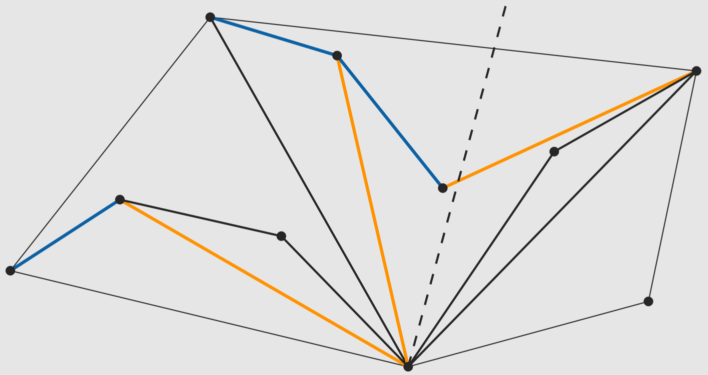
Sweep

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



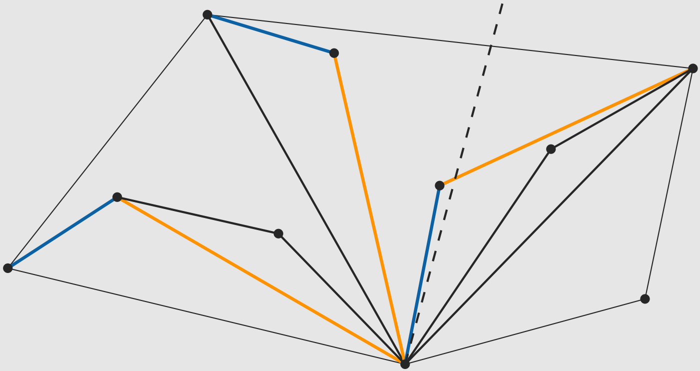
Sweep

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



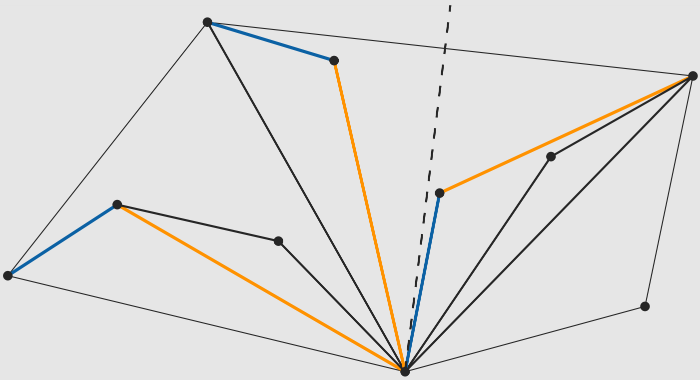
Sweep

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



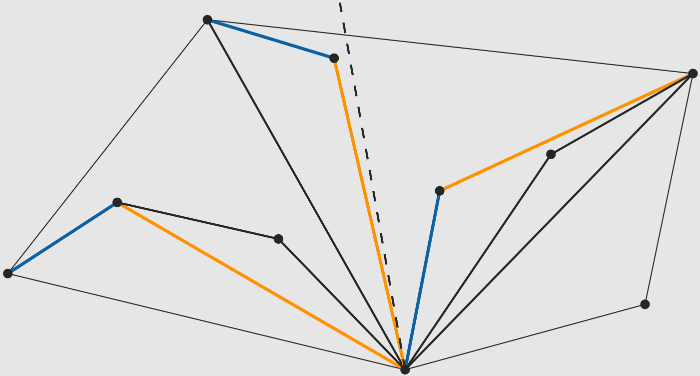
Sweep

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



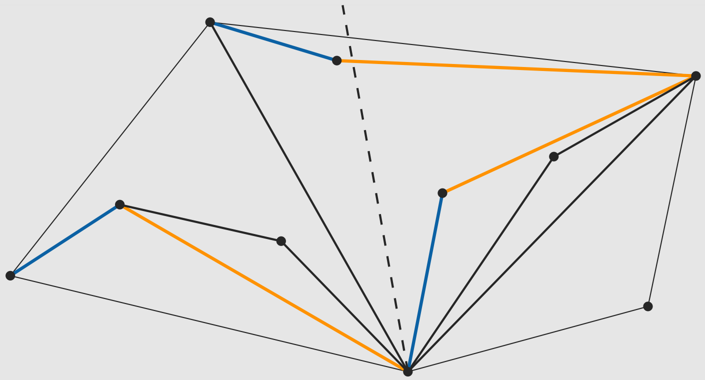
Sweep

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



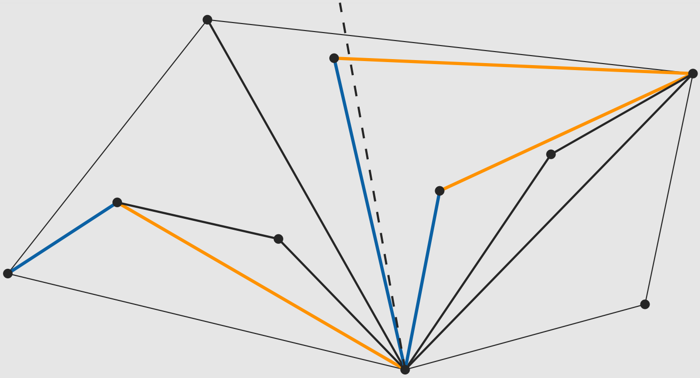
Sweep

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



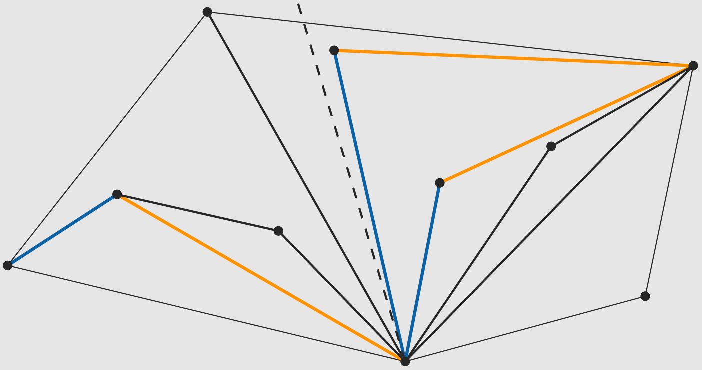
Sweep

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



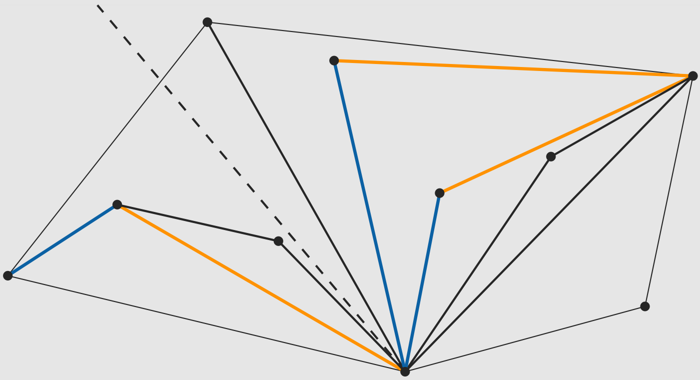
Sweep

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



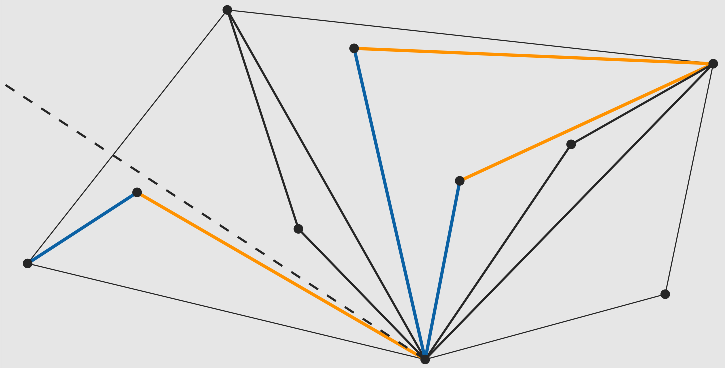
Sweep

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



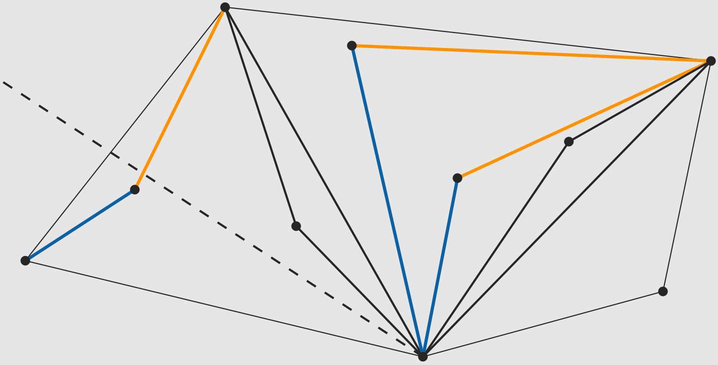
Sweep

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



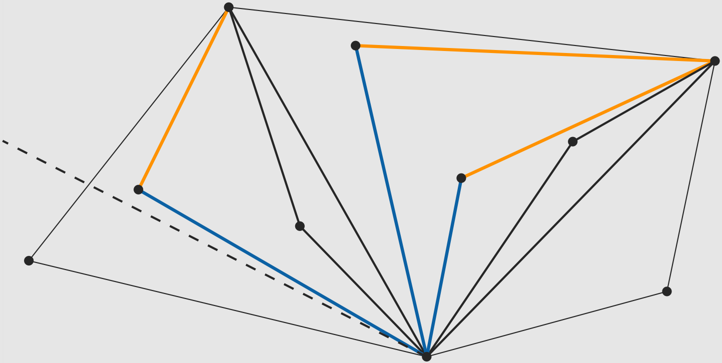
Sweep

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



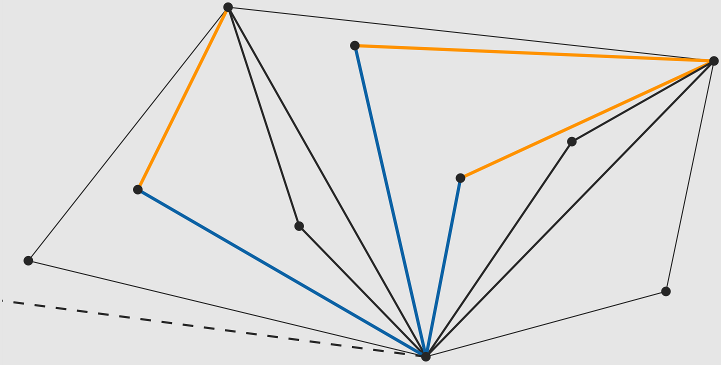
Sweep

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



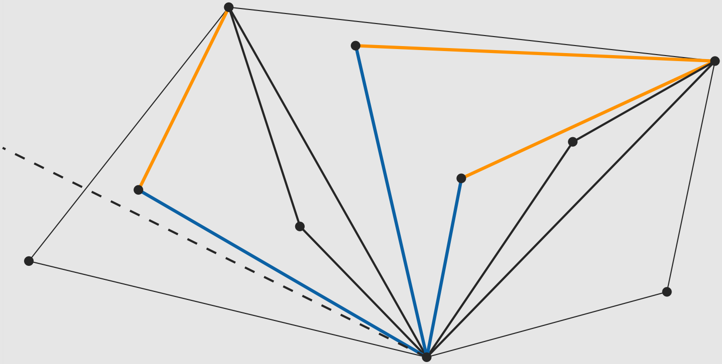
Sweep

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



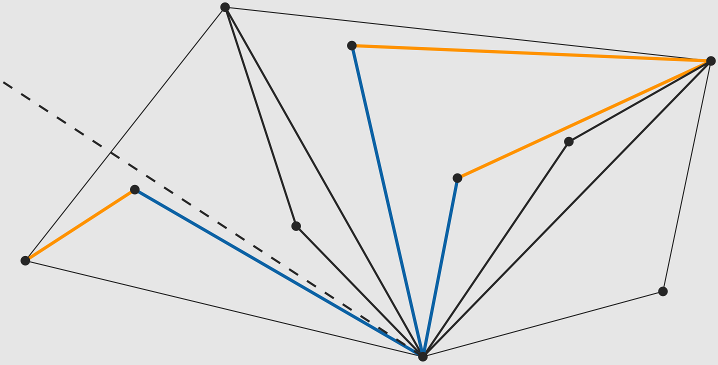
Sweep

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



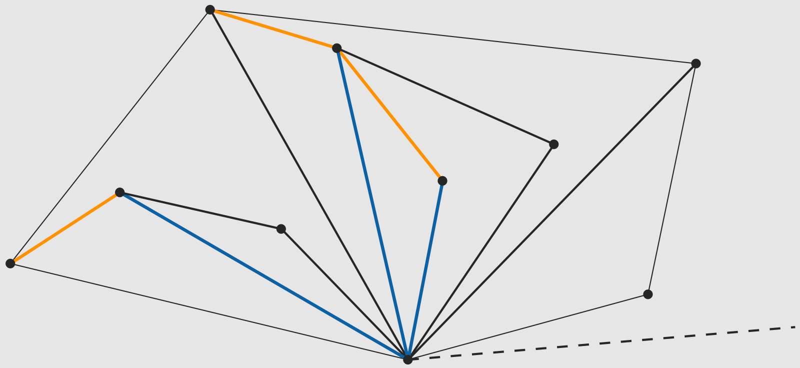
Sweep

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



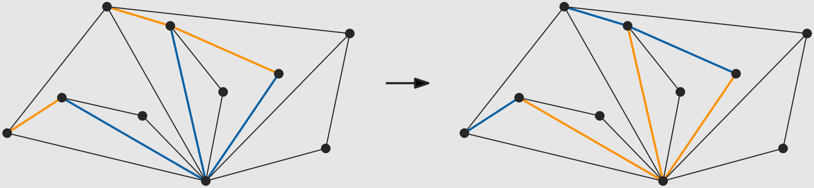
Sweep

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

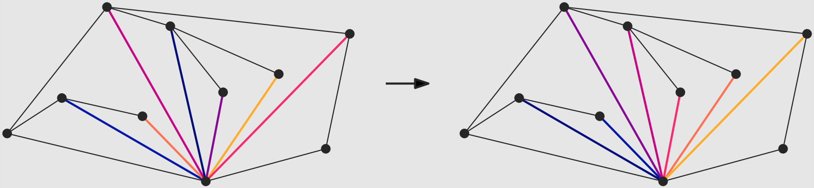


Tools

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

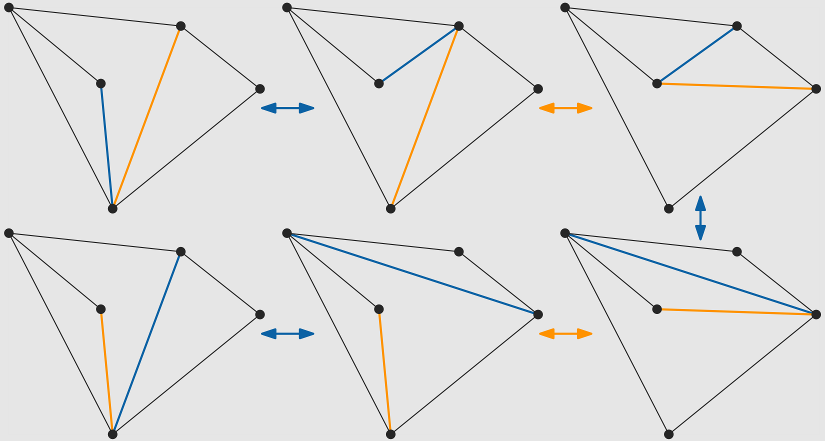


- *Shuffle*: reorder bottom labels



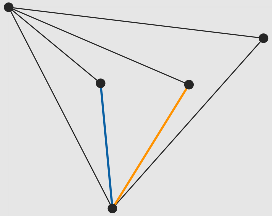
Shuffle

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



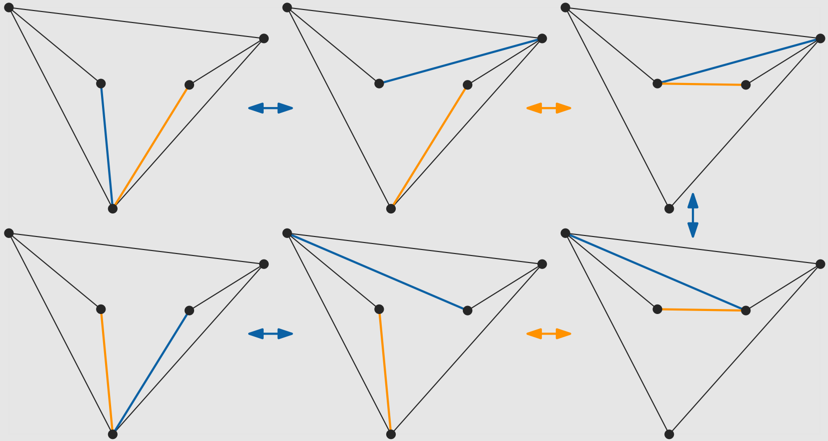
Shuffle

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



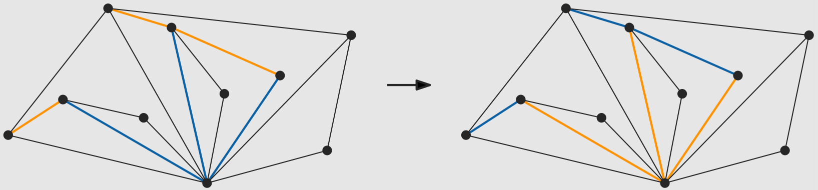
Shuffle

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

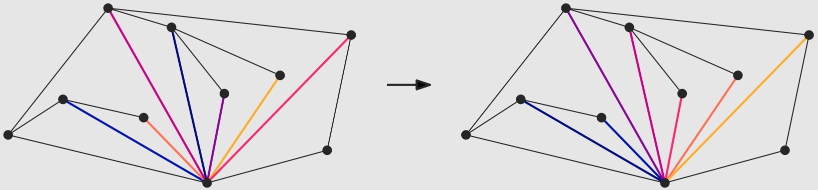


Tools

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



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



Upper bound

Theorem

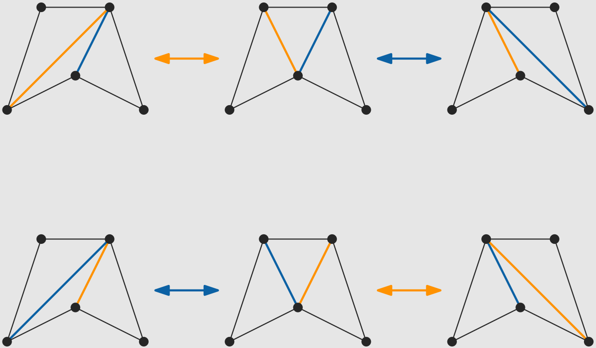
We can sort the labels of a left-shelling pseudo-triangulation with $O(1)$ shuffles and sweeps.

Theorem

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

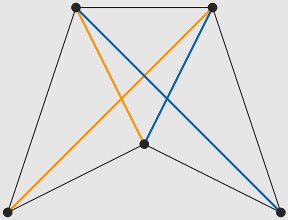
General polygons

- Flip graph might be disconnected



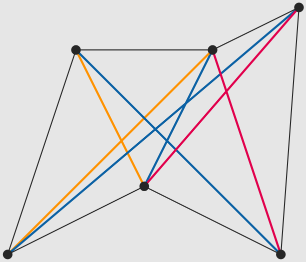
General polygons

- Diagonals form equivalence classes (*orbits*)



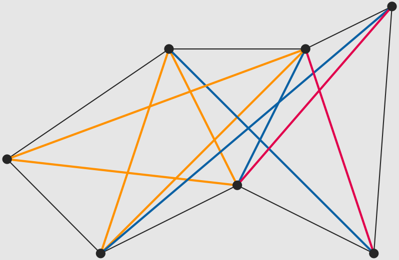
General polygons

- Diagonals form equivalence classes (*orbits*)



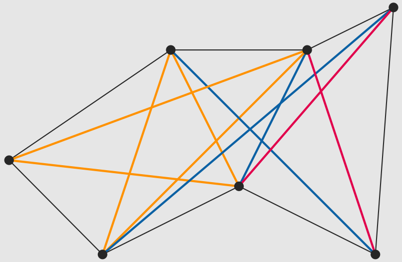
General polygons

- Diagonals form equivalence classes (*orbits*)



General polygons

- Diagonals form equivalence classes (*orbits*)
- *Orbit Conjecture*: We can transform T_1 into T_2 iff edges with the same label are in the same orbit
  - Clearly necessary
  - True for spiral polygons



Open problems

- Settle the Orbit Conjecture for general polygons and triangulations of points in the plane

Open problems

- Settle the Orbit Conjecture for general polygons and triangulations of points in the plane

- Is it NP-hard to compute the flip distance between two edge-labelled triangulations?
  - Variation: allow duplicate labels

Open problems

- Settle the Orbit Conjecture for general polygons and triangulations of points in the plane
- Is it NP-hard to compute the flip distance between two edge-labelled triangulations?
  - Variation: allow duplicate labels
- Can we reduce the $O(n^2)$ upper bound for pointed pseudo-triangulations?