Schematization of Road Networks

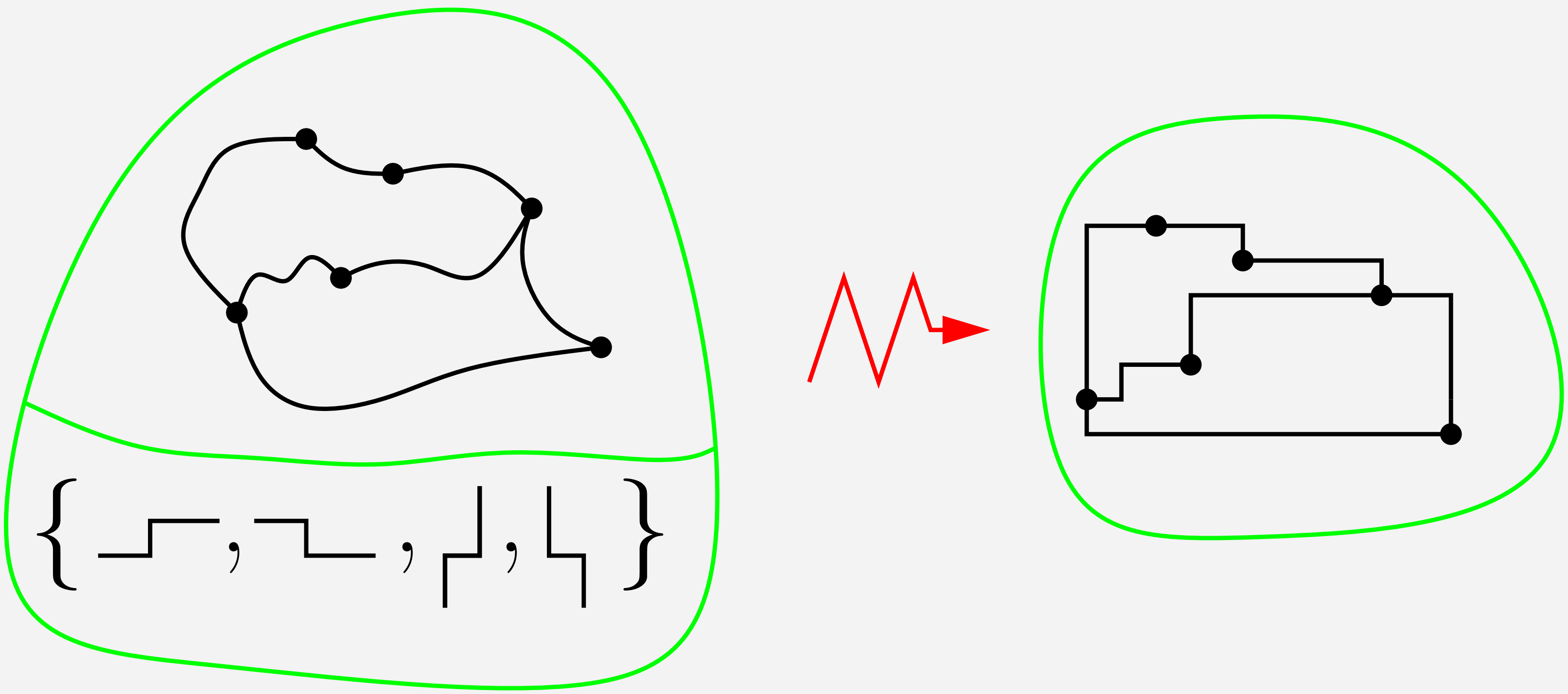
S. Cabello, M. de Berg, S. van Dijk,

M. van Kreveld, T. Strijk

What is a schematic map?

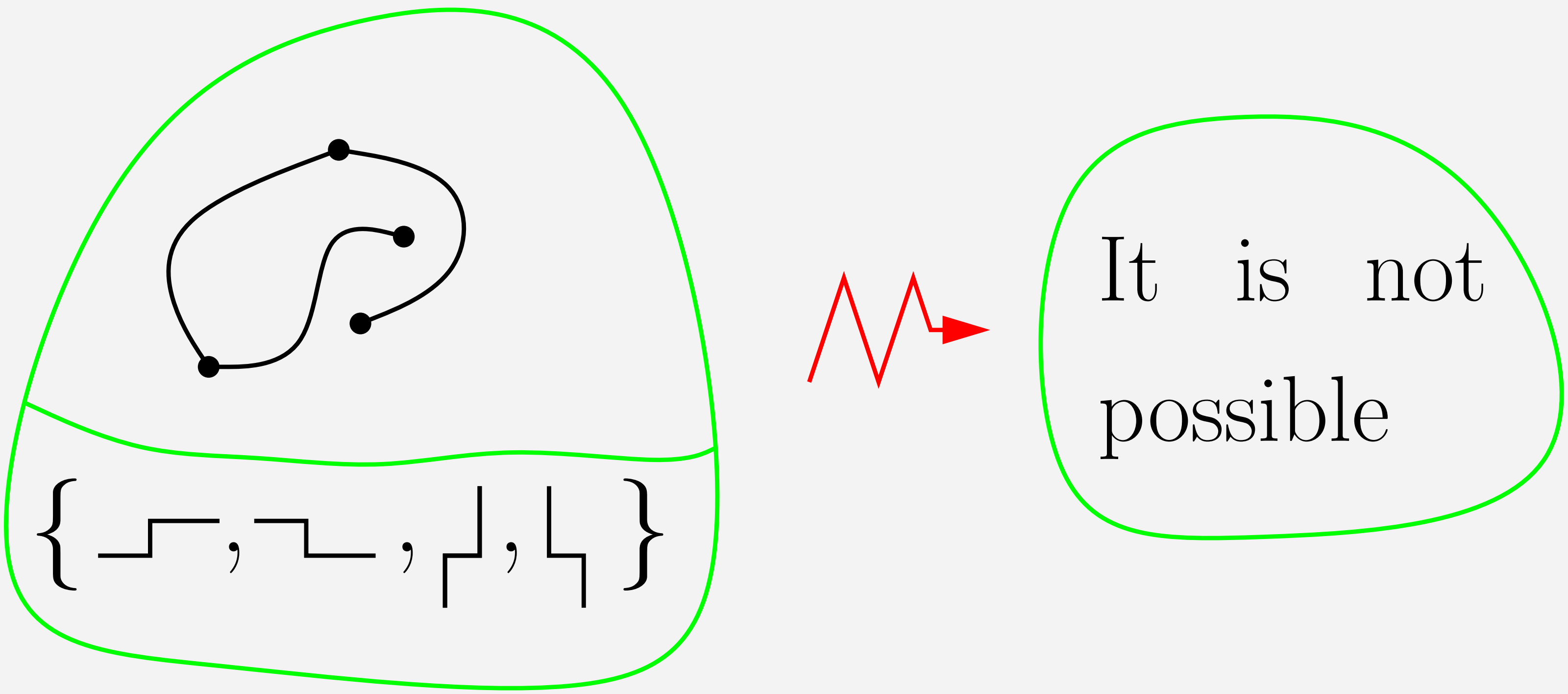
Drastic simplification and conceptualisation of connections.

Framework in our approach



The endpoints of the connections are fixed.

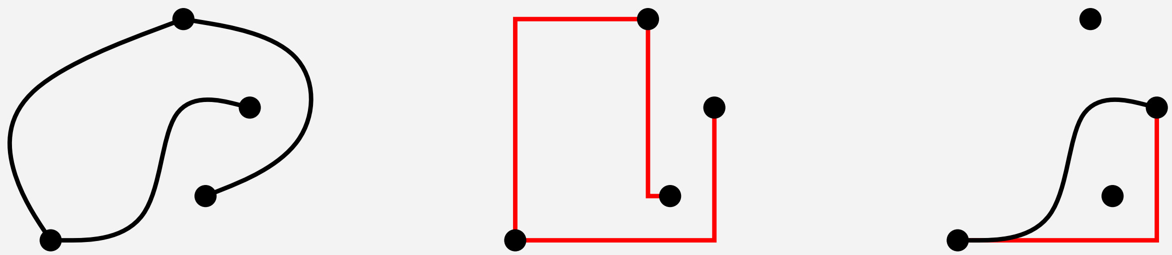
Framework in our approach



The endpoints of the connections are fixed.

Schematic map: deformation of the original

We want to keep some properties in the schematic map...



$P_M \equiv$ Endpoints of map M.

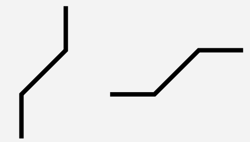
Definition: M and M' are equivalent iff $P_M = P_{M'}$ and each path of M can be continuously deformed in $\mathbb{R}^2 \setminus P_M$ to a path of M'.

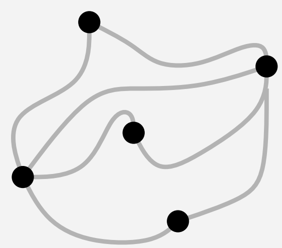
Problem: Given M, find an equivalent M' with 'nice' connections.

Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

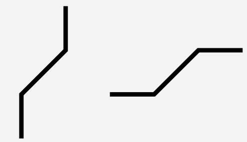
Allowed connections: 



Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

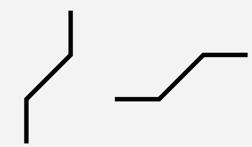
Allowed connections: 

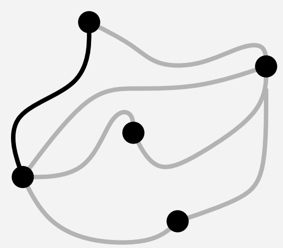
Idea behind everything

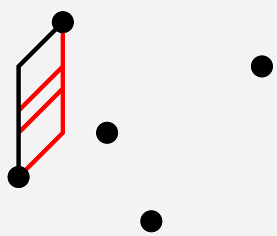
How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections:





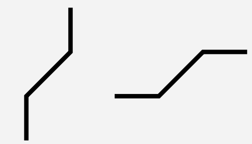


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

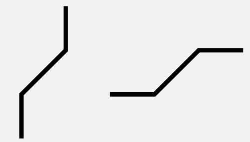
Allowed connections:

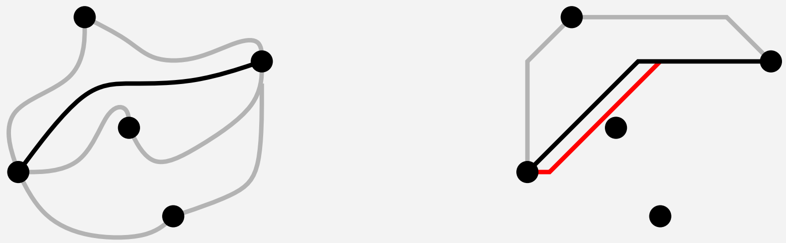


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

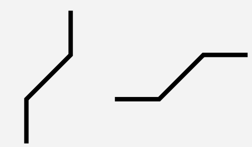
Allowed connections: 

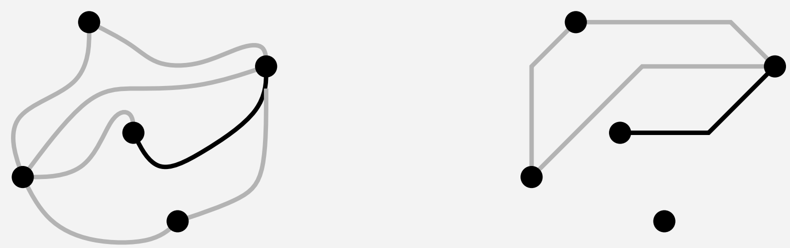


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

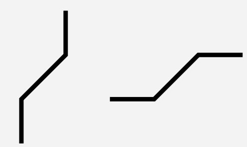
Allowed connections: 

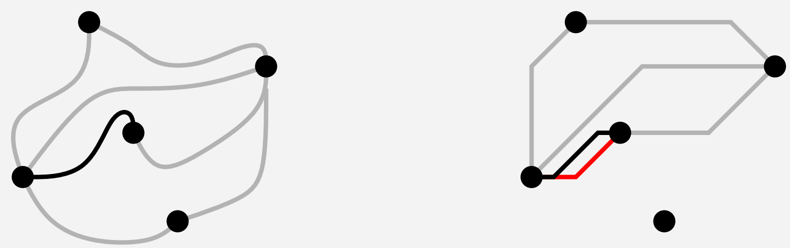


Idea behind everything

How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

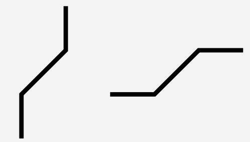
Allowed connections: 



Idea behind everything

How to construct the schematic map?

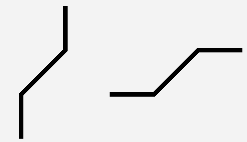
Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections: 

Idea behind everything

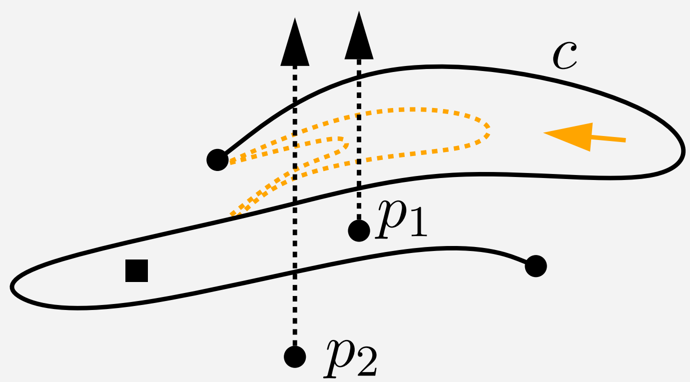
How to construct the schematic map?

Place the schematic connections in a top to bottom fashion, each connection topmost.

Allowed connections: 

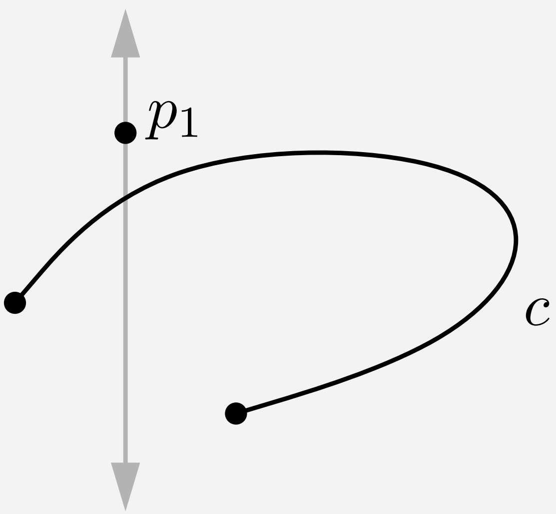
Points above and below a curve

Definition: Point p is below connection c if any continuous deformation in $\mathbb{R}^2 \setminus \{p\}$ of c intersects the vertical upper ray from p. Below is defined similarly.



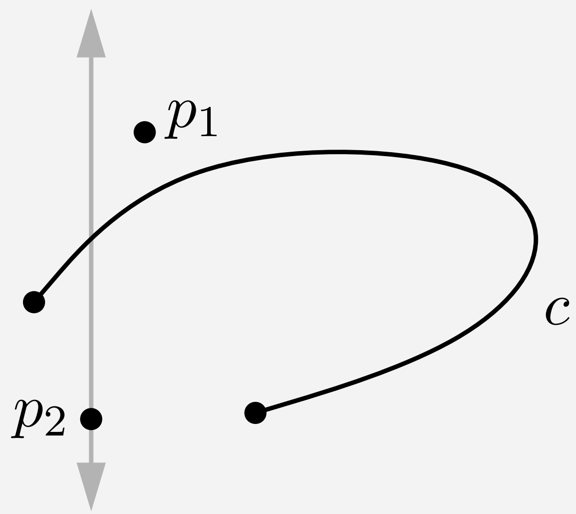
Remark: For comparing a point with a connection we don't worry of other points.

Example relations point-connection



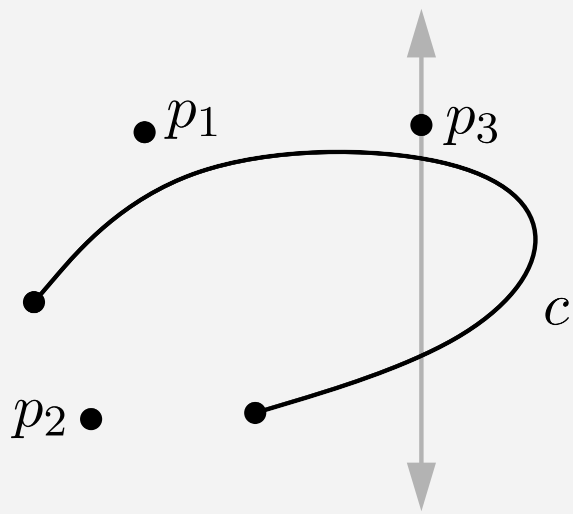
- p_1 is above c.

Example relations point-connection



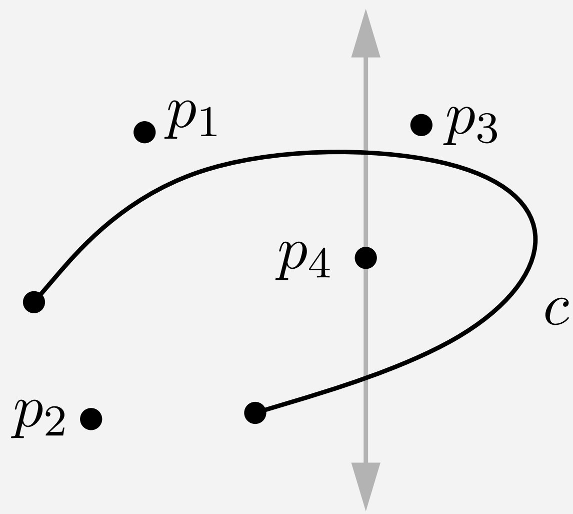
- p_1 is above c.
- p_2 is below c.

Example relations point-connection



- p_1 is above c.
- p_2 is below c.
- p_3 has no relation with c.

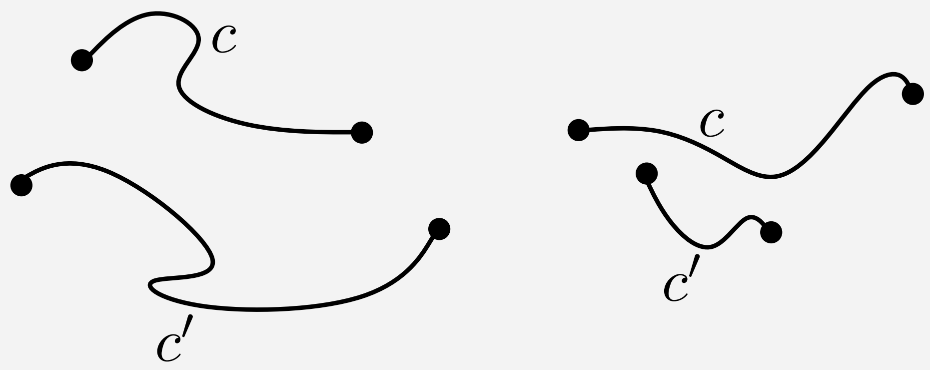
Example relations point-connection



- p_1 is above c.
- p_2 is below c.
- p_3 has no relation with c.
- p_4 is above and below c.

Relation connection-connection

Definition: Connection c is above connection c' if any endpoint of c is above c' or any endpoint of c' is below c



Property: This order is invariant by equivalence of maps.

The order of the input map is preserved in the schematic one.

This is the top-to-bottom order we follow to place connections, leaving always as much freedom as possible.

Our algorithm: (n: complexity of the map)

Input: Original map. Allowed types of connections.

Output: Schematic map, or 'It is not possible'.

Procedure:

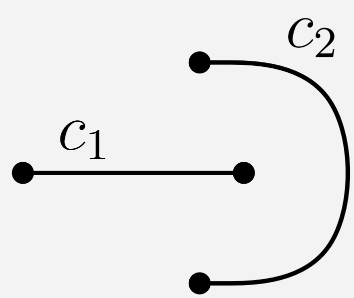
- compute order $\begin{cases} \text{monotone map} & \rightarrow O(n \log n) \\ \text{general map} & \rightarrow O(n \log^3 n) \end{cases}$
- place connections $O(n \log n)$

(Monotone map: each path c is monotone in some direction ϕ_c.)

Remark: 'It is not possible' means that there is no equivalent schematic map with the given types of connections.

When 'It is not possible'?

- We have a cyclic order.

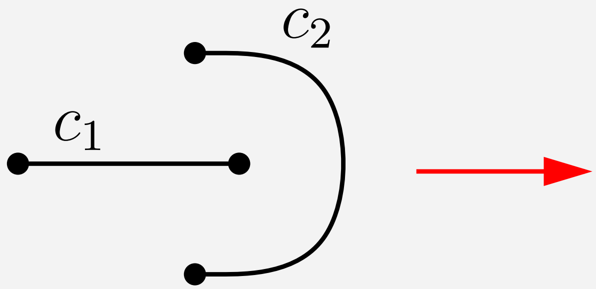


$$c_2 > c_1 > c_2$$

Which one do we place first?

When 'It is not possible'?

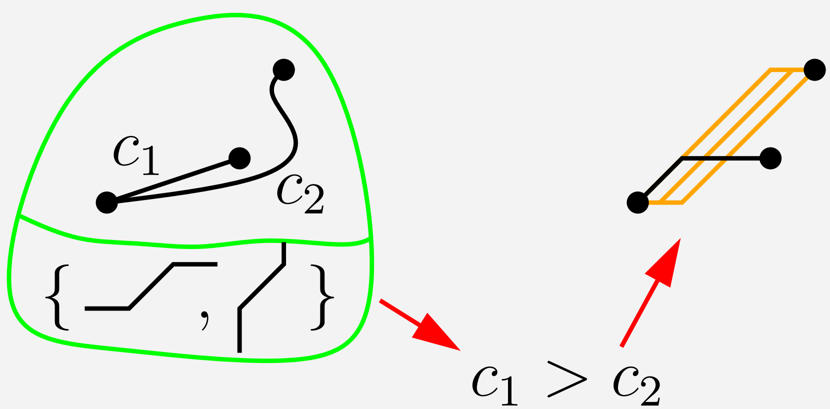
- We have a cyclic order.



$c_2 > c_1 > c_2$

Which one do we place first?

- We cannot place next connection.



Computing the order

Procedure:

1. compute an intermediate map $\tilde{M}$
   - monotone map $\rightarrow$ x-monotone map in $O(n \log n)$ time
   - general map $\rightarrow$ x-confined map in $O(n \log n)$ time
2. compute the order in $\tilde{M}$
   - x-monotone map $\rightarrow$ order in $O(n \log n)$ time
   - x-confined map $\rightarrow$ order in $O(n \log^3 n)$ time

(x-confined map: each path doesn't cross the vertical lines through its endpoints)

Compute order: reduce points

In a map M of complexity n: $|P_M|$, $|M| \in O(n)$.

Considering all points P_M for a path $c_i \in M$ takes $\Omega(|c_i| + |P_M|)$.

This has $\Omega(n^2)$ worst time for the whole map.

Compute order: reduce points

In a map M of complexity n: $|P_M|, |M| \in O(n)$.

Considering all points P_M for a path $c_i \in M$ takes $\Omega(|c_i| + |P_M|)$.

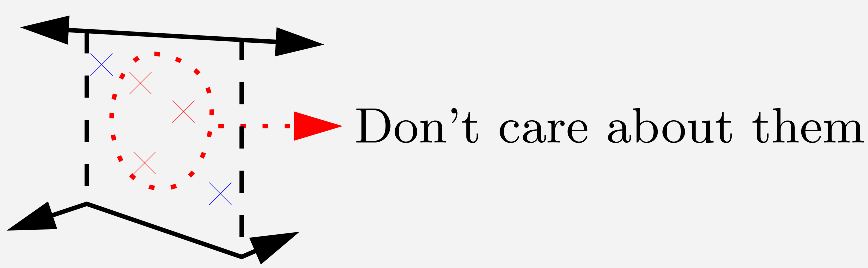
This has $\Omega(n^2)$ worst time for the whole map.

Consider $\mathcal{T}_{c_i}$, trapezoidal decomposition of c_i.

P_{c_i}: set of leftmost and rightmost points in $t \in \mathcal{T}_{c_i}$.

Let $\tilde{c}_i$ path with segments from $\mathcal{T}_{c_i}$. Then $\tilde{c}_i$ and c_i are equivalent in P_M iff equivalent in P_{c_i}.

Now, for a path c_i we only care about $O(c_i)$ points.

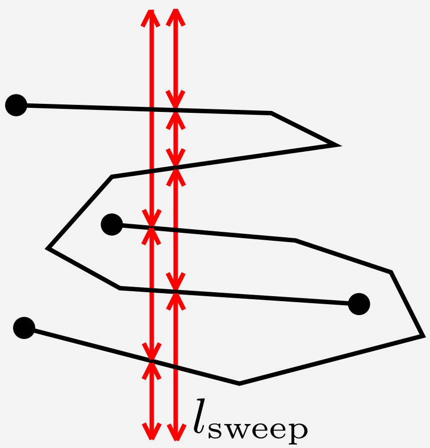


Reducing points

We want to compute P_{c_i}, $\forall c_i \in M$

Leftmost points with l-to-r sweep line simultaneously $\forall c_i \in M$.

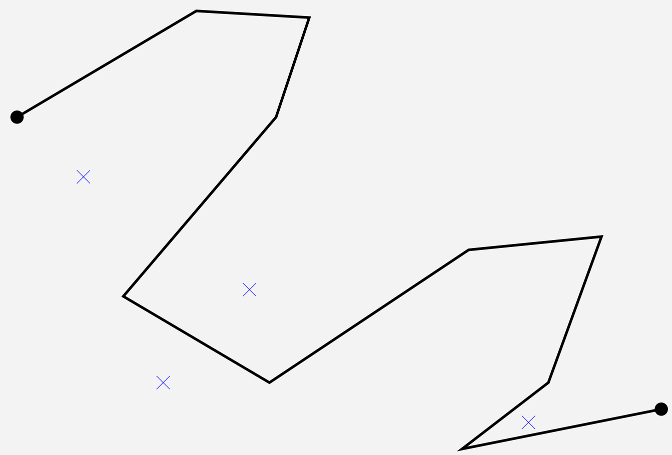
- Maintain the set $l_{\text{sweep}} \cap t$, $t \in \mathcal{T}_{c_i}$ $\forall c_i \in M$ in a tree.



- When find point in P_M report and delete all trapezoids containing it.
- $O(n)$ insertions and deletions $\rightarrow$ $O(n \log n)$

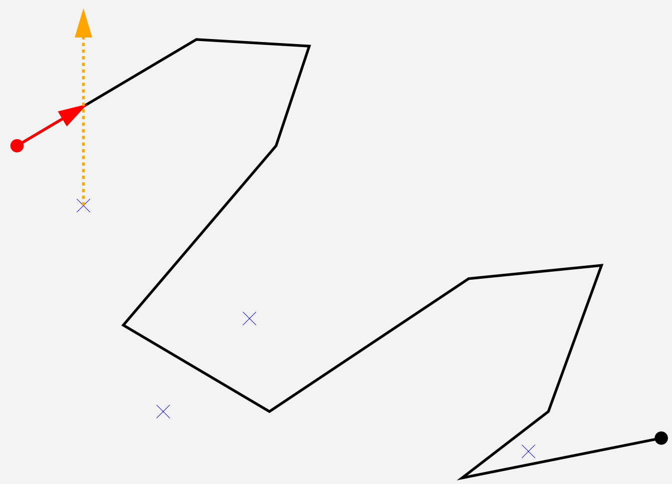
Making an intermediate x-monotone map

Compute for each $c_i \in M$ a monotone path $\tilde{c}_i \in \tilde{M}$:
incremental construction from left to right



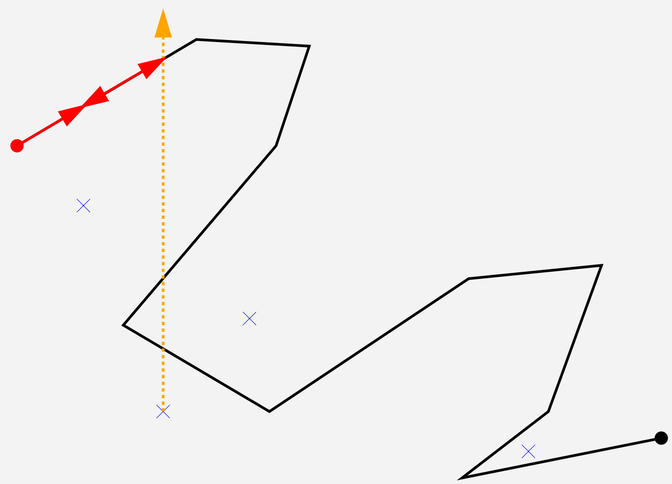
Making an intermediate x-monotone map

Compute for each $c_i \in M$ a monotone path $\tilde{c}_i \in \tilde{M}$:
incremental construction from left to right



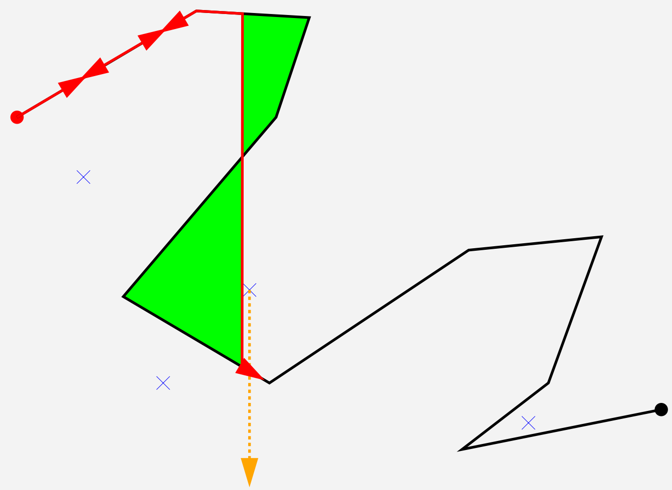
Making an intermediate x-monotone map

Compute for each $c_i \in M$ a monotone path $\tilde{c}_i \in \tilde{M}$:
incremental construction from left to right



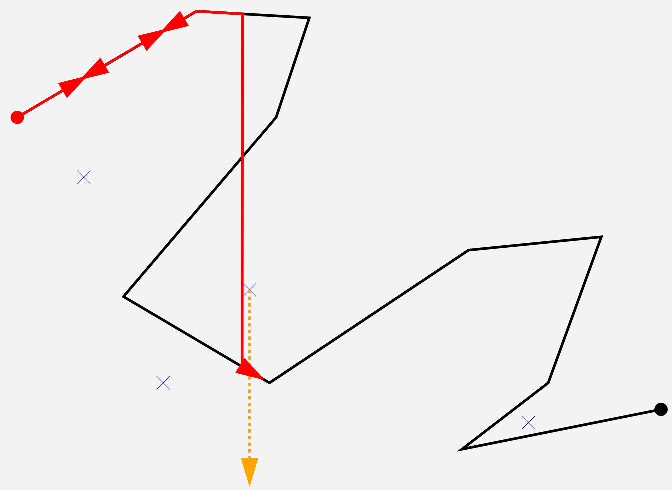
Making an intermediate x-monotone map

Compute for each $c_i \in M$ a monotone path $\tilde{c}_i \in \tilde{M}$:
incremental construction from left to right



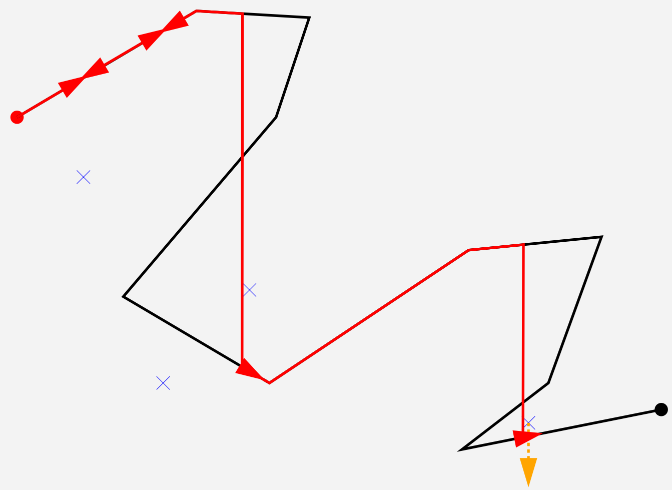
Making an intermediate x-monotone map

Compute for each $c_i \in M$ a monotone path $\tilde{c}_i \in \tilde{M}$:
incremental construction from left to right



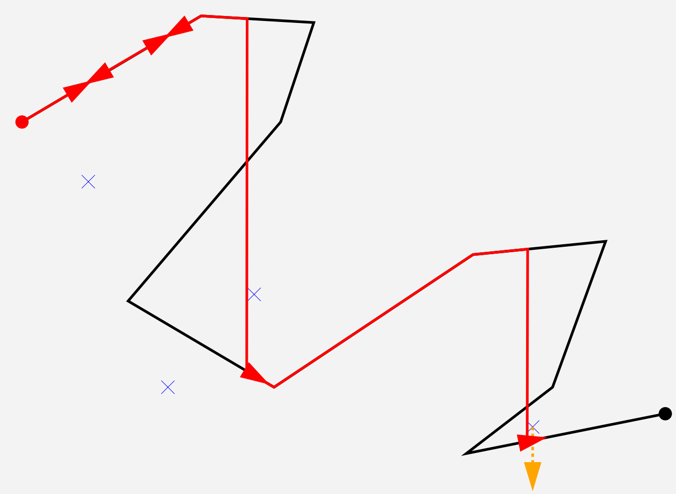
Making an intermediate x-monotone map

Compute for each $c_i \in M$ a monotone path $\tilde{c}_i \in \tilde{M}$:
incremental construction from left to right



Making an intermediate x-monotone map

Compute for each $c_i \in M$ a monotone path $\tilde{c}_i \in \tilde{M}$:
incremental construction from left to right

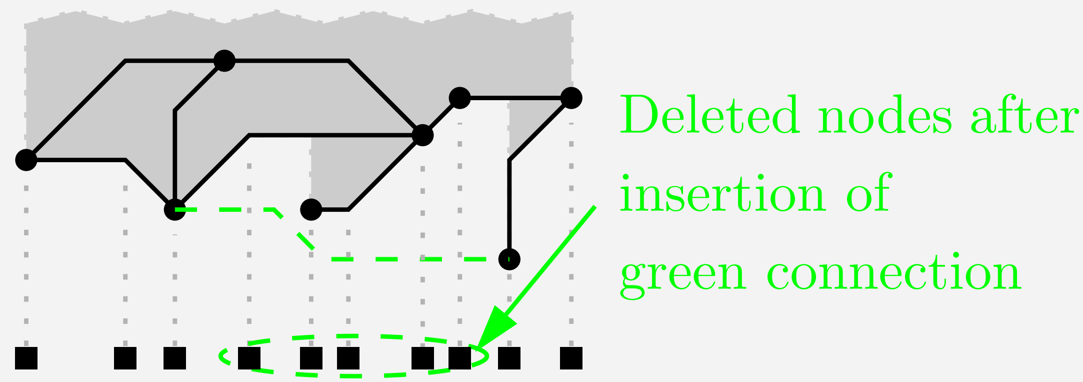


- The output of c_i using P_{c_i} or P_M is the same.
- For c_i, using P_{c_i}, it takes $O(|c_i| \log |c_i|)$ time.
- If c_i and c_j don't intersect, then $\tilde{c}_i$ and $\tilde{c}_j$ don't cross.

Placing connections

Incremental placement following the top to bottom order, maintaining the lower envelope (tree) of placed connections.

Each new placement must be topmost (it has to be defined).

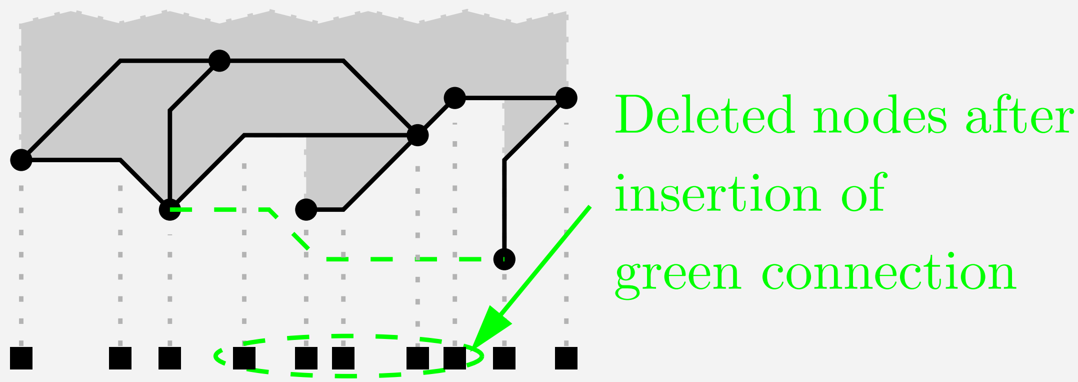


leaves of the tree ≡ lower envelope

Placing connections

Incremental placement following the top to bottom order, maintaining the lower envelope (tree) of placed connections.

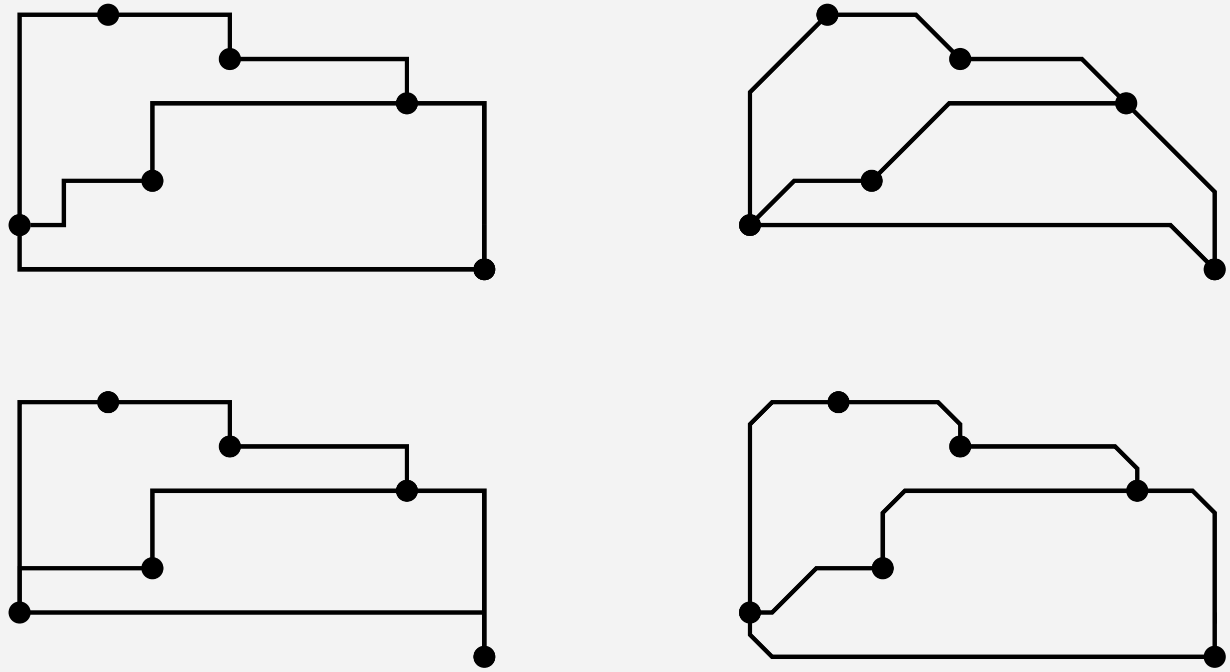
Each new placement must be topmost (it has to be defined).



leaves of the tree ≡ lower envelope

Result: Placement of n connections takes $O(n \log n)$ time.

How nice are the results?

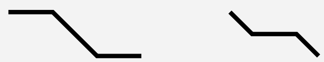


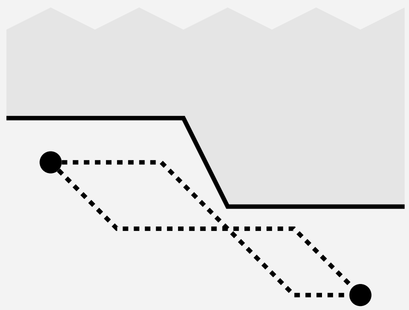
Work to do

- Compute order more quickly in general maps.

Work to do

- Compute order more quickly in general maps.
- Handle connections with topmost not defined.

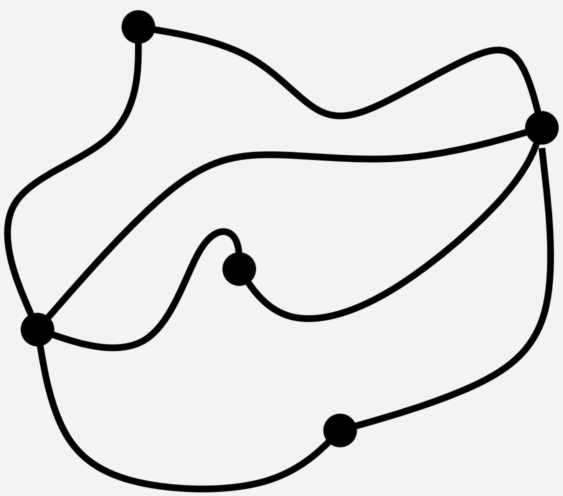


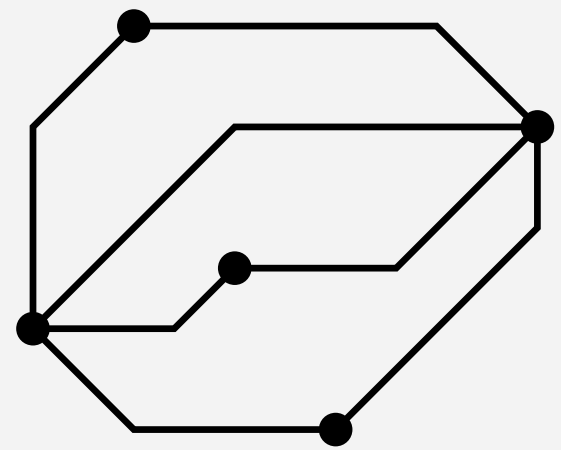


Work to do

- Compute order more quickly in general maps.
- Handle connections with topmost not defined.

- Experimental results.





Universiteit Utrecht

Sergio, sergio@cs.uu.nl