Testing homotopy for paths in the plane

Sergio Cabello — GIVE, UU

Yuanxin Liu
Andrea Mantler
Jack Snoeyink
} Dept. Comp. Sci., UNC

GIVE

Universiteit Utrecht

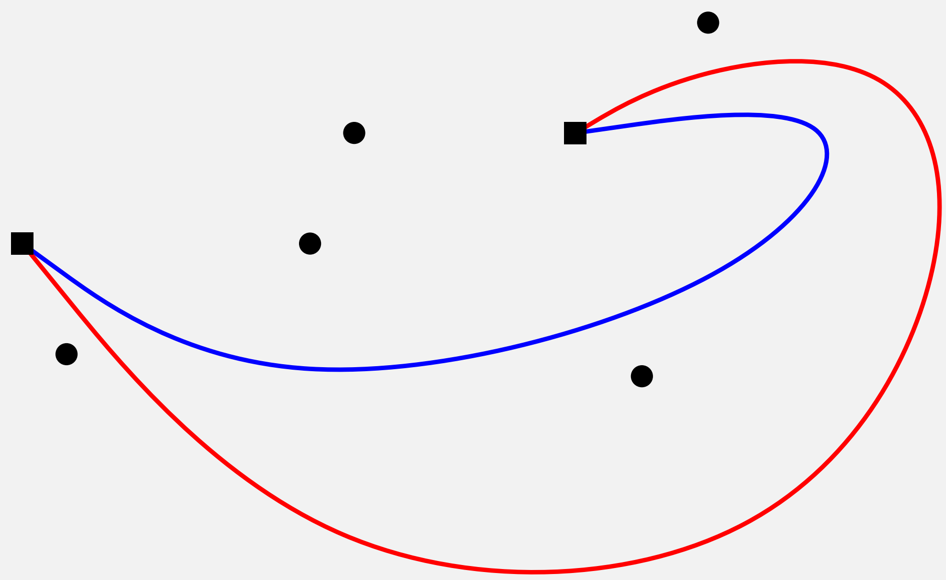
Contents

Homotopy problem

P a set of n points.

Paths $\alpha, \beta : [0, 1] \to \mathbb{R}^2 \setminus P$ of complexity n.

Can α be deformed into β without touching P?

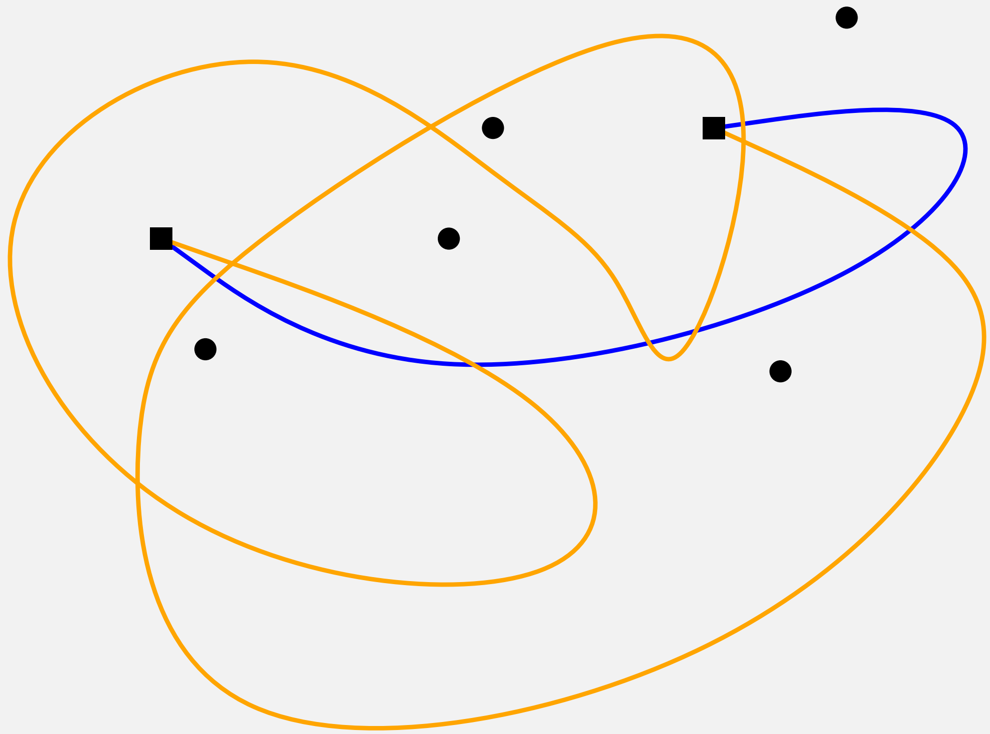


Homotopy problem

P a set of n points.

Paths $\alpha, \beta : [0,1] \to \mathbb{R}^2 \setminus P$ of complexity n.

Can α be deformed into β without touching P?

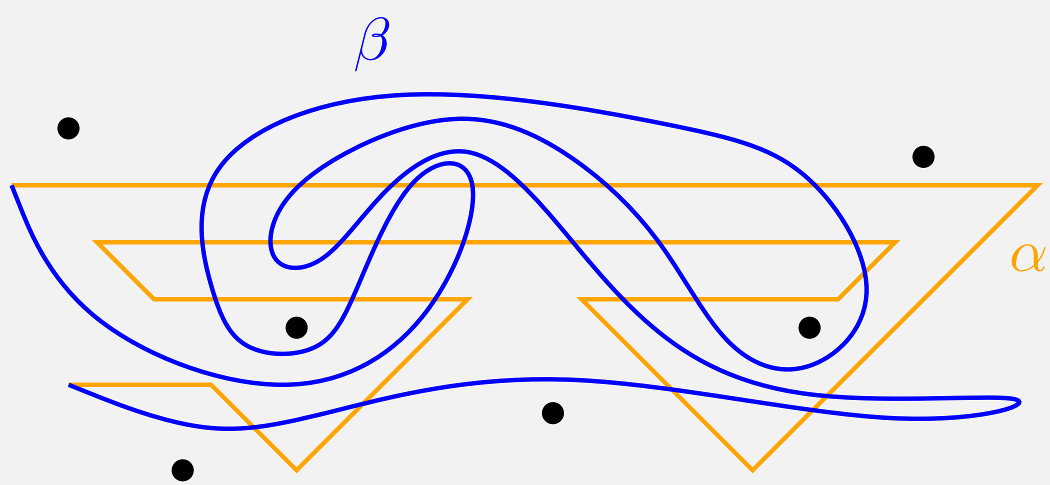


Homotopy problem

P a set of n points.

Paths $\alpha, \beta : [0,1] \to \mathbb{R}^2 \setminus P$ of complexity n.

Can α be deformed into β without touching P?

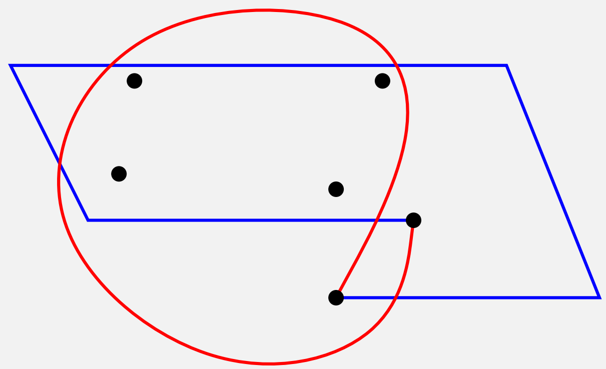


Theoretically solved in 1880.

What about the endpoints?

Are the endpoints of α, β obstacles?

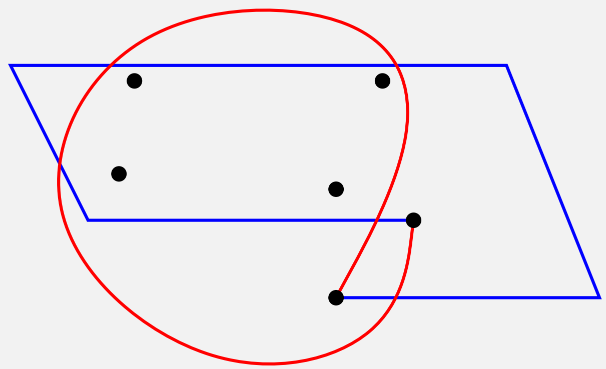
It makes a difference:



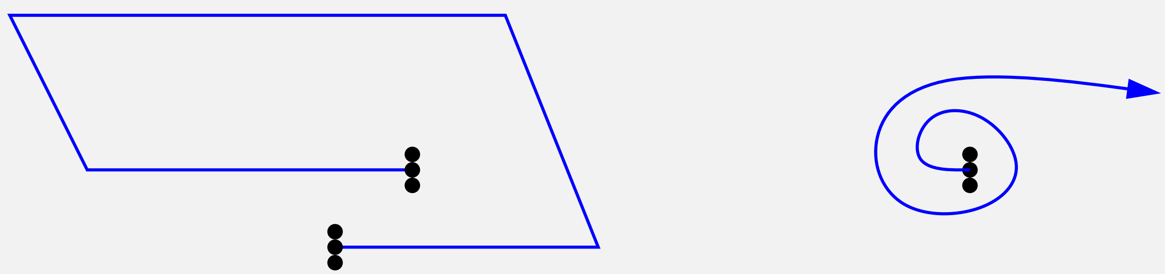
What about the endpoints?

Are the endpoints of α, β obstacles?

It makes a difference:



It is more general to assume they are not obstacles.



Applications

- Continuous homotopic routing problem.
- Geographic information systems.
- Motion path planning.

Related work

Similar problem and techniques

Shortest homotopic path problem [HS'94,EKL'02,B'03]

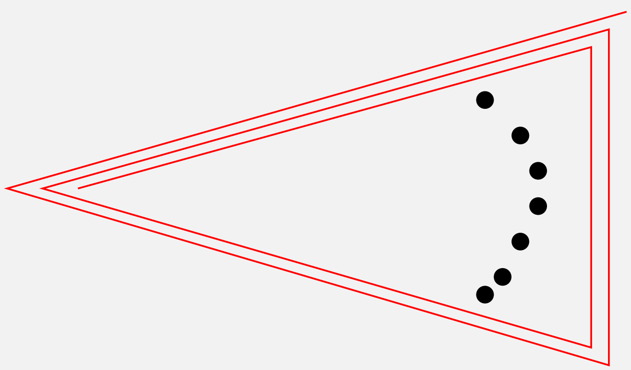
Related work

Similar problem and techniques

Shortest homotopic path problem [HS'94,EKL'02,B'03]

It can be used to test homotopy:

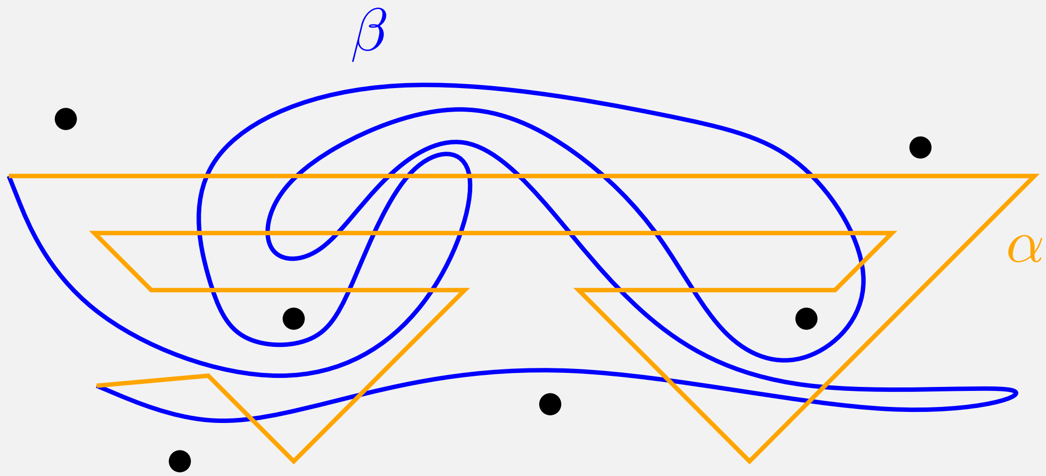
- reduce each path to the shortest path that is homotopic
- compare them
- the problem is that some shortest path have $\Omega(n^2)$ segments.



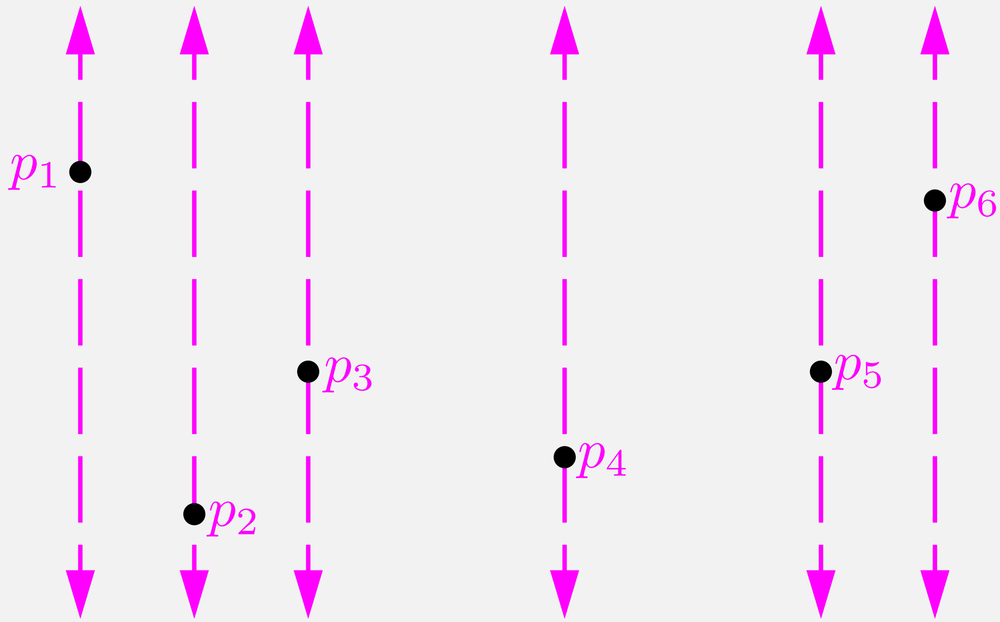
Testing homotopy for simple paths

- Topological oracle.
- Algorithmical issues.
- Mini-example.
- Lower bound.

Topological oracle (Simple)

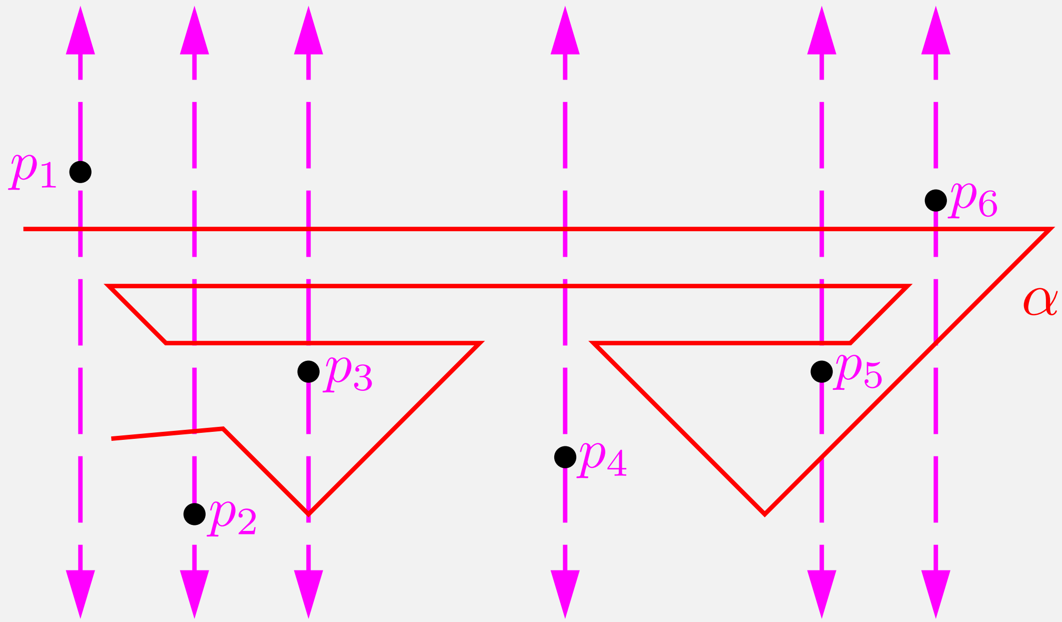


Topological oracle (Simple)



Vertical rays upwards and downwards.

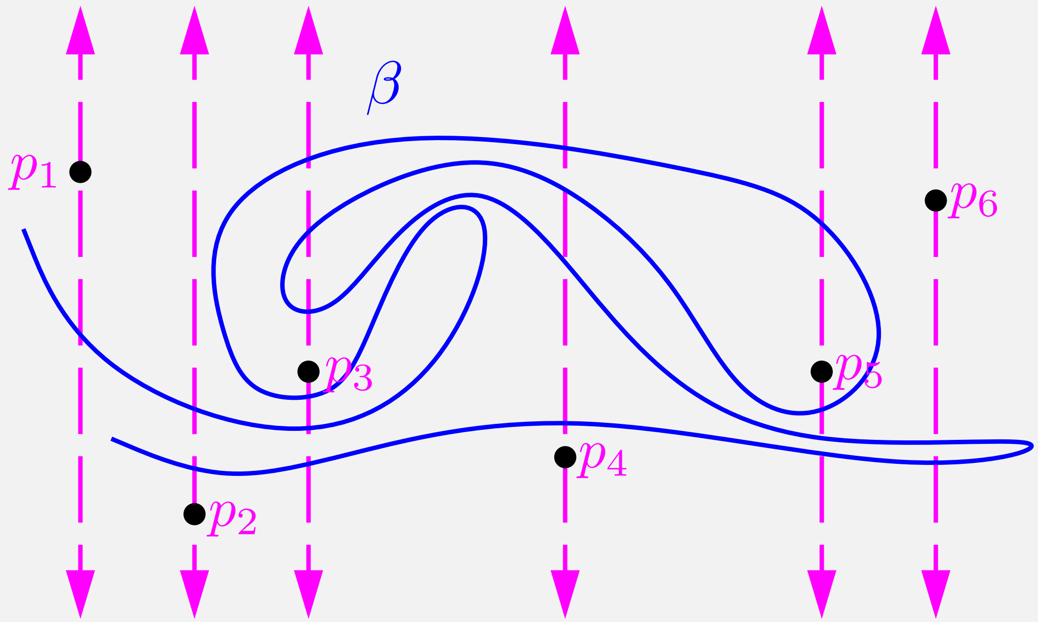
Topological oracle (Simple)



Vertical rays upwards and downwards.

$\alpha \equiv \underline{p_1}\overline{p_2 p_3 p_4 p_5}\underline{p_6 p_6 p_5}\overline{p_5 p_5 p_4 p_3 p_2 p_2 p_3}\underline{p_3}\overline{p_2}.$

Topological oracle (Simple)

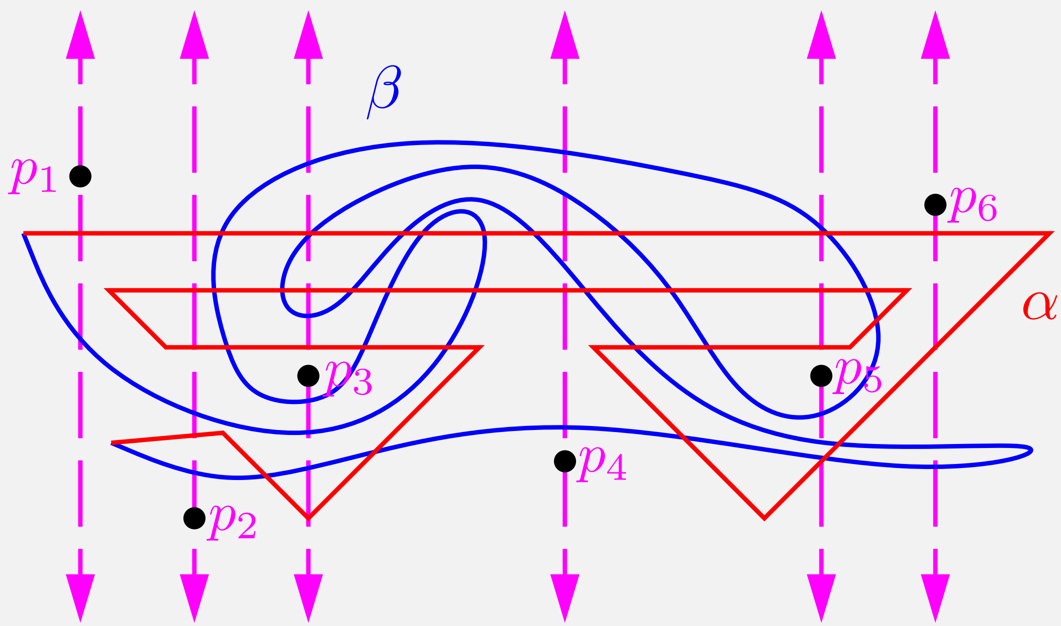


Vertical rays upwards and downwards.

$\alpha \equiv \underline{p_1}\overline{p_2p_3p_4p_5}\underline{p_6p_6p_5}\overline{p_5p_5p_4p_3p_2p_2p_3}\underline{p_3}\overline{p_2}.$

$\beta \equiv \underline{p_1}\overline{p_2}\underline{p_3p_3}\overline{p_3p_4p_5}\underline{p_5}\overline{p_4p_3p_3p_4}\underline{p_5p_6p_6p_5}\overline{p_4}\underline{p_3}\overline{p_2}.$

Topological oracle (Simple)

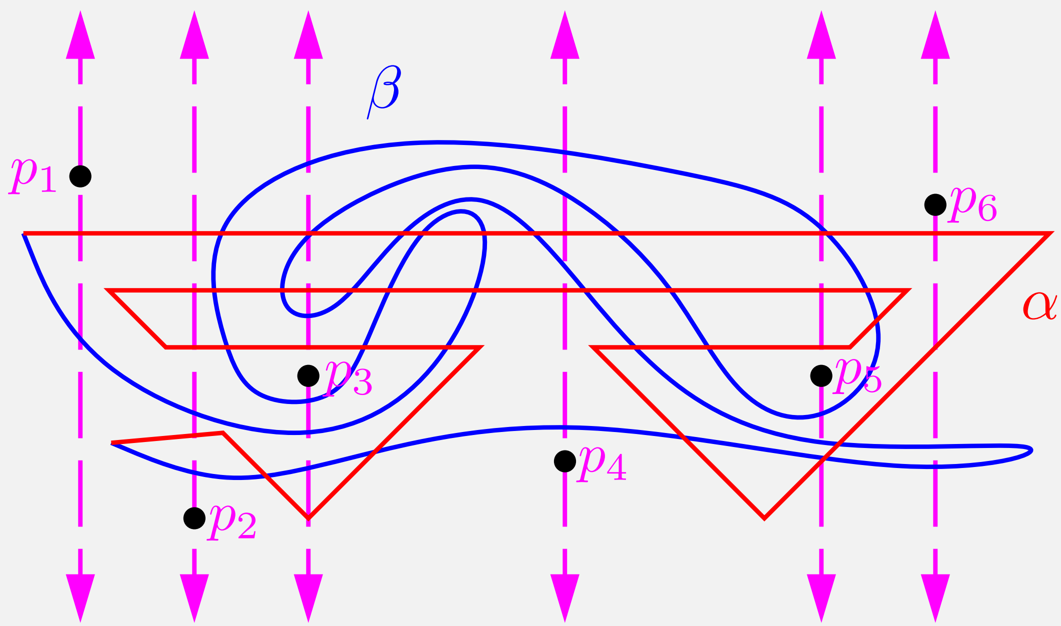


Vertical rays upwards and downwards.

$\alpha \equiv \underline{p_1}\overline{p_2p_3p_4p_5}\underline{p_6p_6p_5}\overline{p_5p_5p_4p_3p_2p_2p_3}\underline{p_3}\overline{p_2}.$

$\beta \equiv \underline{p_1}\overline{p_2}\underline{p_3p_3}\overline{p_3p_4p_5}\underline{p_5}\overline{p_4p_3p_3p_4}\underline{p_5p_6p_6p_5}\overline{p_4}\underline{p_3}\overline{p_2}.$

Topological oracle (Simple)



Vertical rays upwards and downwards.

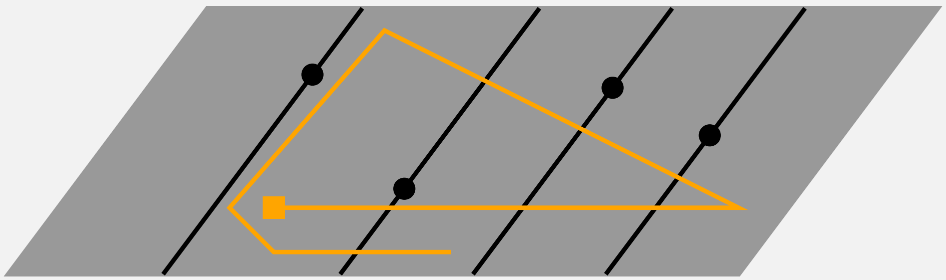
$\alpha \equiv \underline{p_1}\overline{p_2p_3p_4p_5}\underline{p_6p_6p_5}\overline{p_5p_5p_4p_3p_2p_2p_3}\underline{p_3}\overline{p_2} \equiv \underline{p_1}\overline{p_2p_3p_4p_5}\underline{p_5}\overline{p_4}\underline{p_3}\overline{p_2}$.

$\beta \equiv \underline{p_1}\overline{p_2}\underline{p_3p_3}\overline{p_3p_4p_5}\underline{p_5}\overline{p_4p_3p_3p_4}\underline{p_5p_6p_6p_5}\overline{p_4}\underline{p_3}\overline{p_2} \equiv \underline{p_1}\overline{p_2p_3p_4p_5}\underline{p_5}\overline{p_4}\underline{p_3}\overline{p_2}$.

α, β homotopic iff same canonical sequence.

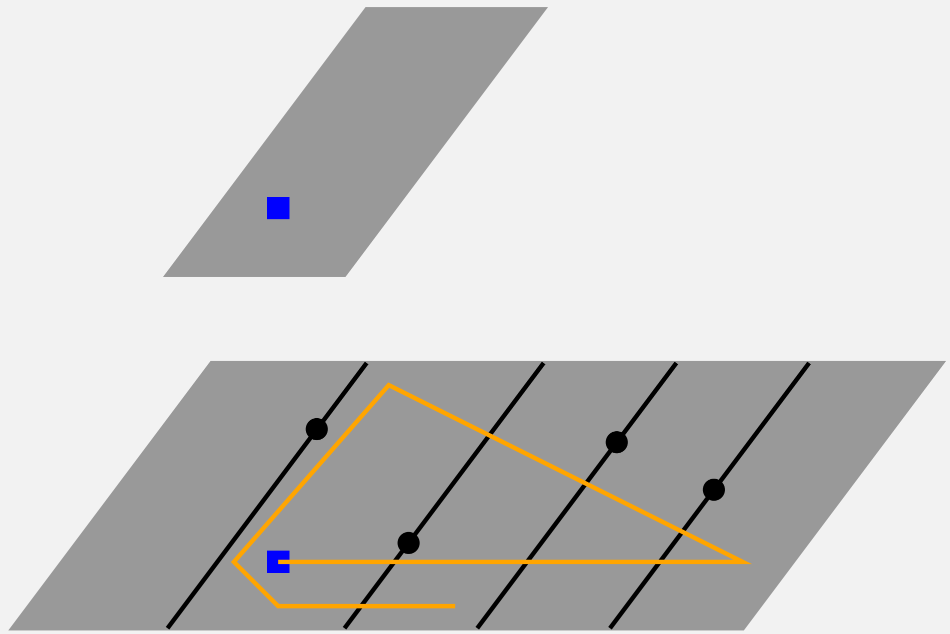
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



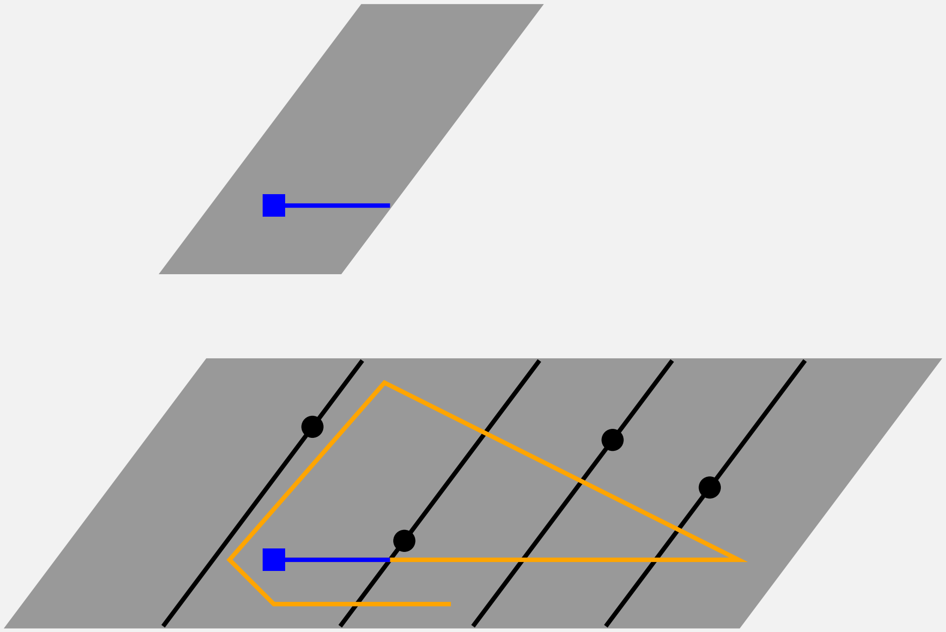
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



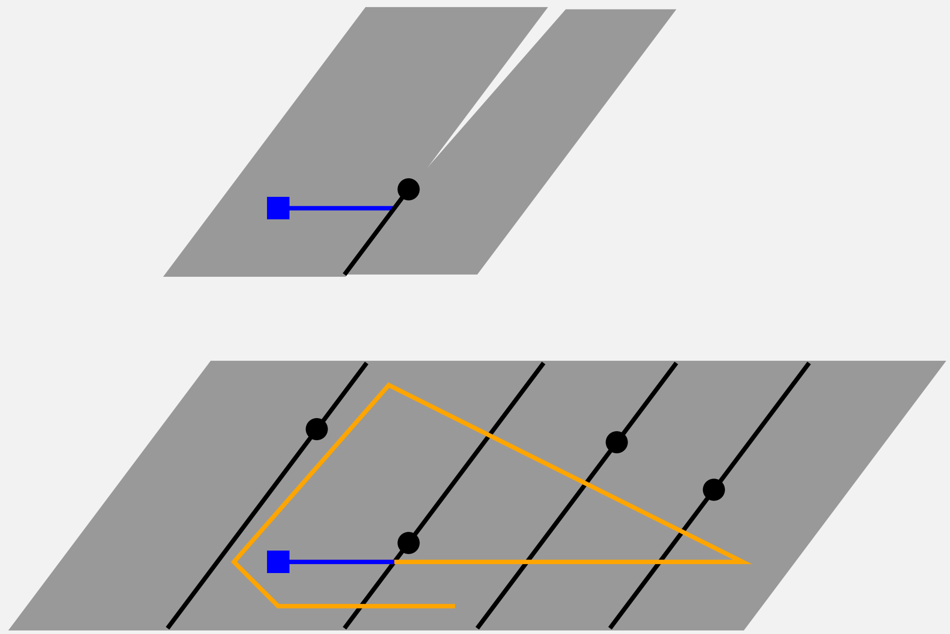
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



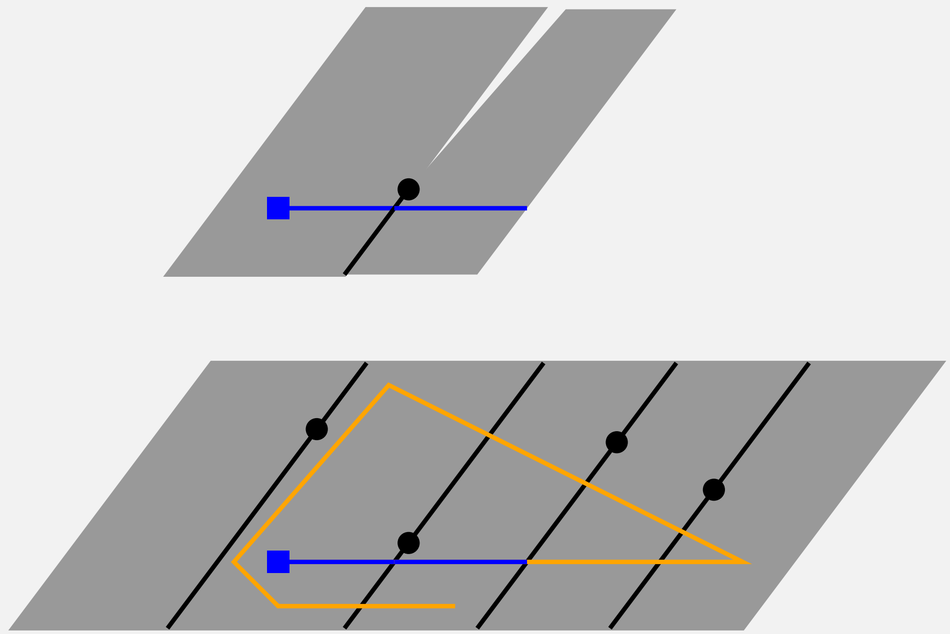
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



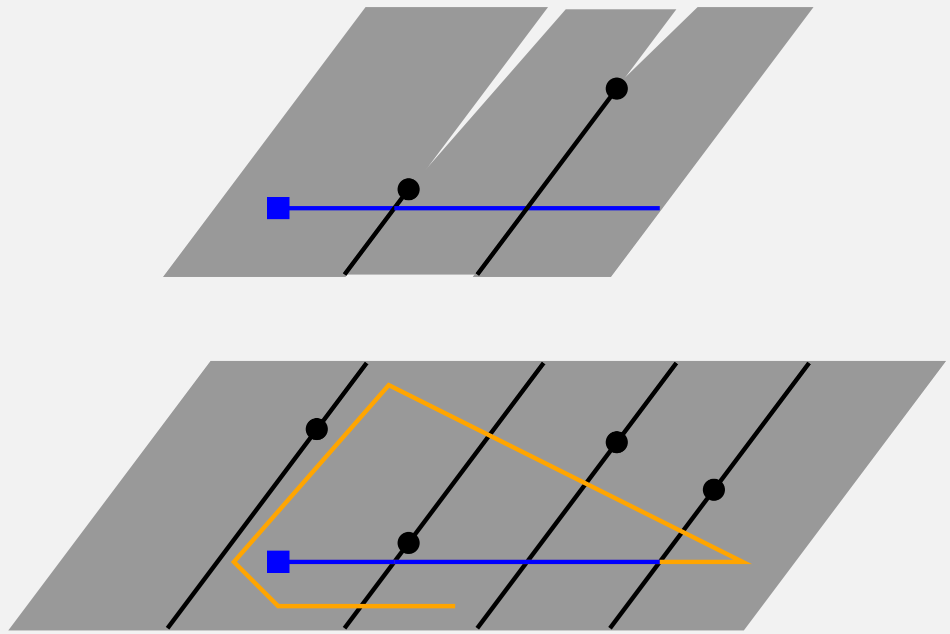
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



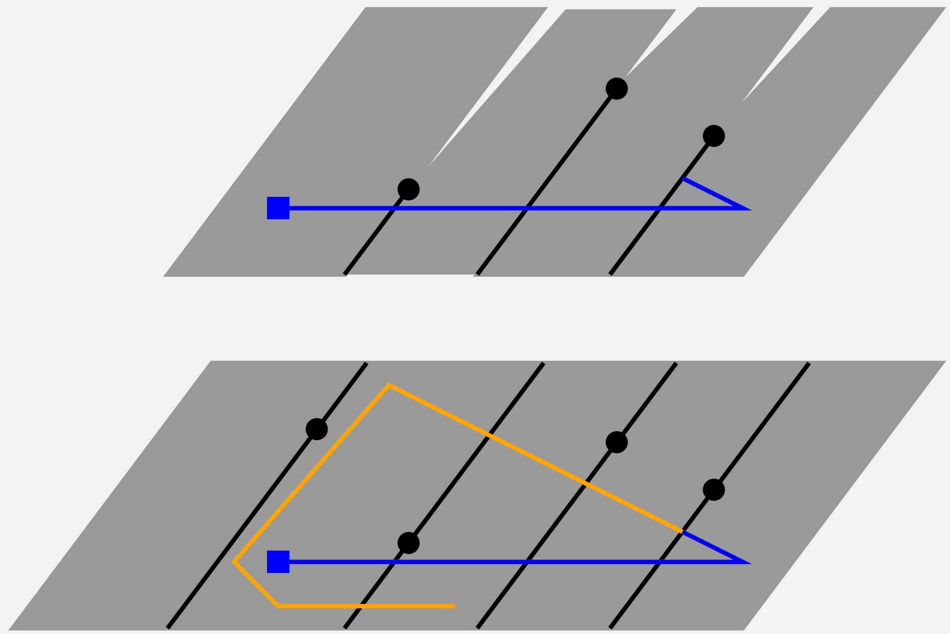
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



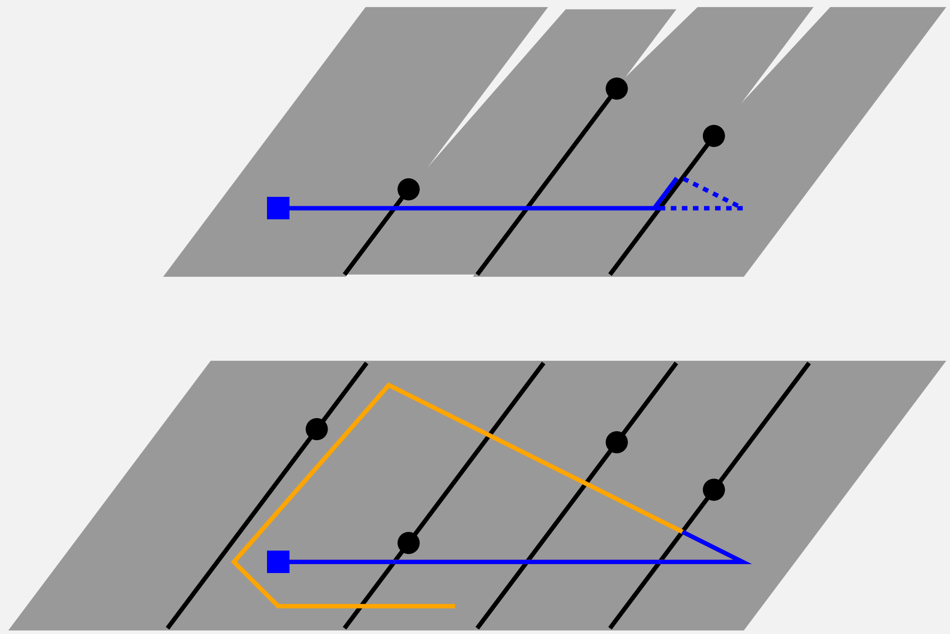
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



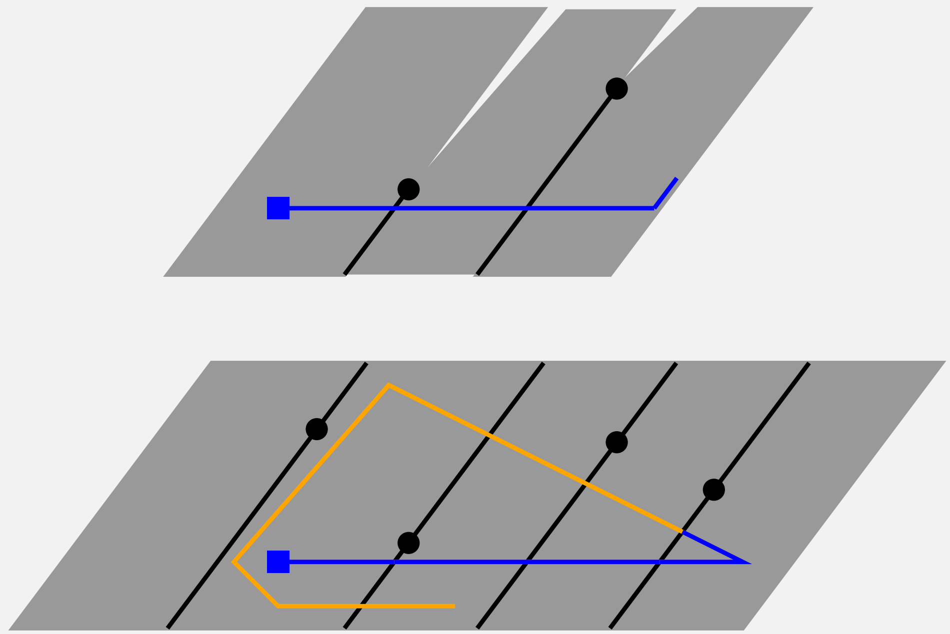
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



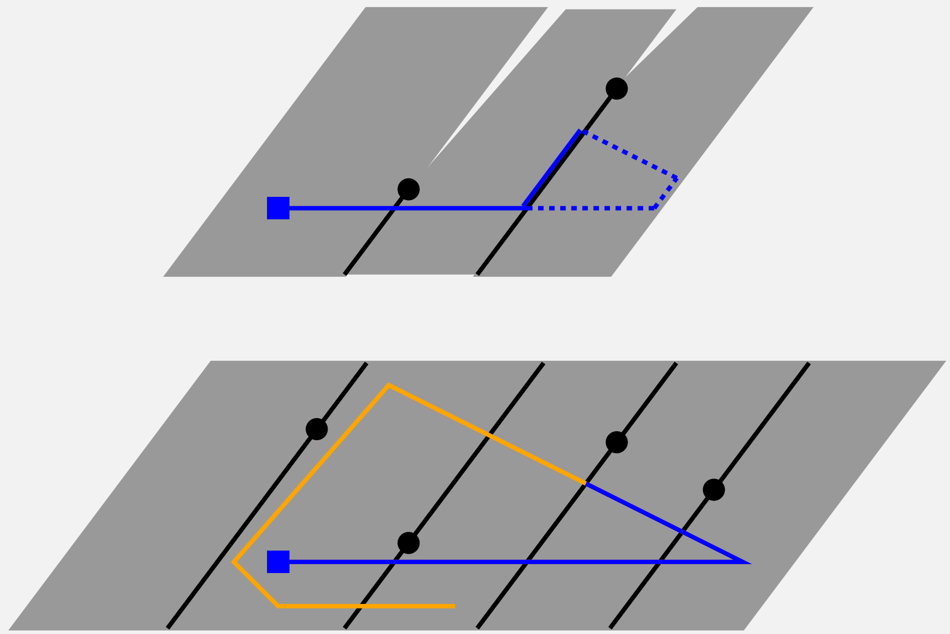
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



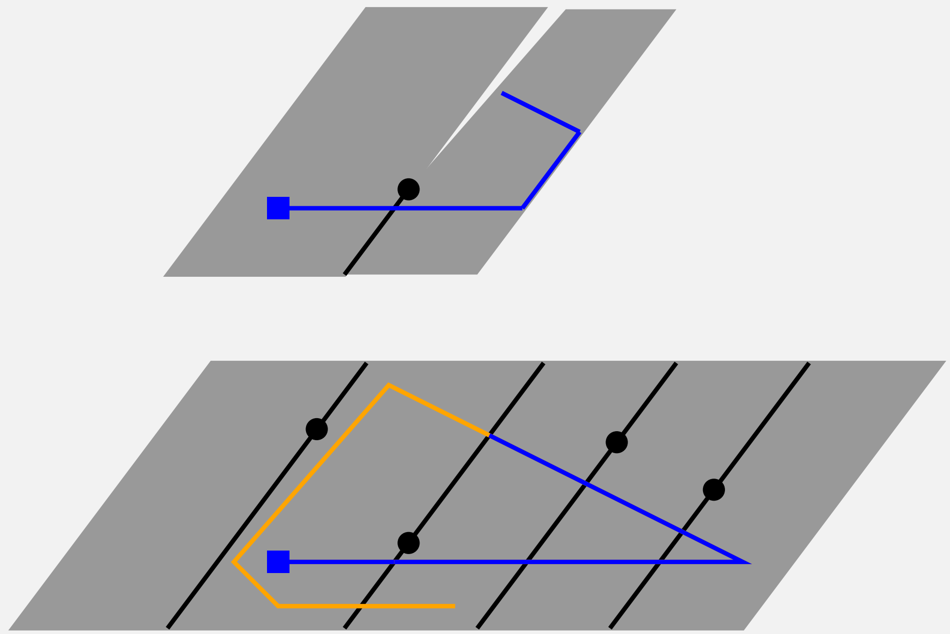
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



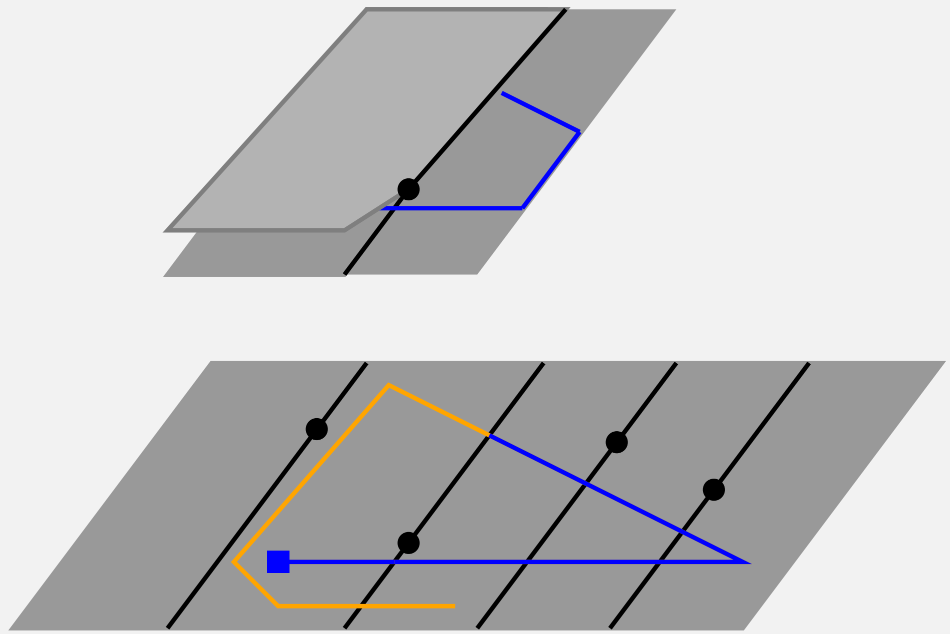
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



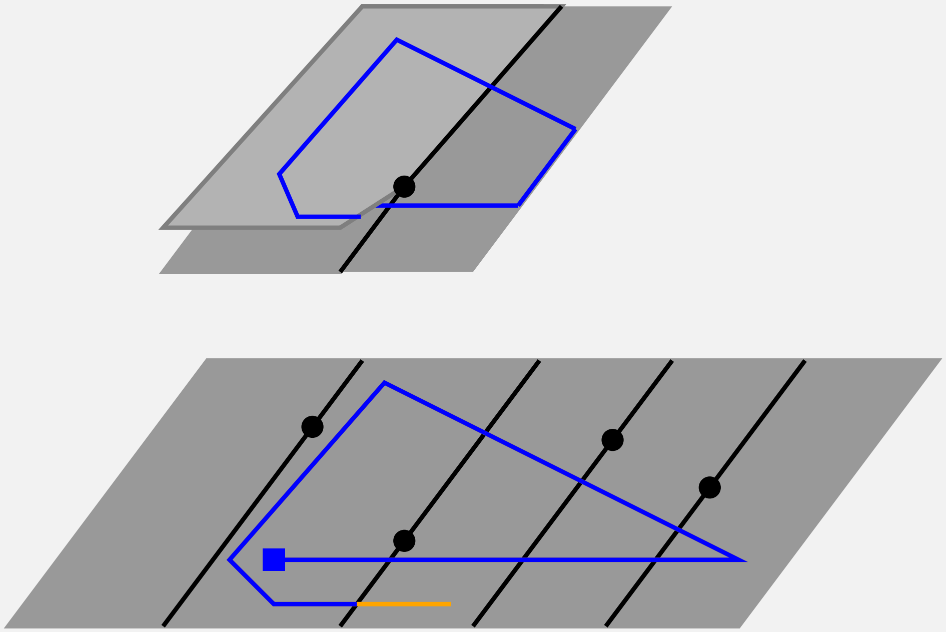
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



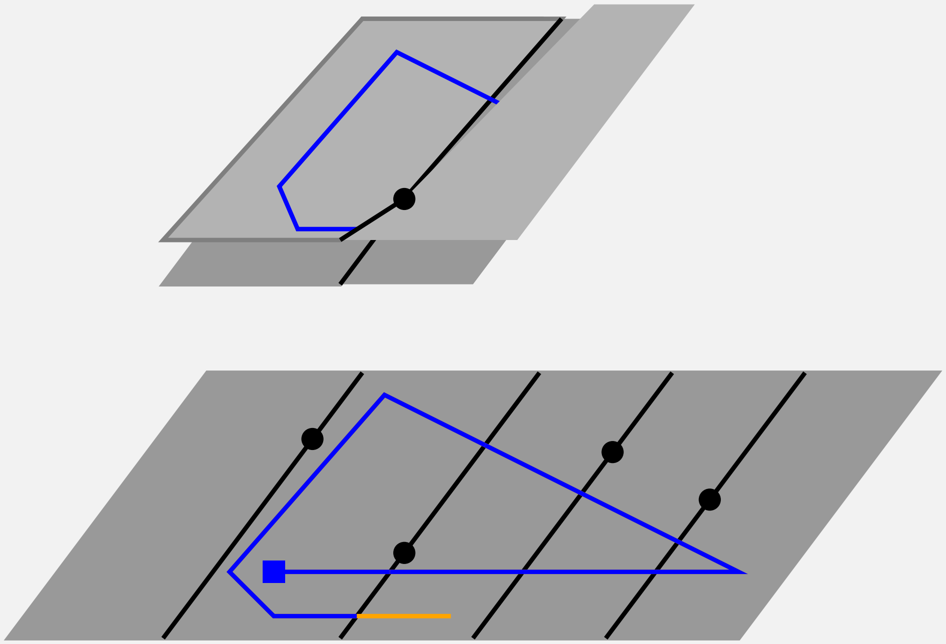
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



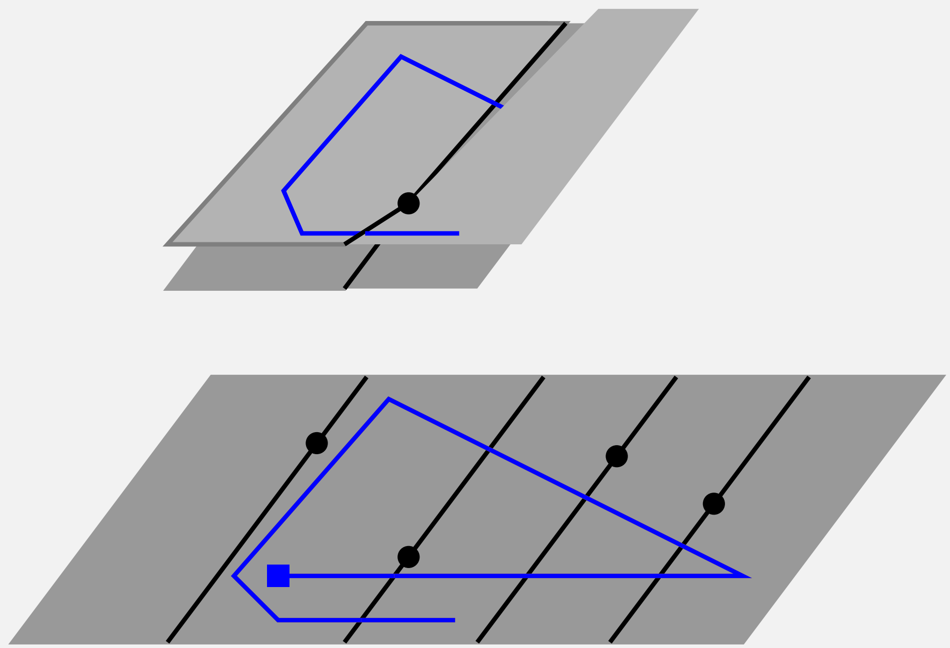
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



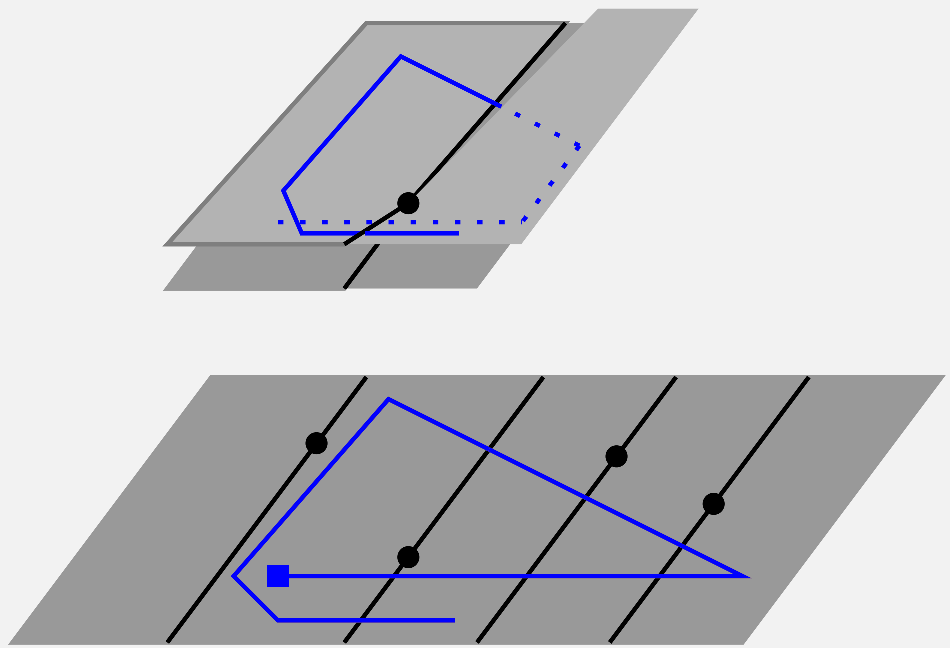
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



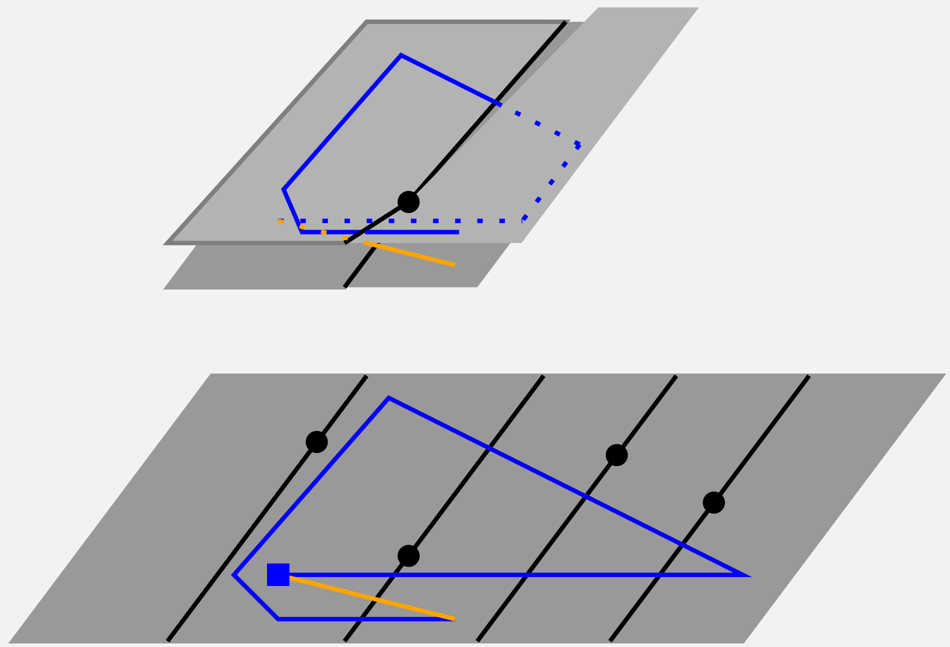
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



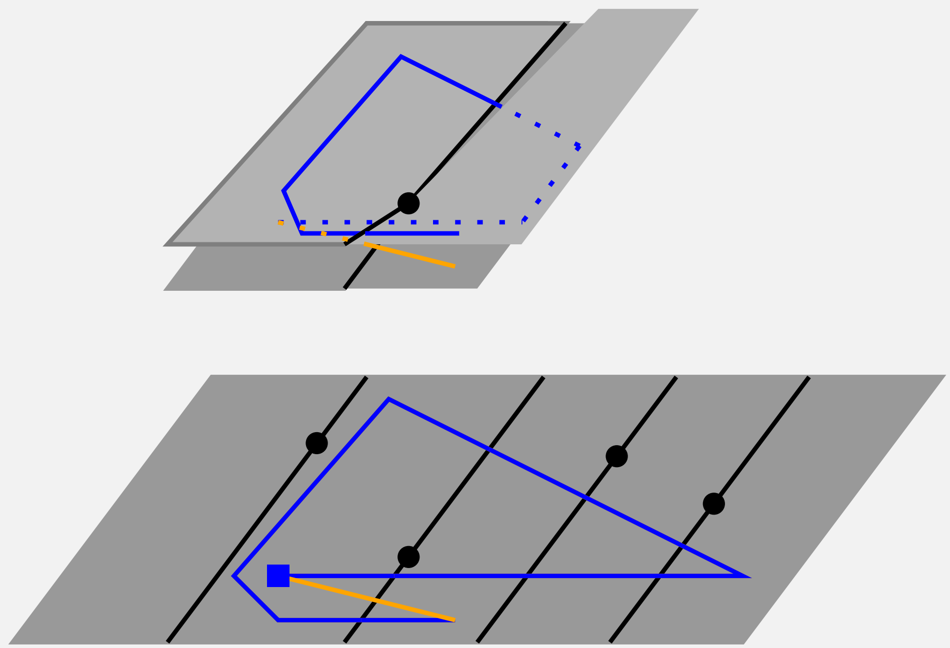
Topological oracle. Why? (Simple)

Lifting the path to the universal cover.



Topological oracle. Why? (Simple)

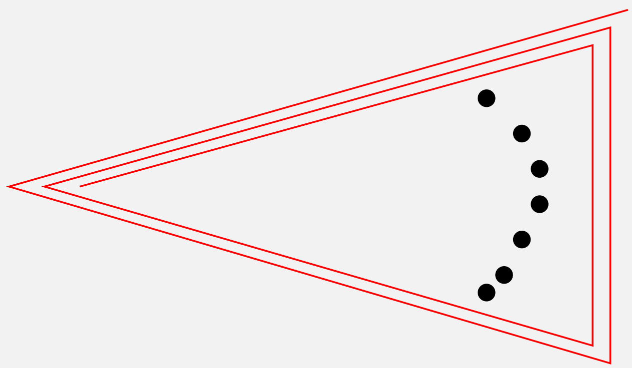
Lifting the path to the universal cover.



Same canonical sequences ⇔ Same endpoints in the lifts ⇔ Homotopic

Algorithmical issues (Simple)

The canonical sequence can have $\theta(n^2)$ symbols.



Our approach

Compare $can(\alpha)$ and $can(\beta)$ without computing them

ALGORITHMICAL ISSUES (SIMPLE)

Theorem: We can decide in $O(n \log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

Proof:

1. convert the problem to an orthogonal one
2. path α' whose $seq(\alpha') = can(\alpha)$
3. path β' whose $seq(\beta') = can(\beta)$
4. compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$

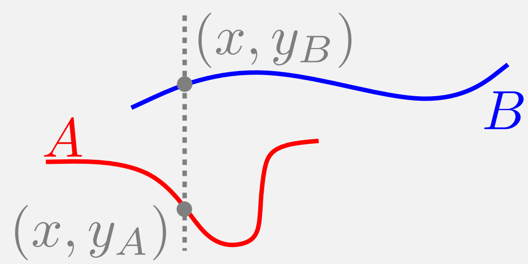
Algorithmical issues (Simple)

(convert the problem to an orthogonal one)

- split α into monotone chains: $\alpha_1, \ldots, \alpha_m$

- order among $\{P, \alpha_1, \ldots, \alpha_m\}$:

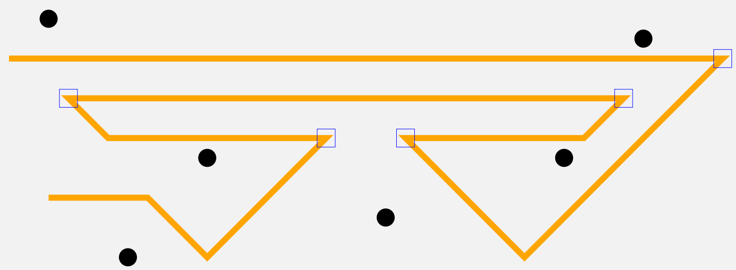
$$A < B \;\Leftrightarrow\; \exists (x, y_A) \in A, (x, y_B) \in B,\; y_A < y_B$$



- replace all y-cordinate by the ranking

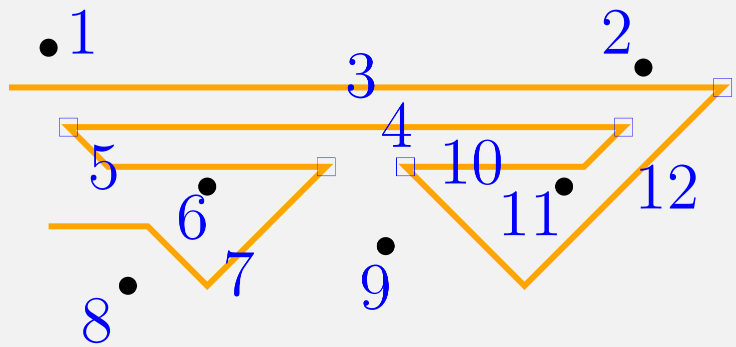
Algorithmical issues (Simple)

(convert the problem to an orthogonal one)



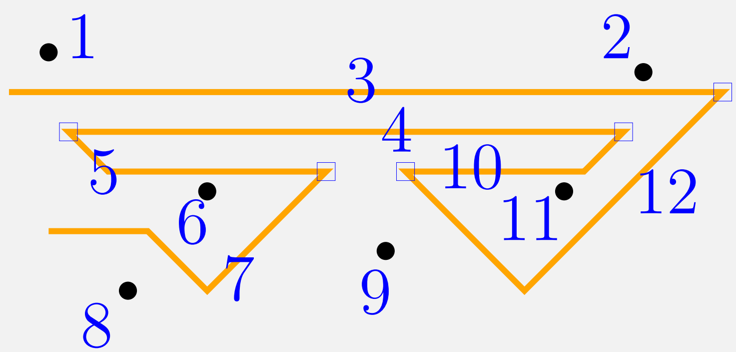
Algorithmical issues (Simple)

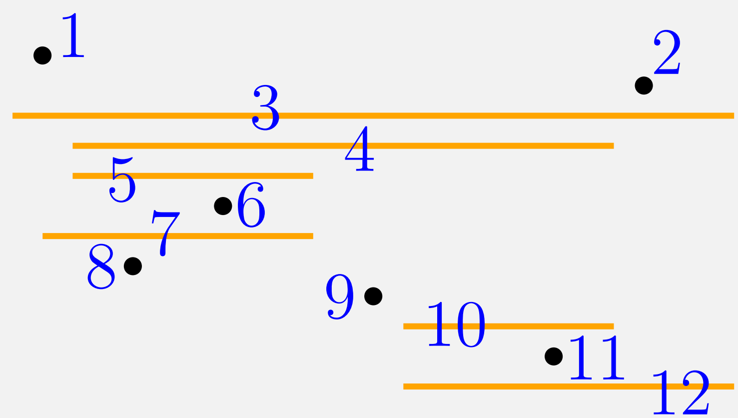
(convert the problem to an orthogonal one)



Algorithmical issues (Simple)

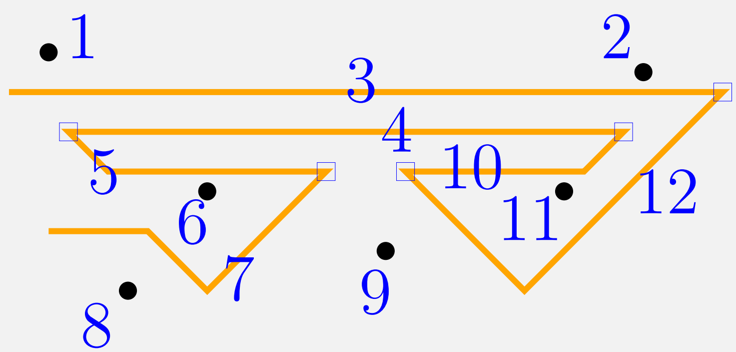
(convert the problem to an orthogonal one)

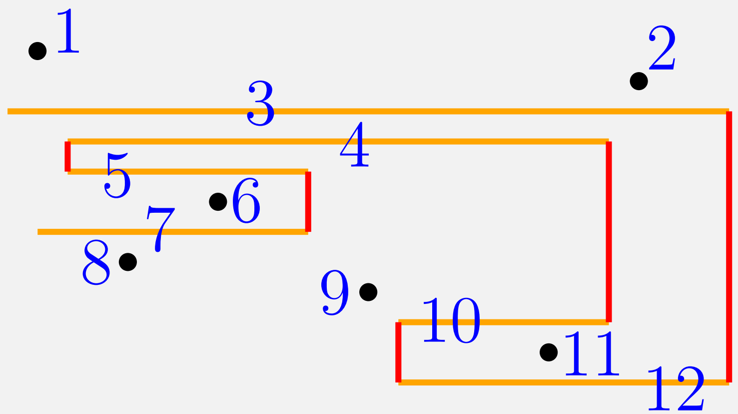




Algorithmical issues (Simple)

(convert the problem to an orthogonal one)

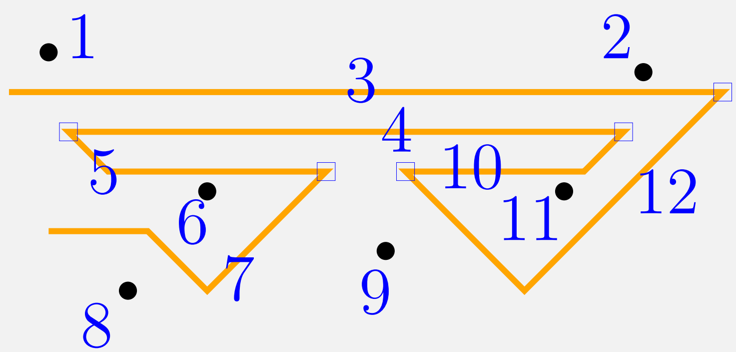


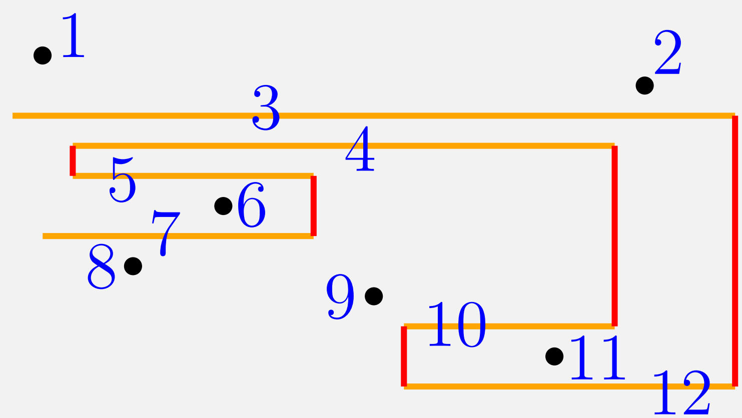


The (canonical) sequence of the path is kept!

Algorithmical issues (Simple)

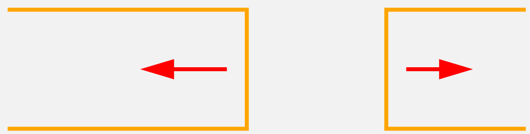
(convert the problem to an orthogonal one)





The (canonical) sequence of the path is kept!

What else do we gain? We can perform three-sided range querys:



$O(n \log n)$ preprocessing, $O(\log n)$ per query.

ALGORITHMICAL ISSUES (SIMPLE)

Theorem: We can decide in $O(n \log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

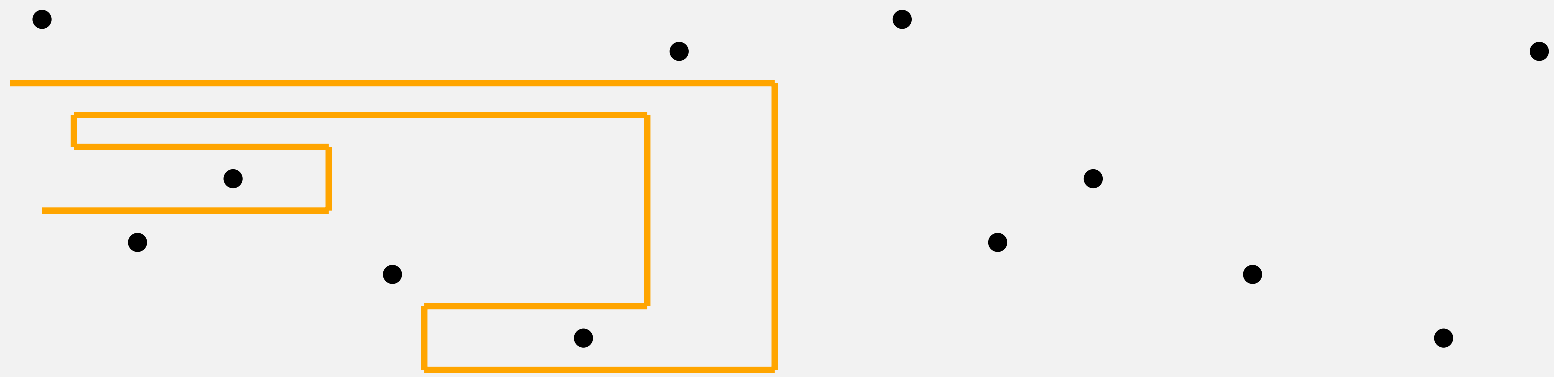
Proof:

1. convert the problem to an orthogonal one (done)
2. path α' whose $seq(\alpha') = can(\alpha)$
3. path β' whose $seq(\beta') = can(\beta)$
4. compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$

Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

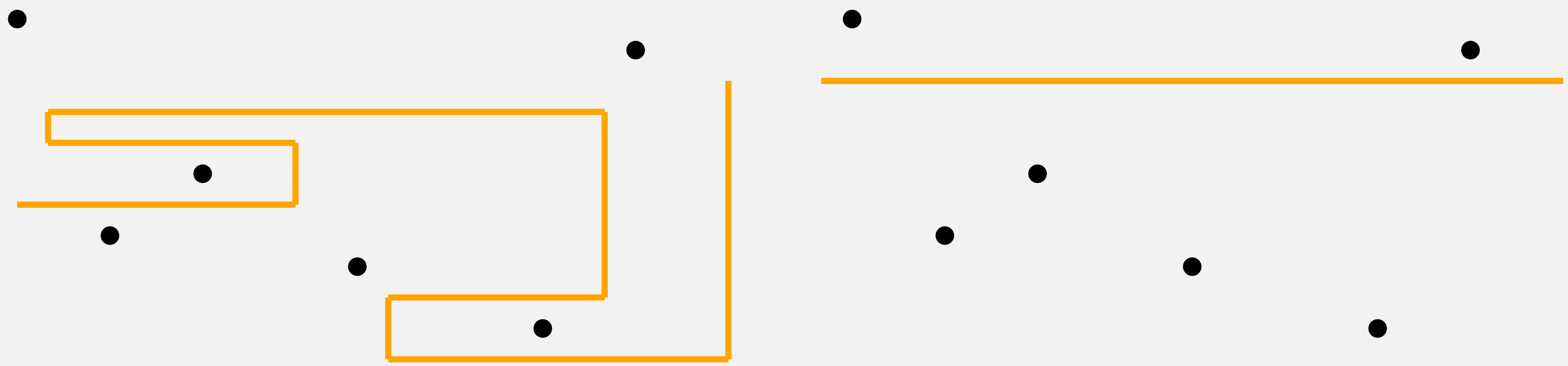
incrementally, following the orthogonal version of α



Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

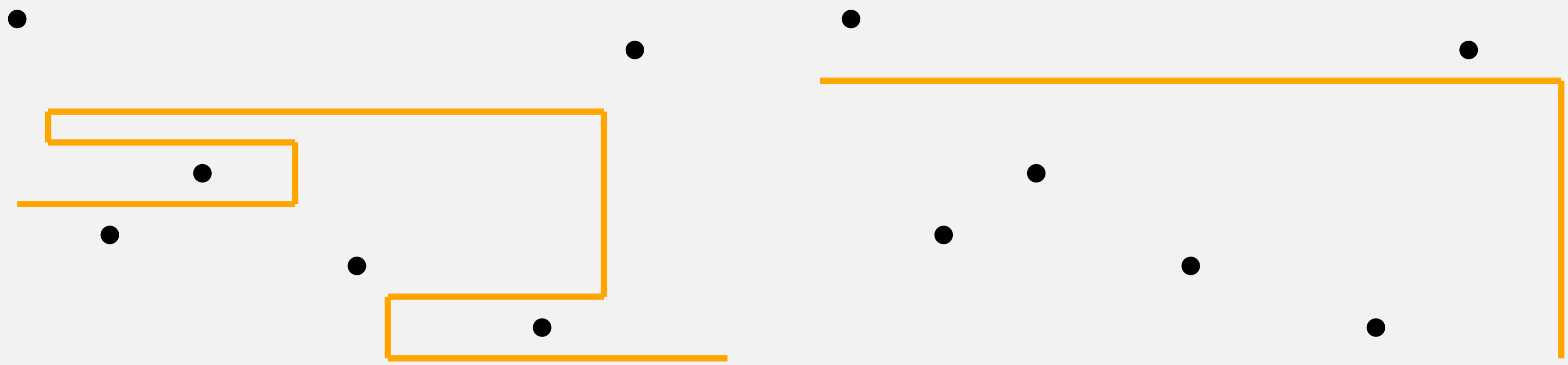
incrementally, following the orthogonal version of α



Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

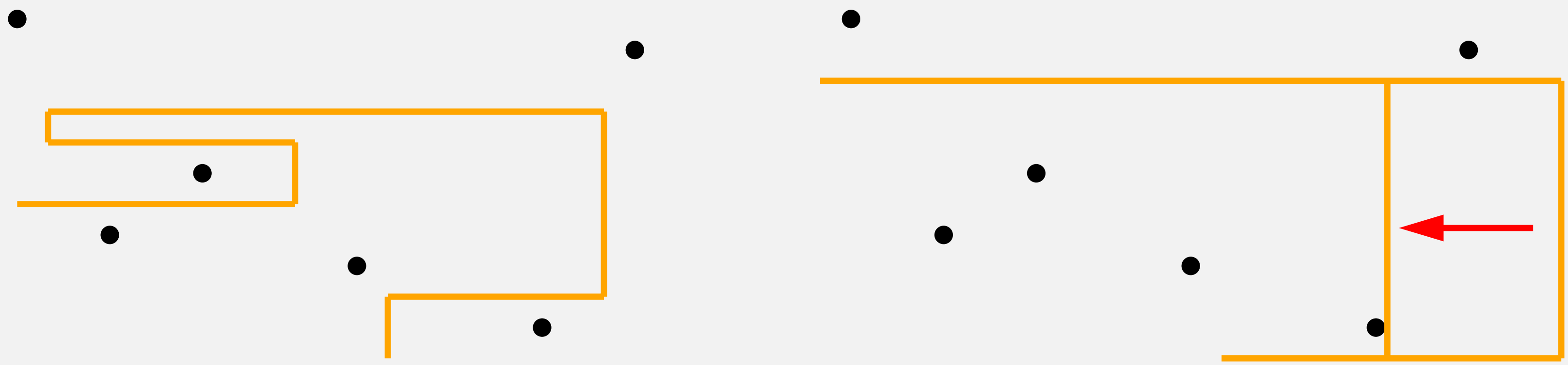
incrementally, following the orthogonal version of α



Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

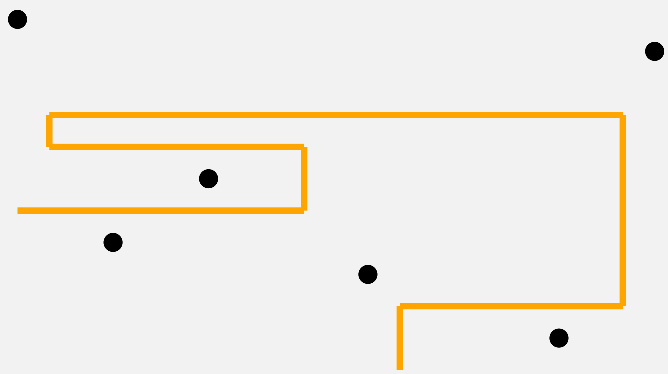
incrementally, following the orthogonal version of α

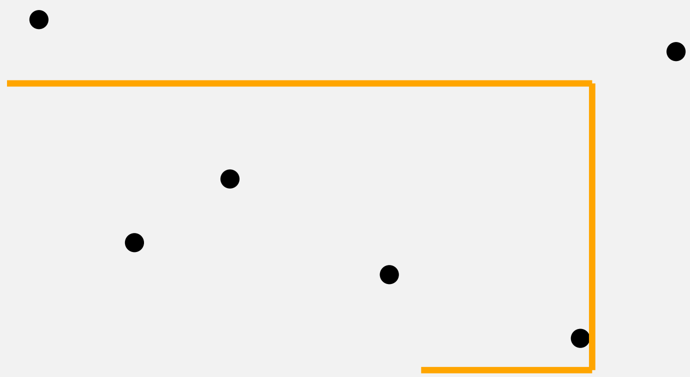


Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

incrementally, following the orthogonal version of α

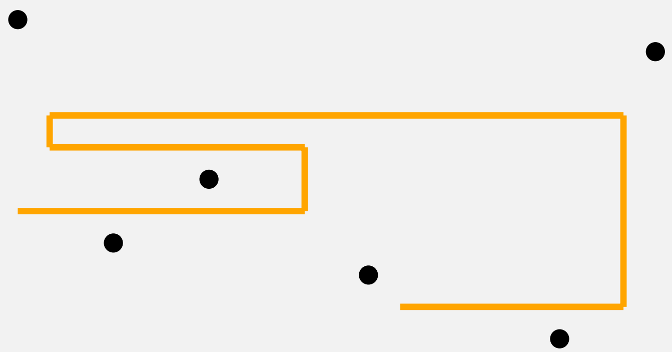


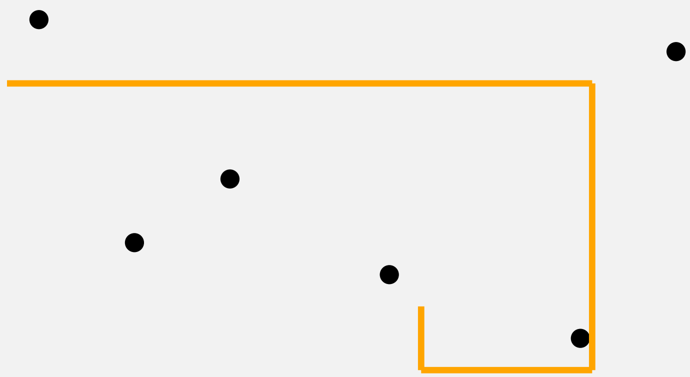


Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

incrementally, following the orthogonal version of α

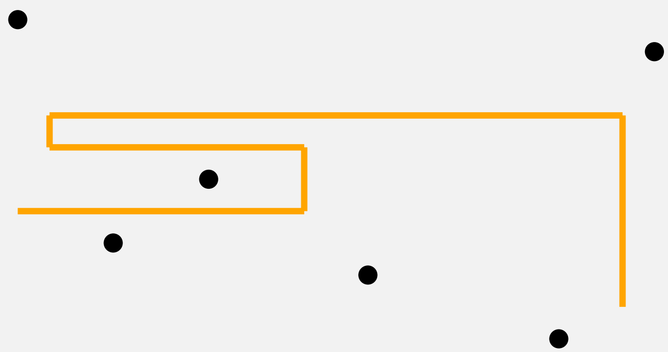


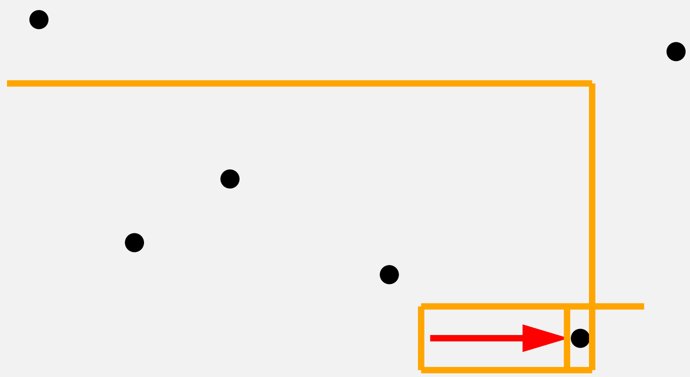


Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

incrementally, following the orthogonal version of α

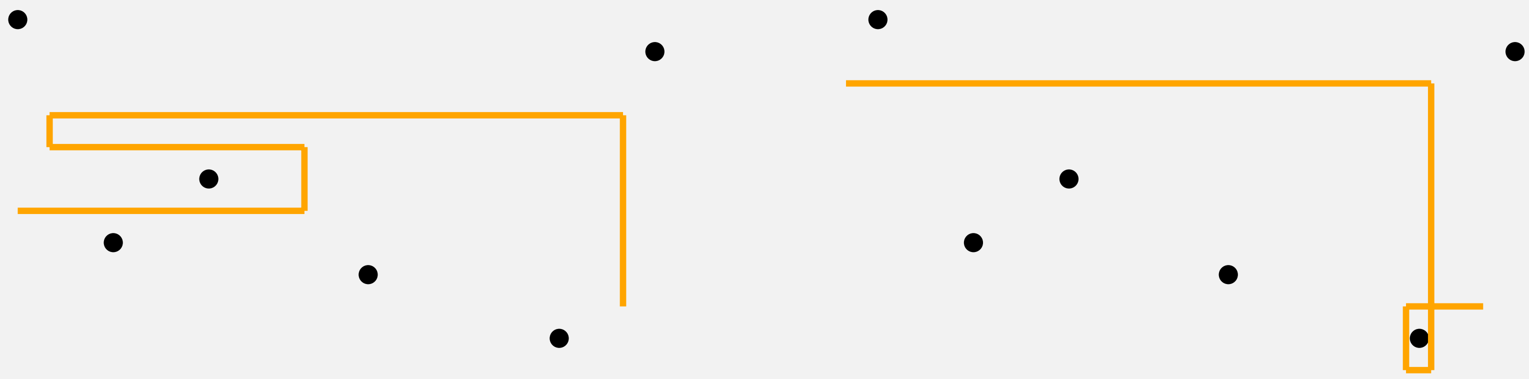




Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

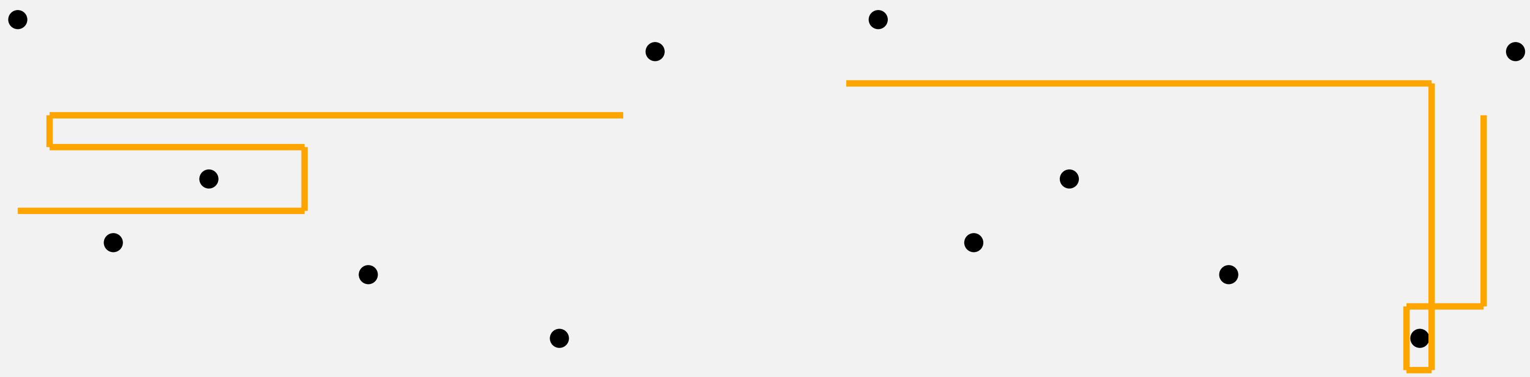
incrementally, following the orthogonal version of α



Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

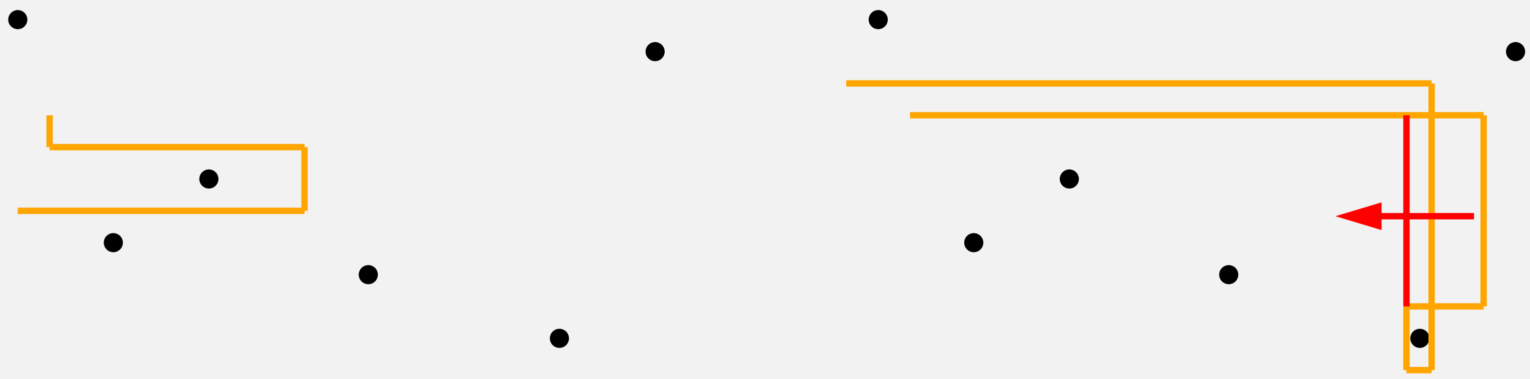
incrementally, following the orthogonal version of α



Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

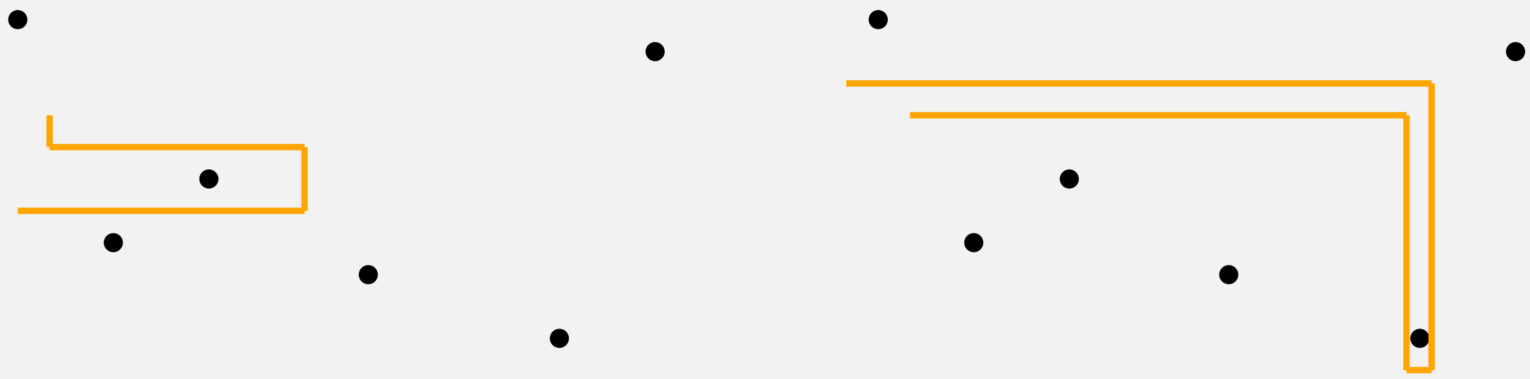
incrementally, following the orthogonal version of α



Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

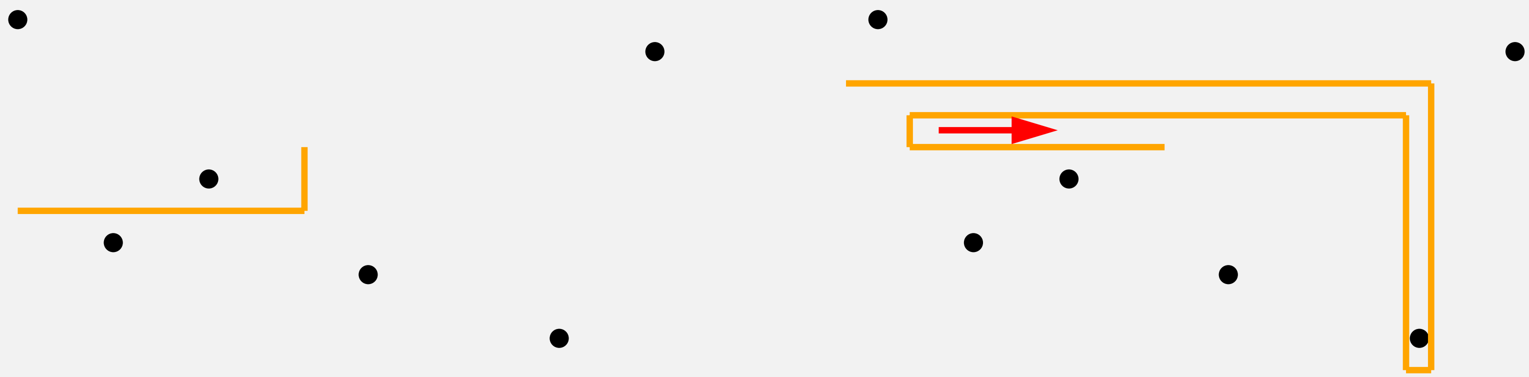
incrementally, following the orthogonal version of α



Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

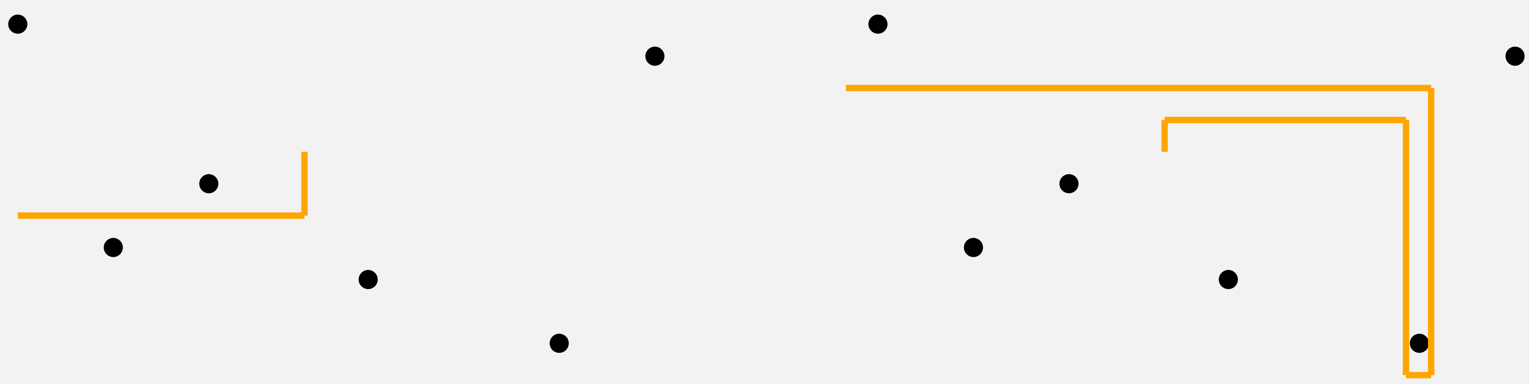
incrementally, following the orthogonal version of α



Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

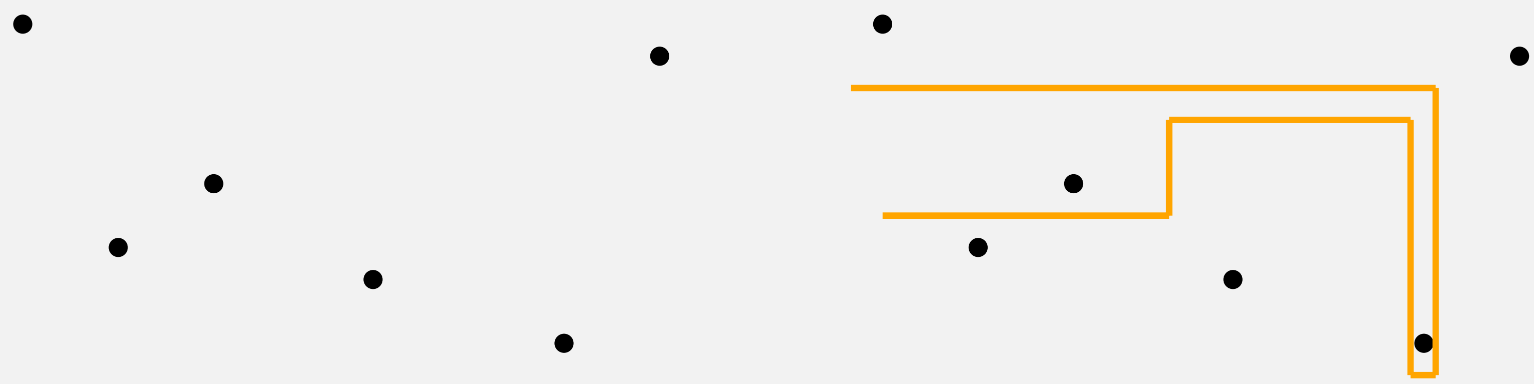
incrementally, following the orthogonal version of α



Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

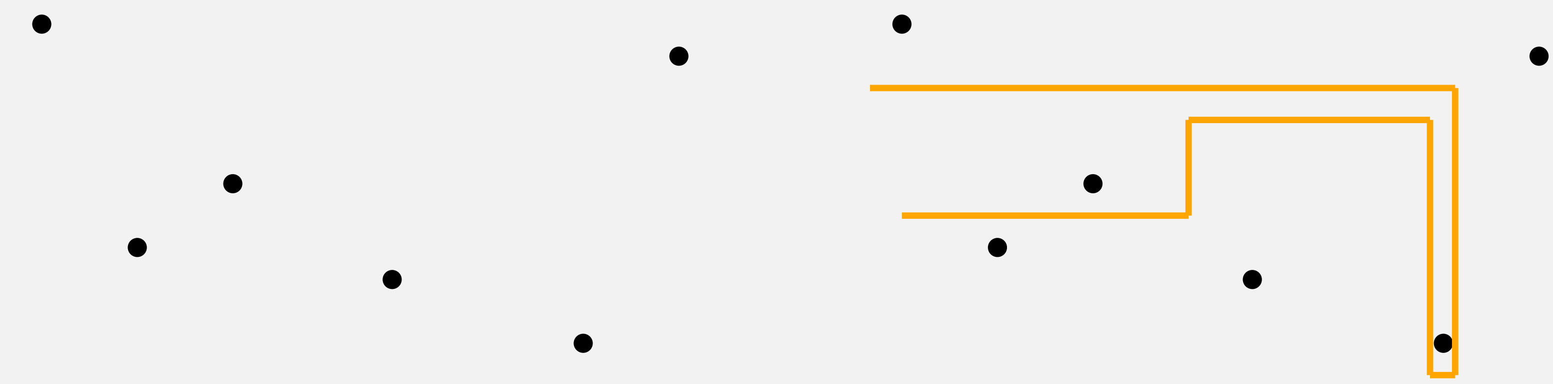
incrementally, following the orthogonal version of α



Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

incrementally, following the orthogonal version of α

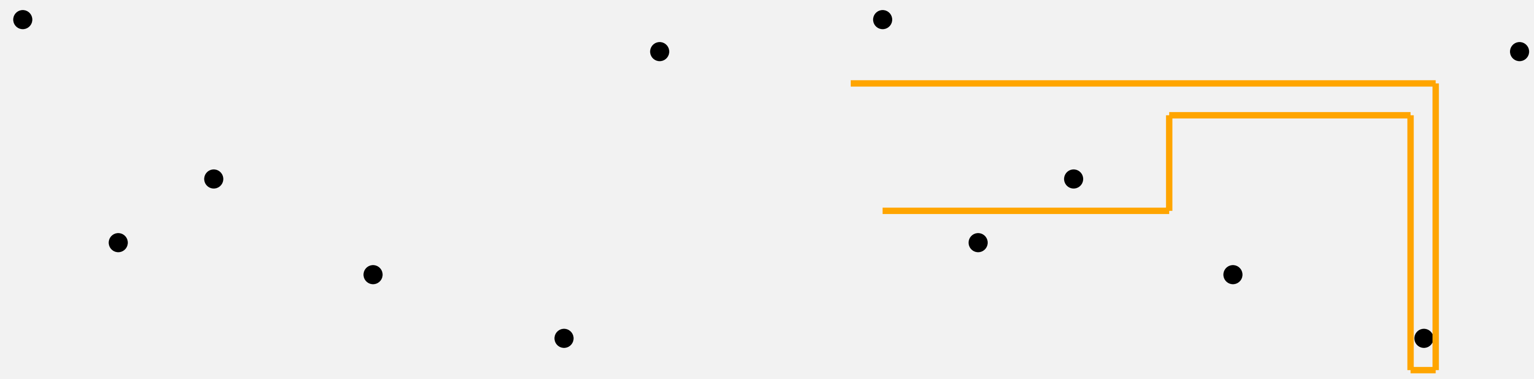


Problem: α' can have self-intersections

Algorithmical issues (Simple)

(path α' whith $seq(\alpha') = can(\alpha)$)

incrementally, following the orthogonal version of α



Problem: α' can have self-intersections

Lemma: α' has $\leq 2n$ self-intersections.

Algorithmical issues (Simple)

Theorem: We can decide in $O(n \log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

Proof:

1. convert the problem to an orthogonal one (done)
2. path α' whose $seq(\alpha') = can(\alpha)$ (done)
3. path β' whose $seq(\beta') = can(\beta)$ (done)
4. compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$

ALGORITHMICAL ISSUES (SIMPLE)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

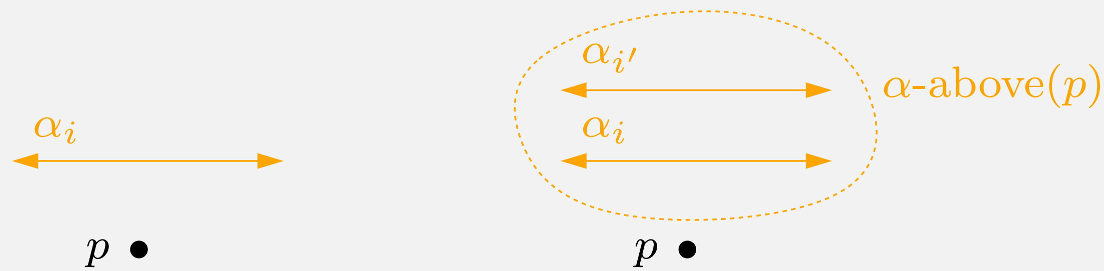
- "turn points" ($\overline{p}\underline{p}$ or $\underline{p}\overline{p}$) in α', β' should be the same
- split α' into monotone pieces $\alpha'_1, \ldots, \alpha'_m$ at turn points
- split β' into monotone pieces $\beta'_1, \ldots, \beta'_m$ at turn points

Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

- "turn points" ($\overline{p}\underline{p}$ or $\underline{p}\overline{p}$) in α', β' should be the same
- split α' into monotone pieces $\alpha'_1, \ldots, \alpha'_m$ at turn points
- split β' into monotone pieces $\beta'_1, \ldots, \beta'_m$ at turn points

$$p \in seq(\alpha'_i) \quad \Leftrightarrow \quad \alpha'_i \in \{\alpha'_k \mid p \in \alpha'_k\} =: \alpha'\text{-above}(p)$$

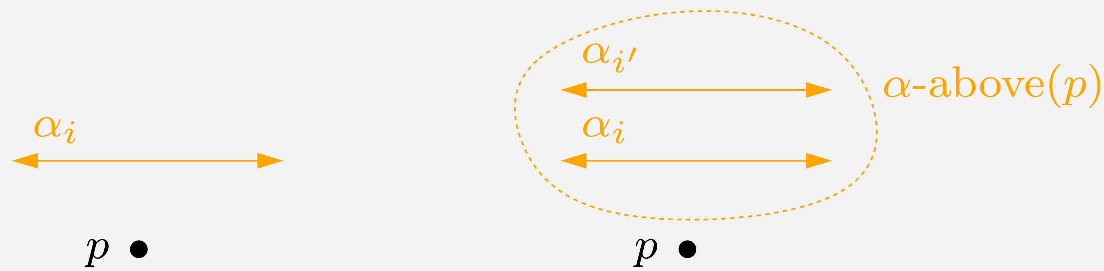


Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

- "turn points" ($\overline{p}\underline{p}$ or $\underline{p}\overline{p}$) in α', β' should be the same
- split α' into monotone pieces $\alpha'_1, \ldots, \alpha'_m$ at turn points
- split β' into monotone pieces $\beta'_1, \ldots, \beta'_m$ at turn points

$$p \in seq(\alpha'_i) \quad \Leftrightarrow \quad \alpha'_i \in \{\alpha'_k \mid p \in \alpha'_k\} =: \alpha'\text{-above}(p)$$

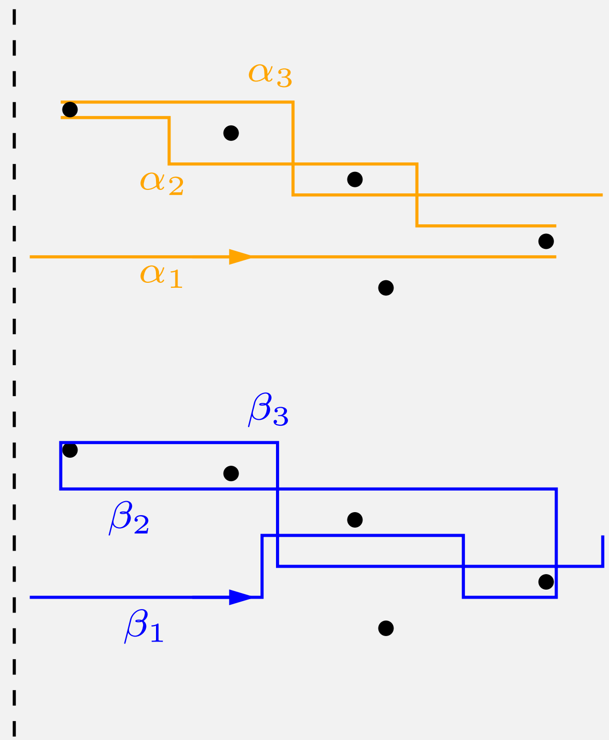


$$seq(\alpha') = seq(\beta') \quad \Leftrightarrow \quad \alpha'\text{-above}(p) = \beta'\text{-above}(p) \text{ for all } p \in P$$

Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

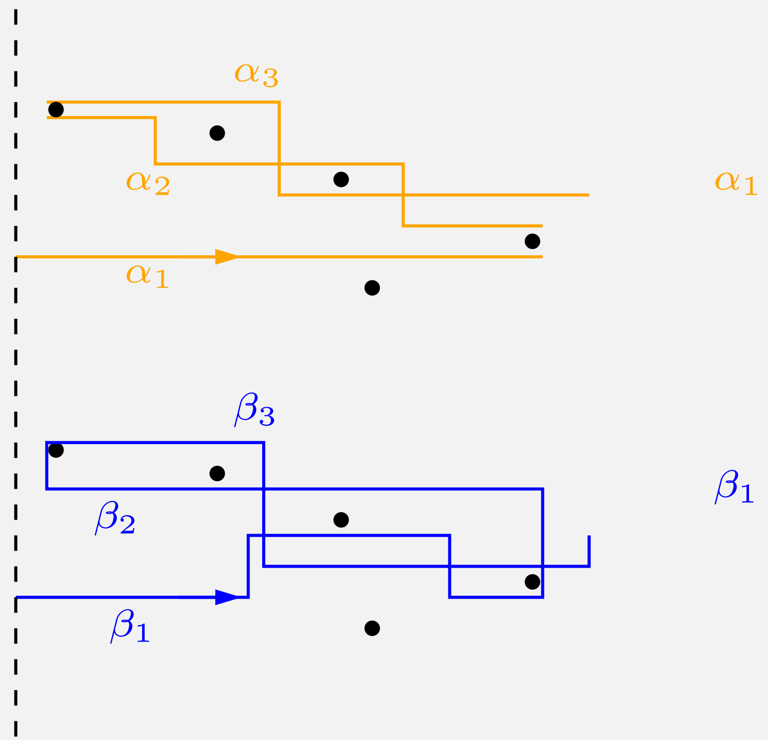
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

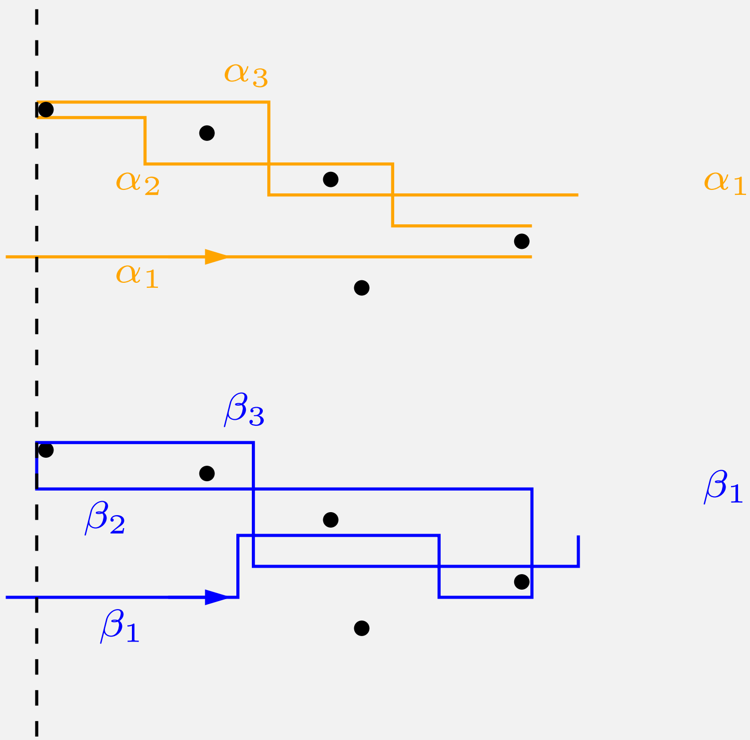
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

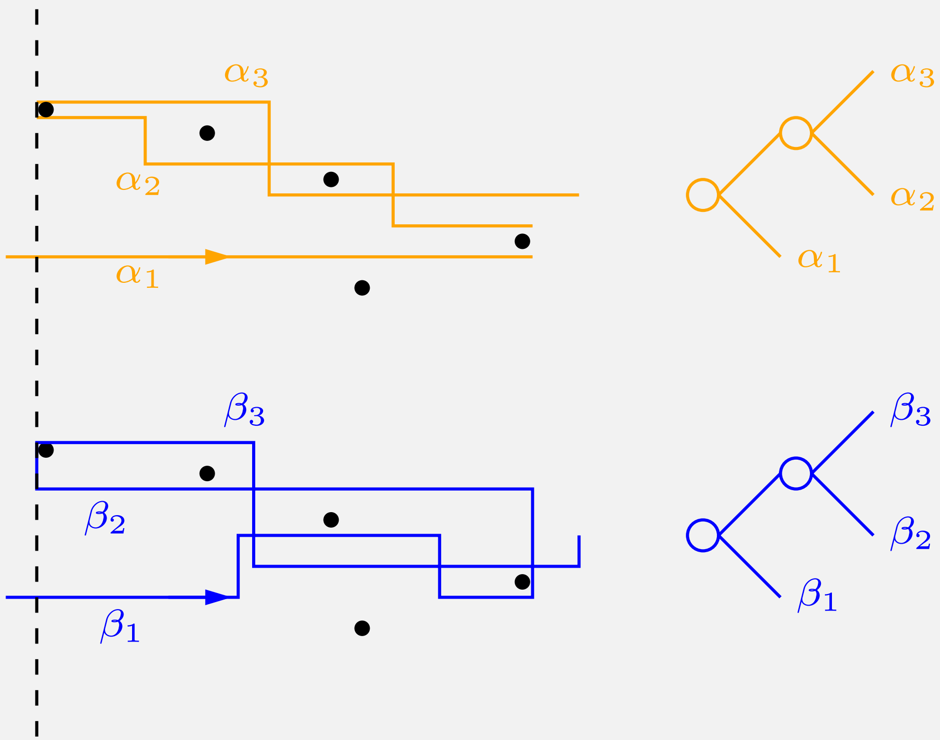
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

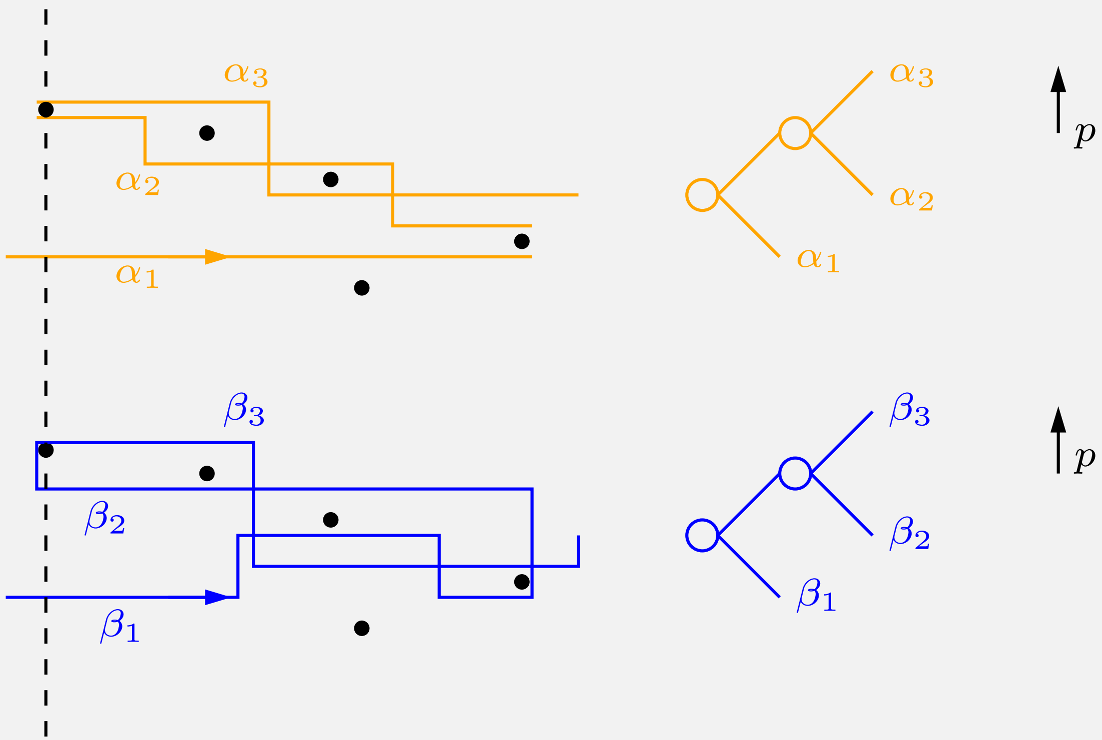
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

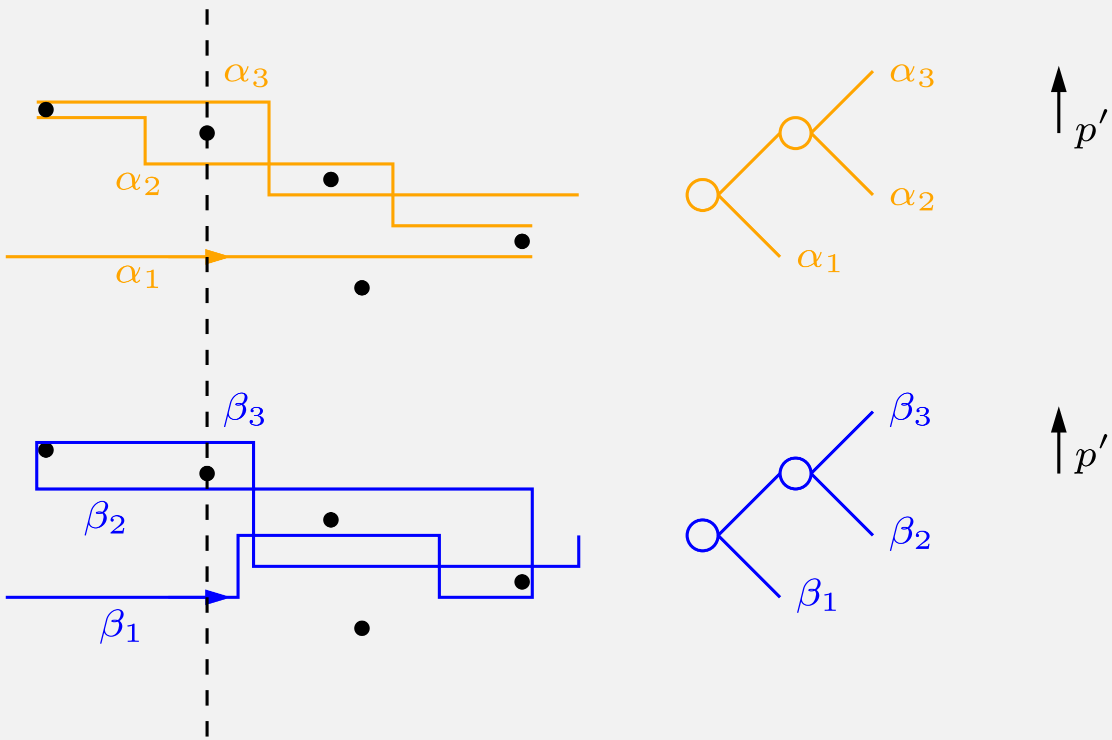
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

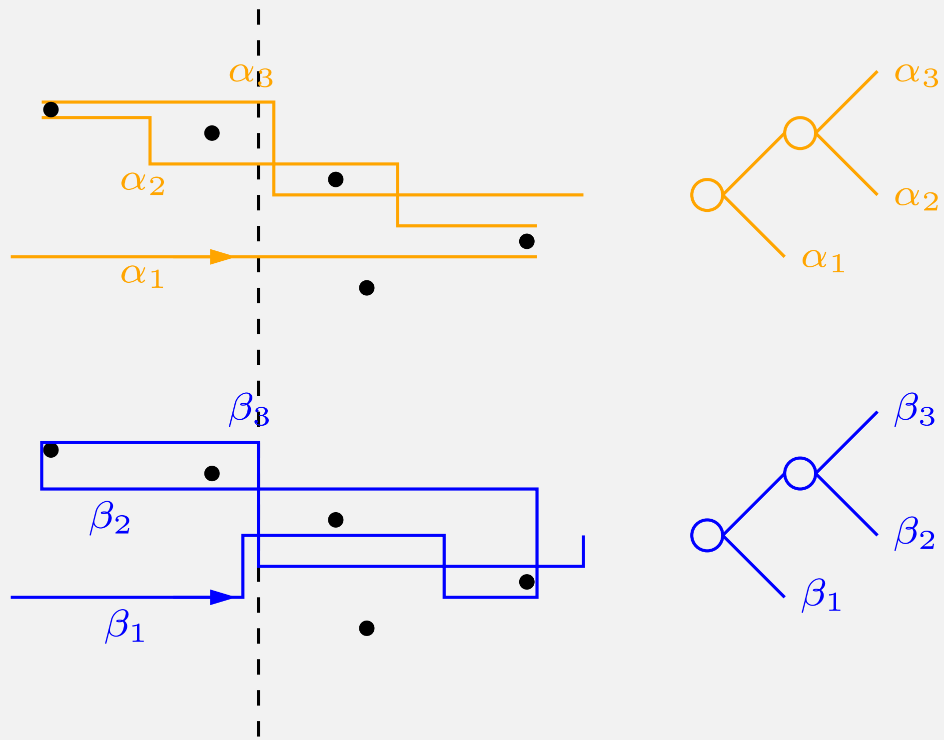
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

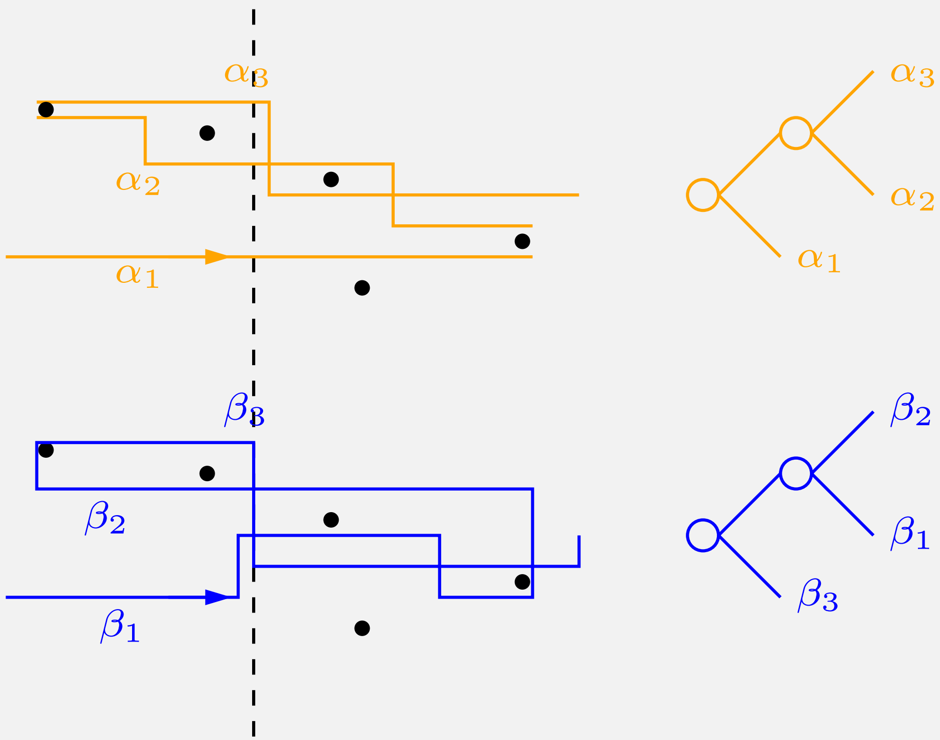
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

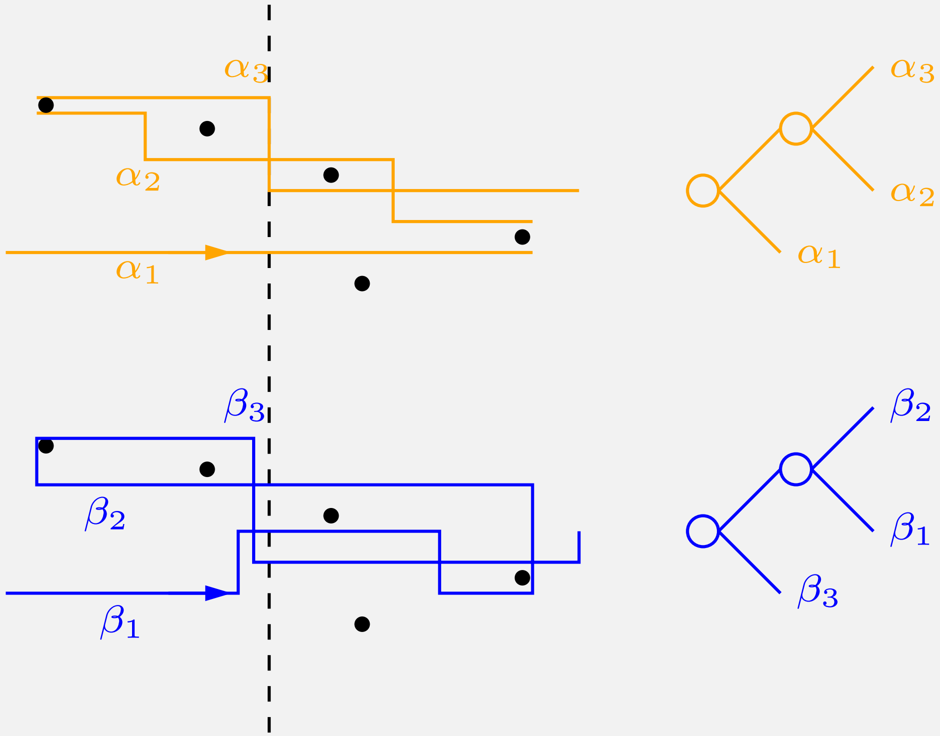
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

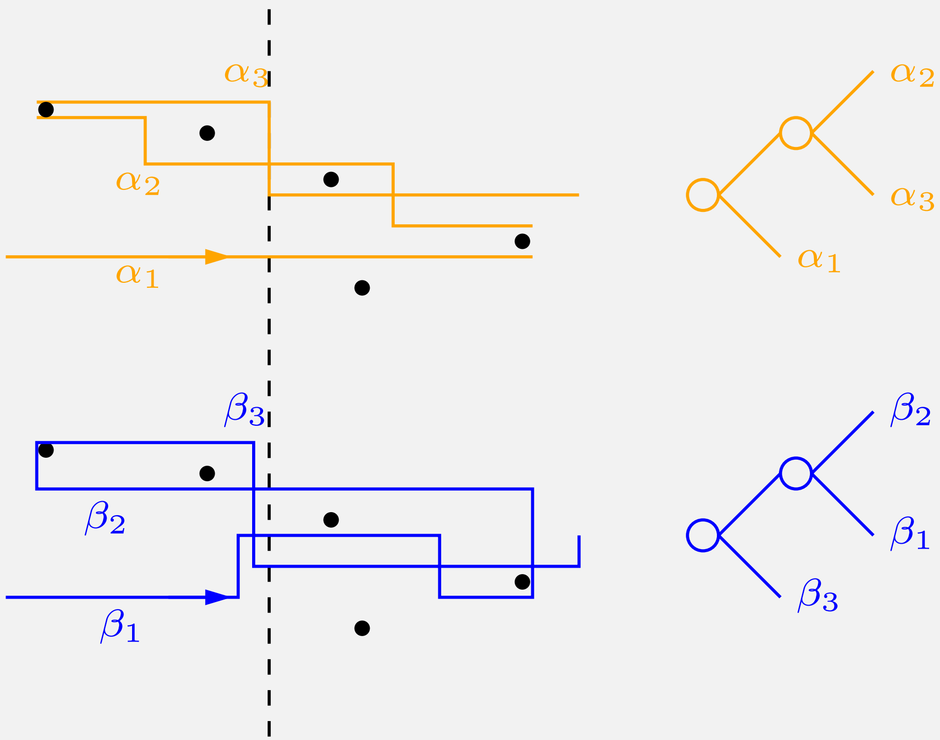
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

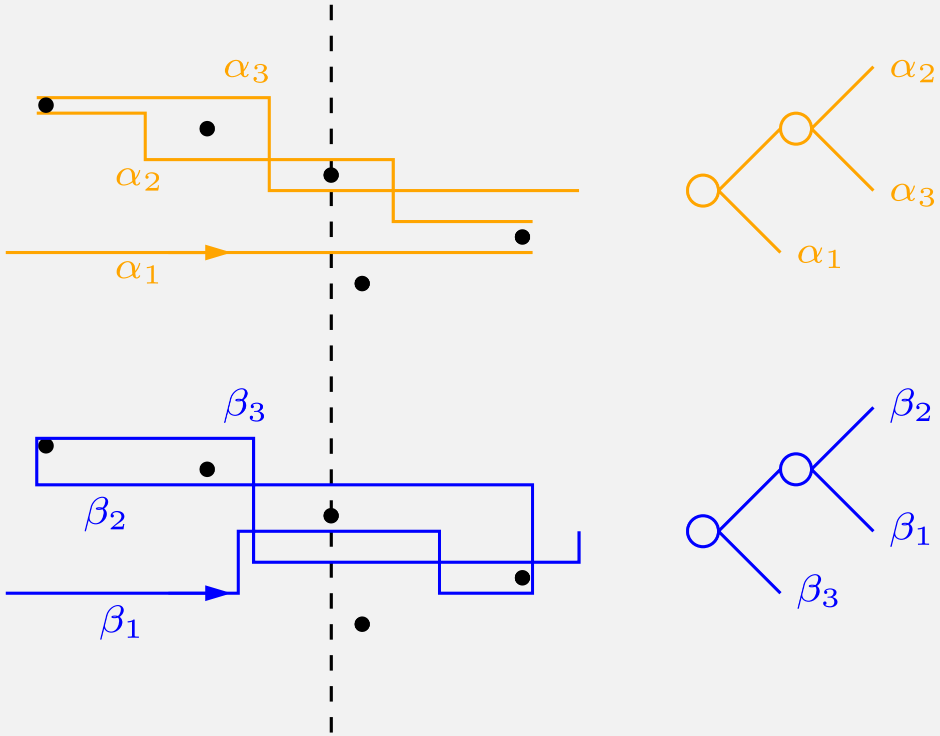
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

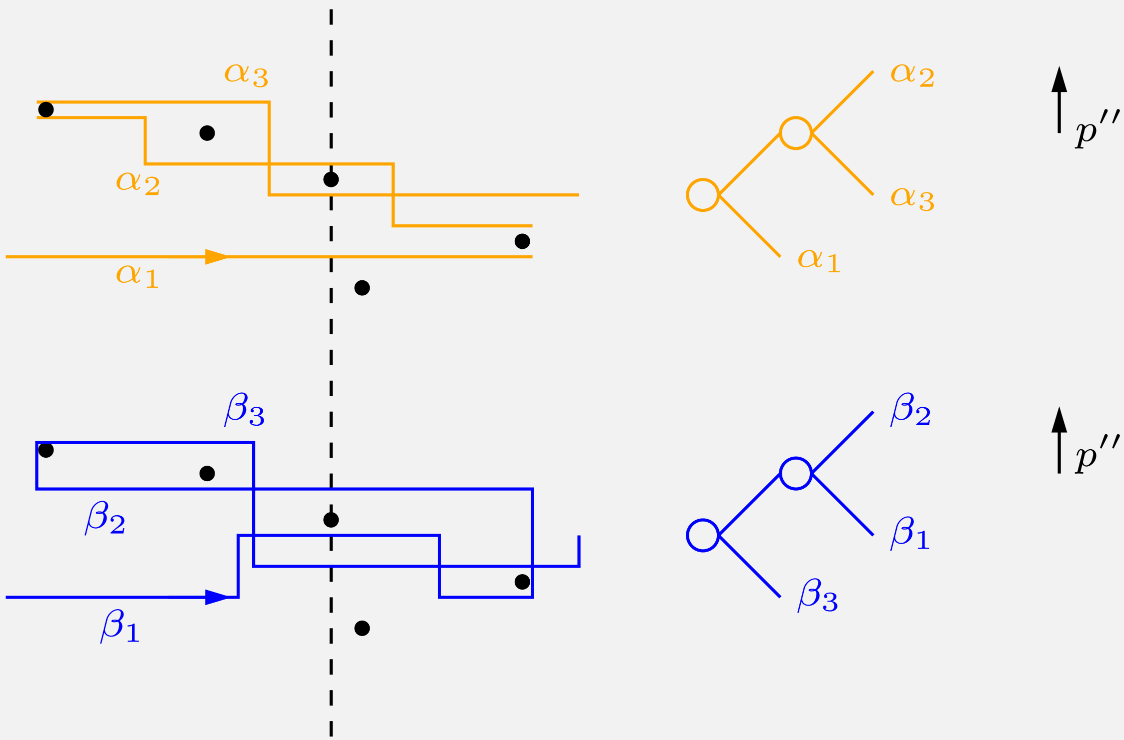
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

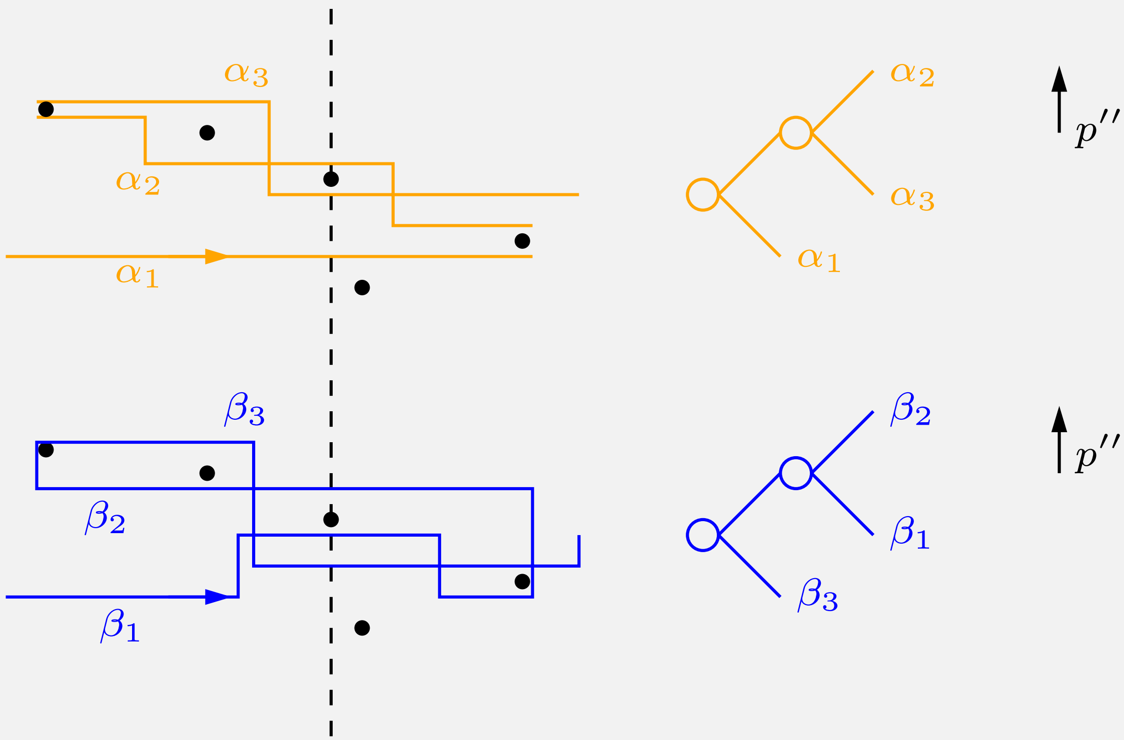
We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



Algorithmical issues (Simple)

(compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$)

We compare α'-above$(p) = \beta'$-above(p) for all $p \in P$ with a sweep.



We store $|\alpha'\text{-above}(k) \triangle \beta'\text{-above}(k)|$ for any level k. $O(n)$ updates!

Algorithmical issues (Simple)

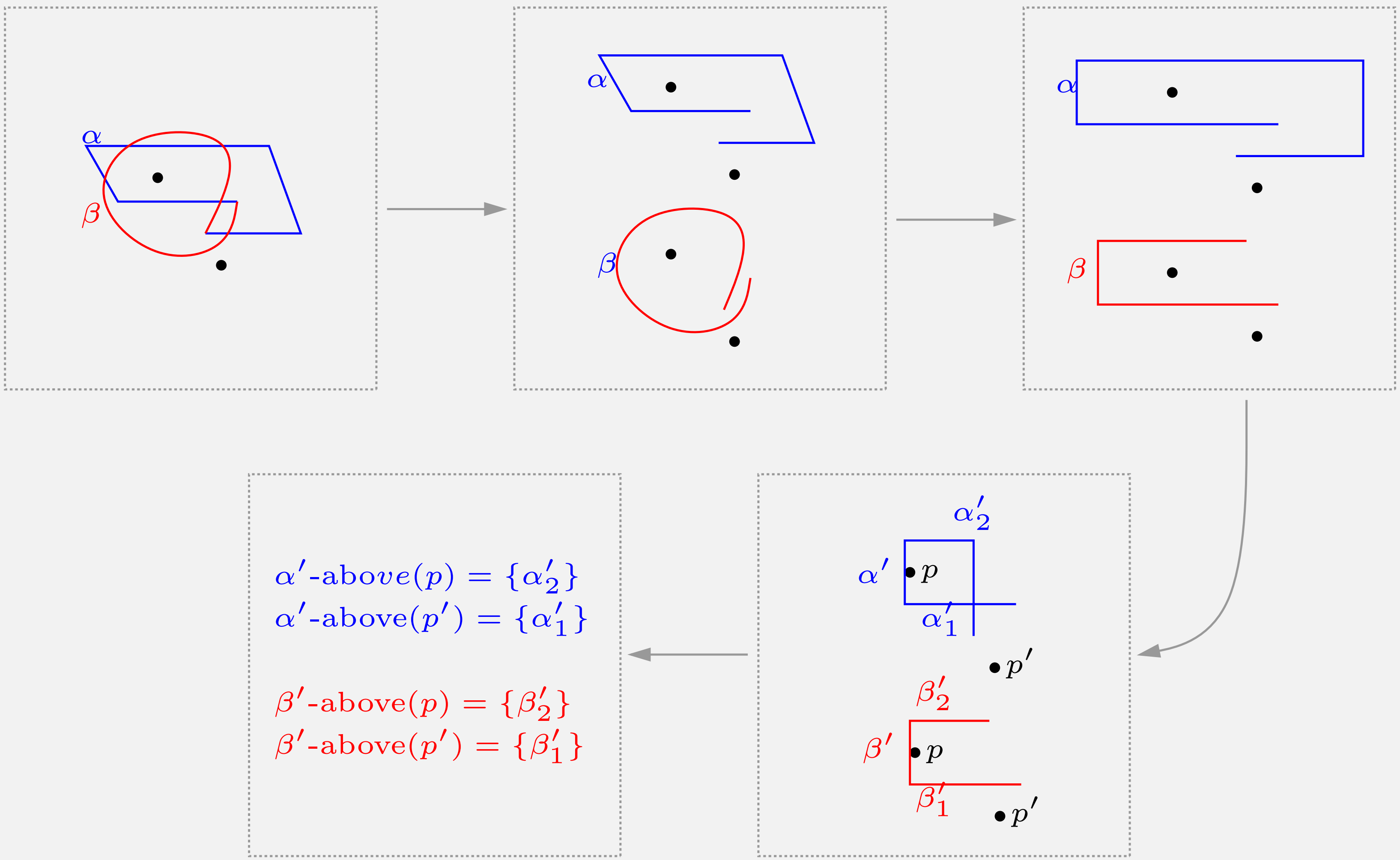
Theorem: We can decide in $O(n \log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

Proof:

1. convert the problem to an orthogonal one (done)
2. path α' whose $seq(\alpha') = can(\alpha)$ (done)
3. path β' whose $seq(\beta') = can(\beta)$ (done)
4. compare $seq(\alpha')$ and $seq(\beta')$ without using $seq(\alpha'), seq(\beta')$ (done)

$\diamond$

Mini-example



Lower bound (Simple)

Reduction from:

> Given n disordered numbers $x_1, \ldots, x_n$, and n disjoint intervals $[a_1, b_1], \ldots, [a_n, b_n]$ in increasing order.

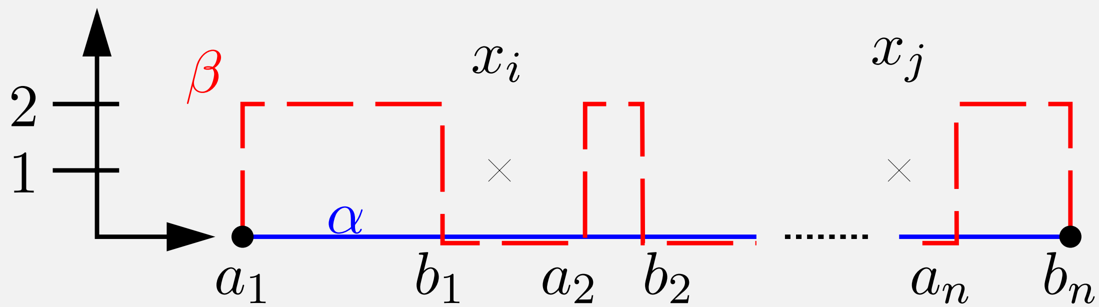
Does any $x_i \in [a_j, b_j]$?

Lower bound (Simple)

Reduction from:

Given n disordered numbers $x_1, \ldots, x_n$, and n disjoint intervals $[a_1, b_1], \ldots, [a_n, b_n]$ in increasing order.

Does any $x_i \in [a_j, b_j]$?

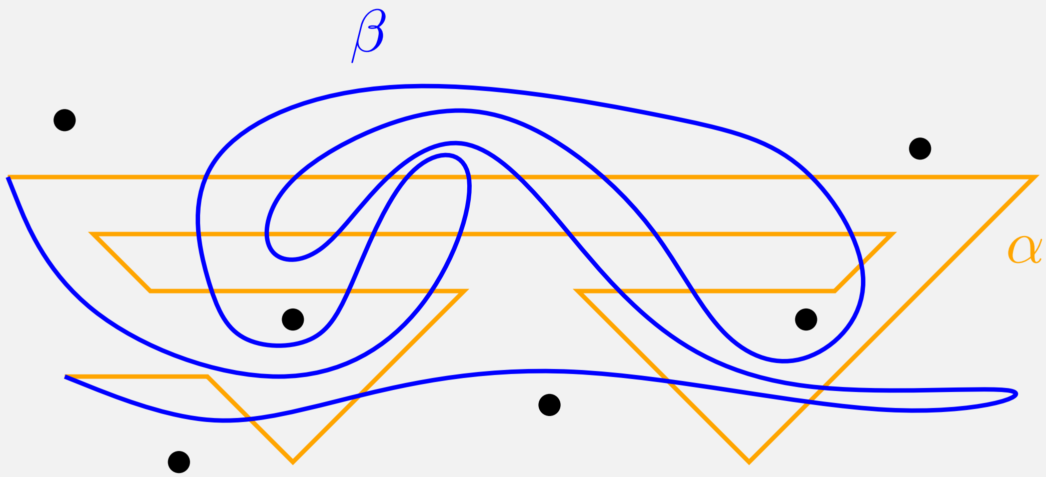


Both have $\Omega(n \log n)$ in the decision tree model.

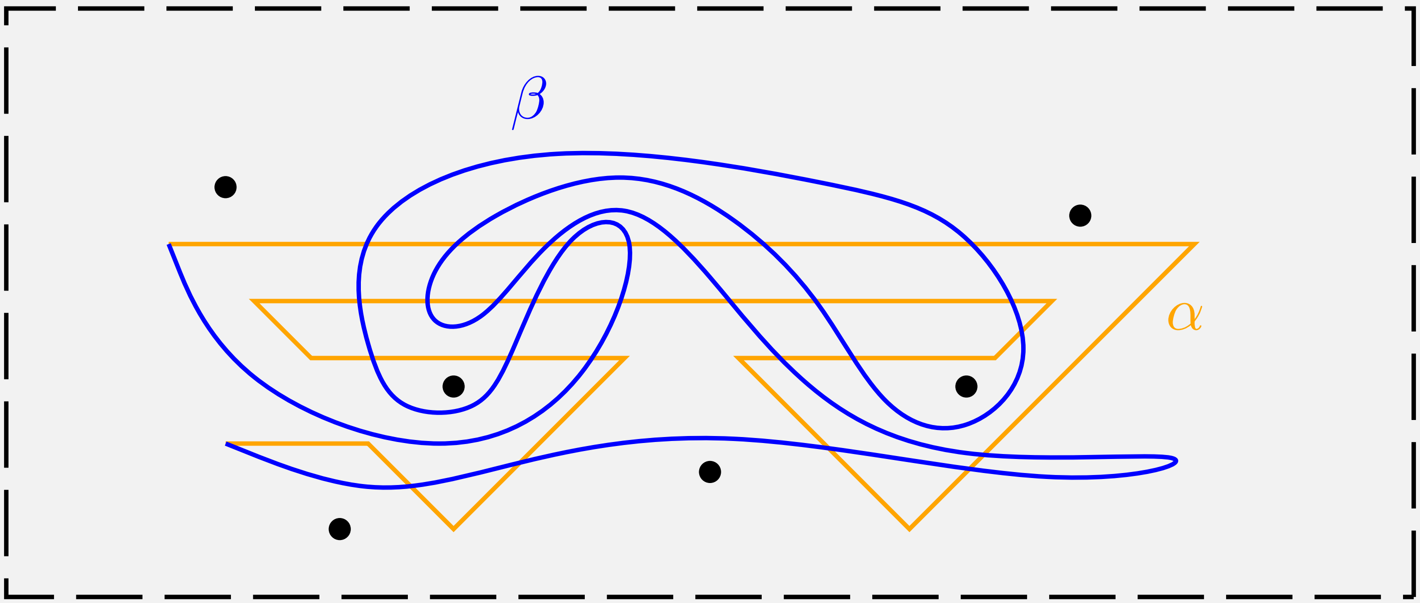
Testing homotopy for non-simple paths

- Topological oracle.
- Algorithmical issues.
- Lower bound.

Topological oracle (Non-simple)

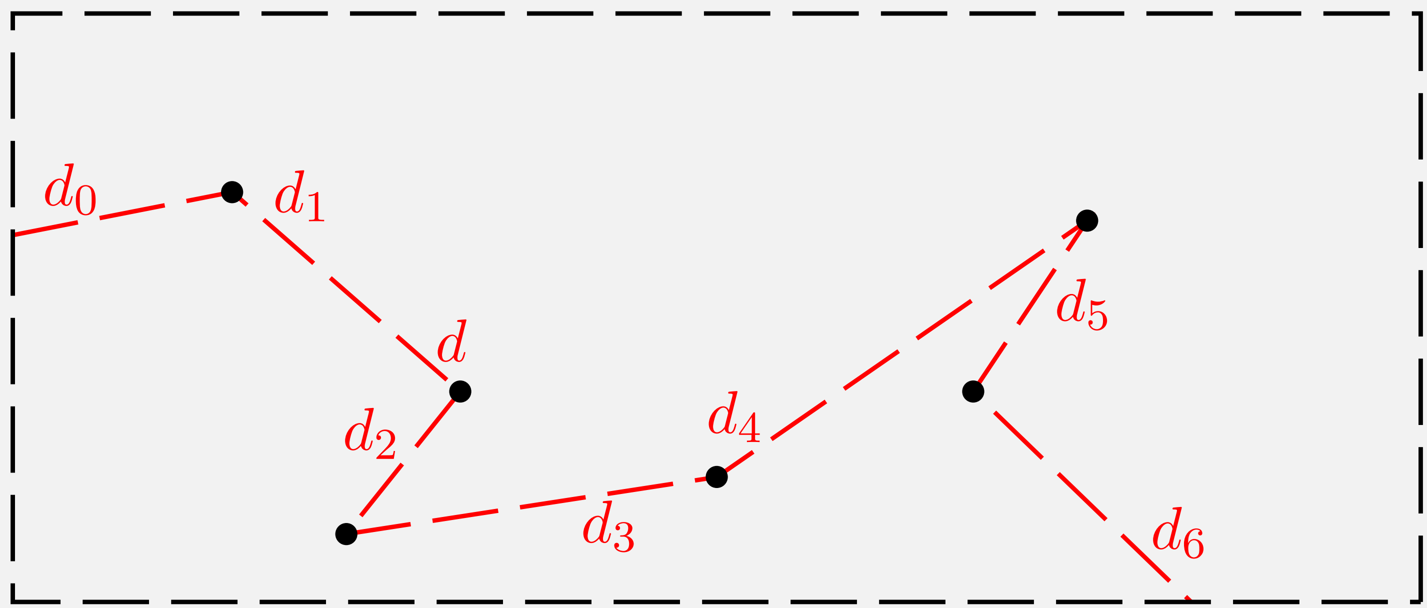


Topological oracle (Non-simple)



B a big bounding box.

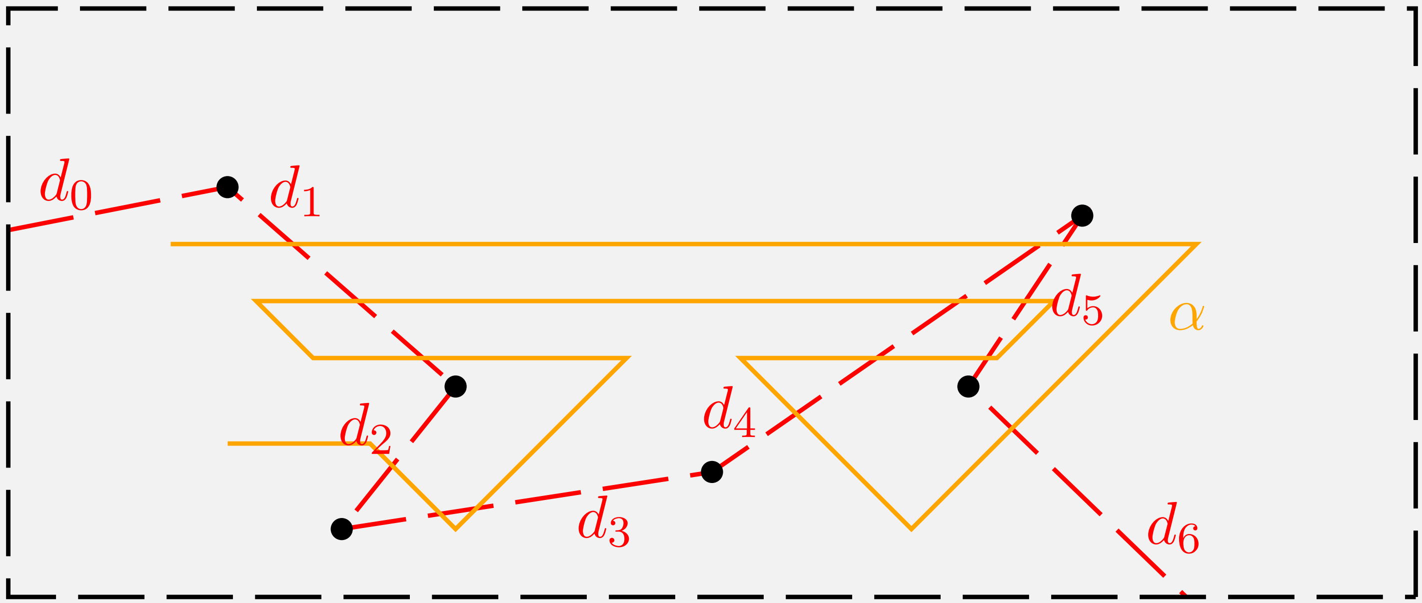
Topological oracle (Non-simple)



B a big bounding box.

d a simple path through P and splitting B.

Topological oracle (Non-simple)

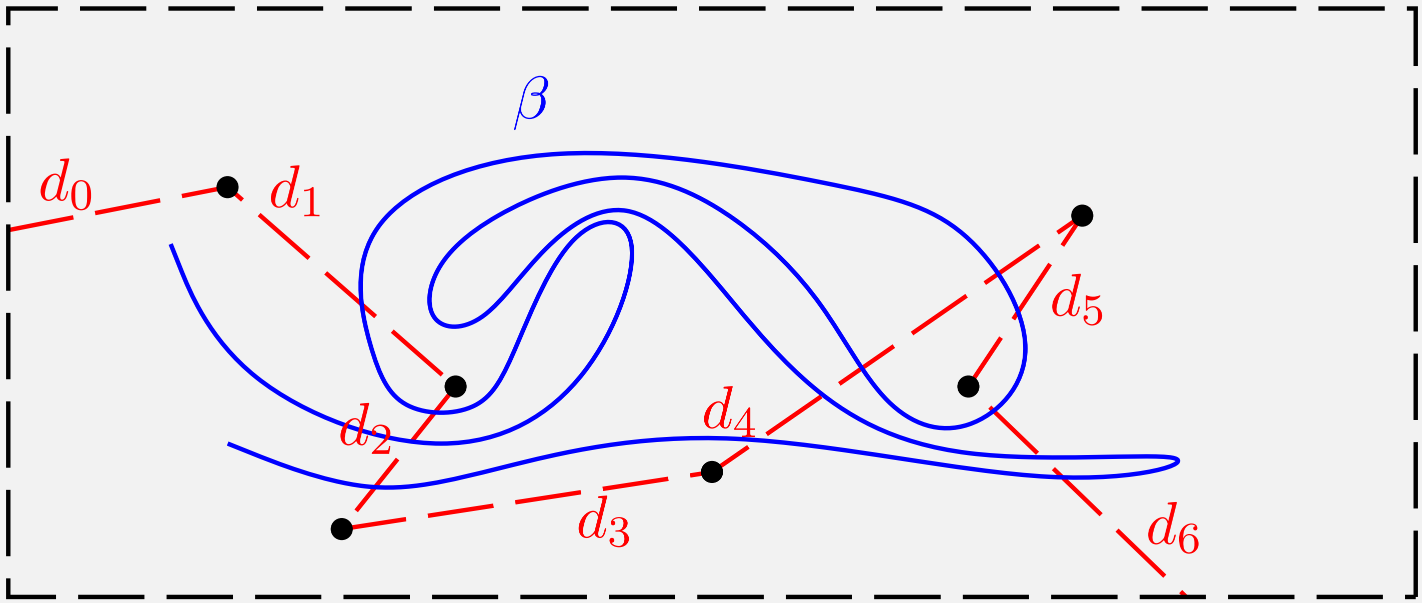


B a big bounding box.

d a simple path through P and splitting B.

$\alpha \equiv 14564455411332$.

Topological oracle (Non-simple)



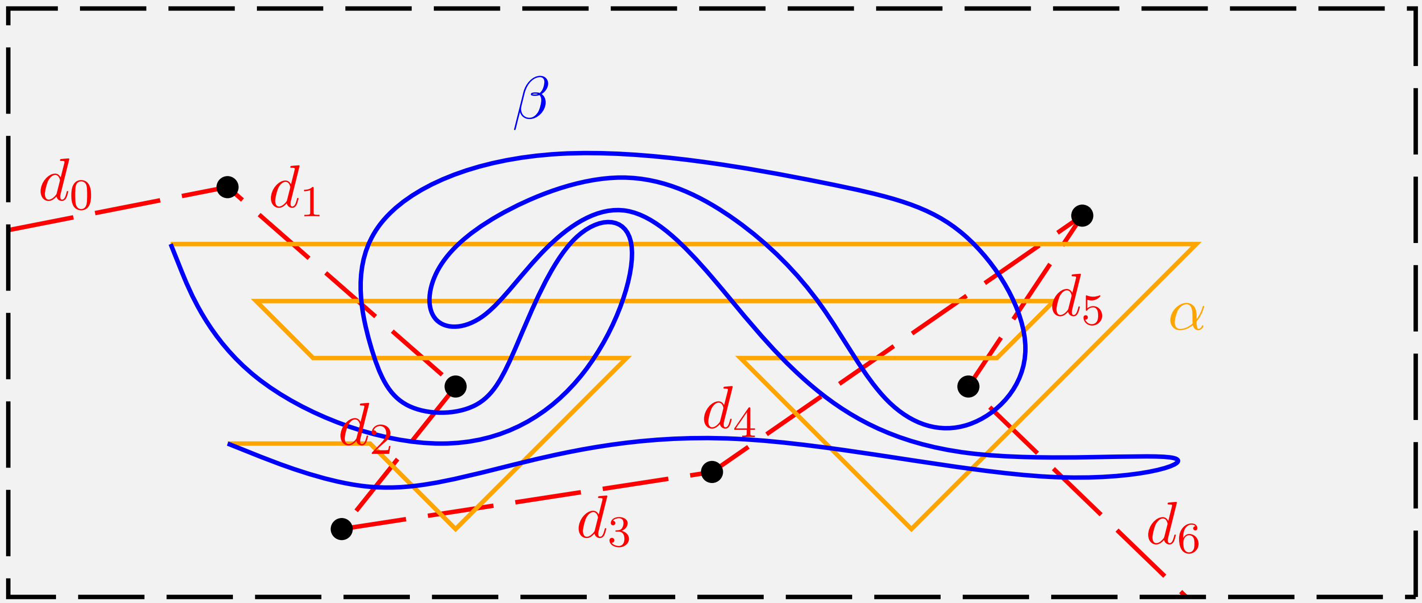
B a big bounding box.

d a simple path through P and splitting B.

$\alpha \equiv 14564455411332$.

$\beta \equiv 221456446642$.

Topological oracle (Non-simple)



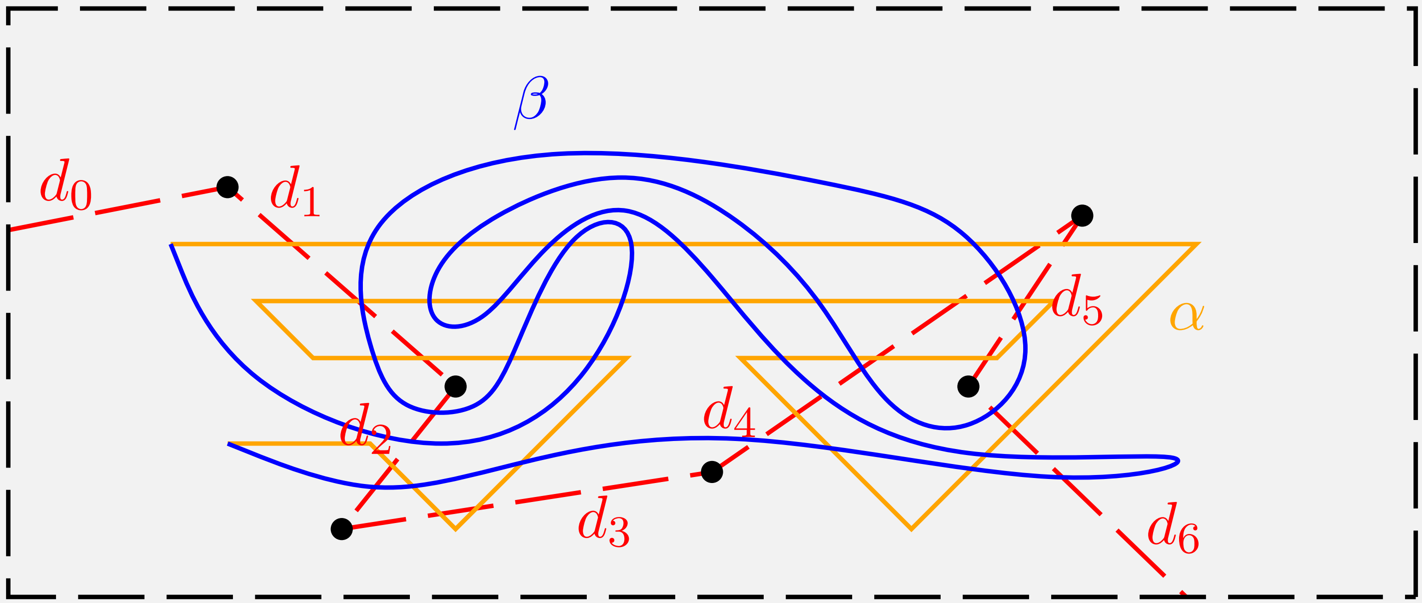
B a big bounding box.

d a simple path through P and splitting B.

$\alpha \equiv 14564455411332$.

$\beta \equiv 221456446642$.

Topological oracle (Non-simple)



B a big bounding box.

d a simple path through P and splitting B.

$\alpha \equiv 14564455411332 \;\Rightarrow\; \alpha \equiv 145642.$

$\beta \equiv 221456446642 \;\Rightarrow\; \beta \equiv 145642.$

α, β homotopic iff same simplified sequence.

Theorem: We can decide in $O(n^{3/2}\log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

Proof:

- compute spanning tree of P with crossing number $O(\sqrt{n})$
- compute simple path d with crossing number $O(\sqrt{n})$
- compute the sequences $\alpha \cap d$ and $\beta \cap d$, simplify them, and compare them

Algorithmical issues (Non-simple)

(spanning tree of P with c.n. $O(\sqrt{n})$)

$G = (P, E)$ a straight-line embeddeding
l a line (disjoint from P)
Definition: The *crossing number of* l in G is

$$cr(l, G) := |l \cap E|.$$

The *crossing number* of G is

$$cr(G) := \max_{l}\{cr(l, G)\}.$$

Algorithmical issues (Non-simple)

(spanning tree of P with c.n. $O(\sqrt{n})$)

$G = (P, E)$ a straight-line embeddeding
l a line (disjoint from P)
Definition: The *crossing number of* l in G is

$$cr(l, G) := |l \cap E|.$$

The *crossing number* of G is

$$cr(G) := \max_{l}\{cr(l, G)\}.$$

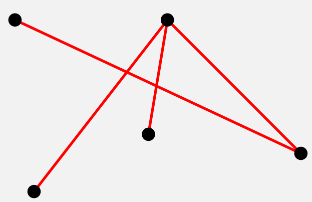
Theorem: For any set P, we can compute in $O(n^{1+\epsilon})$ a tree T that spans P and has $cr(T) = O(\sqrt{n})$. That is, for any line l, $|l \cap T| = O(\sqrt{n})$.

Theorem: We can decide in $O(n^{3/2}\log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

Proof:

1. compute spanning tree of P with crossing number $O(\sqrt{n})$
2. compute simple path d with crossing number $O(\sqrt{n})$
3. compute the sequences $\alpha \cap d$ and $\beta \cap d$, simplify them, and compare them

$O(n^{1+\epsilon})$



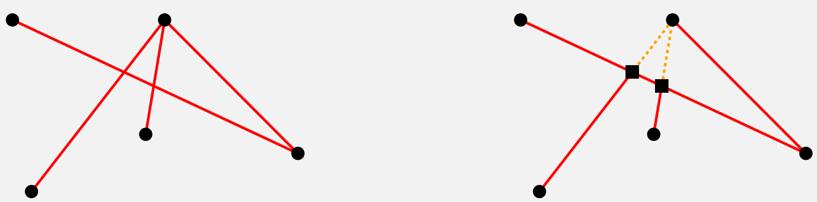
Algorithmical issues (Non-simple)

Theorem: We can decide in $O(n^{3/2} \log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

Proof:

1. compute spanning tree of P with crossing number $O(\sqrt{n})$
2. compute simple path d with crossing number $O(\sqrt{n})$
3. compute the sequences $\alpha \cap d$ and $\beta \cap d$, simplify them, and compare them

$O(n^{1+\epsilon}) + nO(\log^2 n)$

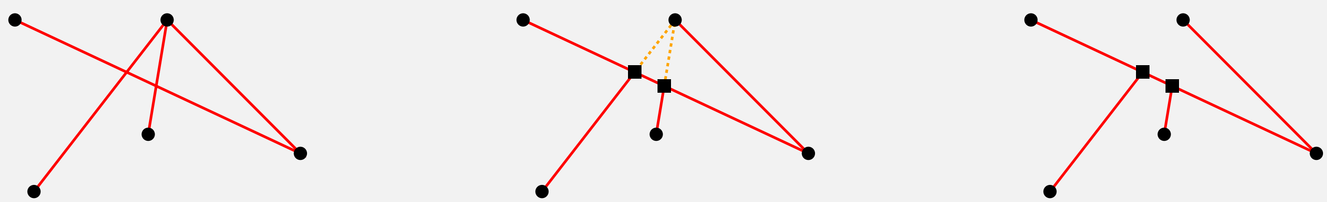


Theorem: We can decide in $O(n^{3/2}\log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

Proof:

1. compute spanning tree of P with crossing number $O(\sqrt{n})$
2. compute simple path d with crossing number $O(\sqrt{n})$
3. compute the sequences $\alpha \cap d$ and $\beta \cap d$, simplify them, and compare them

$O(n^{1+\epsilon}) + nO(\log^2 n)$



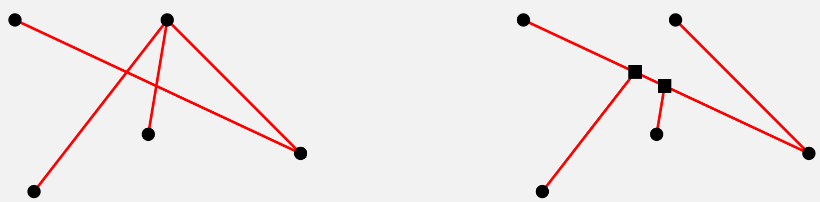
Algorithmical issues (Non-simple)

Theorem: We can decide in $O(n^{3/2} \log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

Proof:

1. compute spanning tree of P with crossing number $O(\sqrt{n})$
2. compute simple path d with crossing number $O(\sqrt{n})$
3. compute the sequences $\alpha \cap d$ and $\beta \cap d$, simplify them, and compare them

$O(n^{1+\epsilon}) + nO(\log^2 n)$



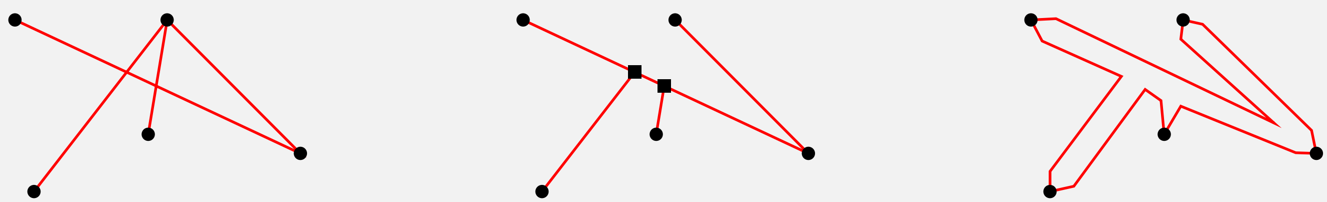
Algorithmical issues (Non-simple)

Theorem: We can decide in $O(n^{3/2}\log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

Proof:

1. compute spanning tree of P with crossing number $O(\sqrt{n})$
2. compute simple path d with crossing number $O(\sqrt{n})$
3. compute the sequences $\alpha \cap d$ and $\beta \cap d$, simplify them, and compare them

$O(n^{1+\epsilon}) + nO(\log^2 n) + O(n)$

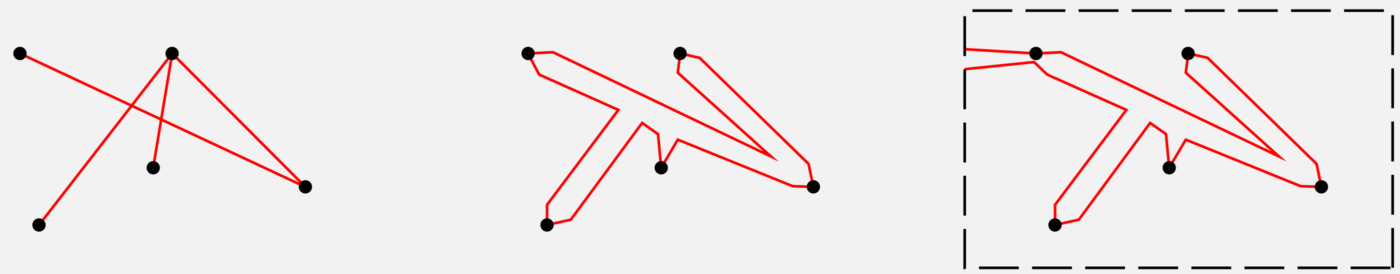


Theorem: We can decide in $O(n^{3/2} \log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

Proof:

1. compute spanning tree of P with crossing number $O(\sqrt{n})$
2. compute simple path d with crossing number $O(\sqrt{n})$
3. compute the sequences $\alpha \cap d$ and $\beta \cap d$, simplify them, and compare them

$O(n^{1+\epsilon}) + nO(\log^2 n) + O(n) + O(1)$



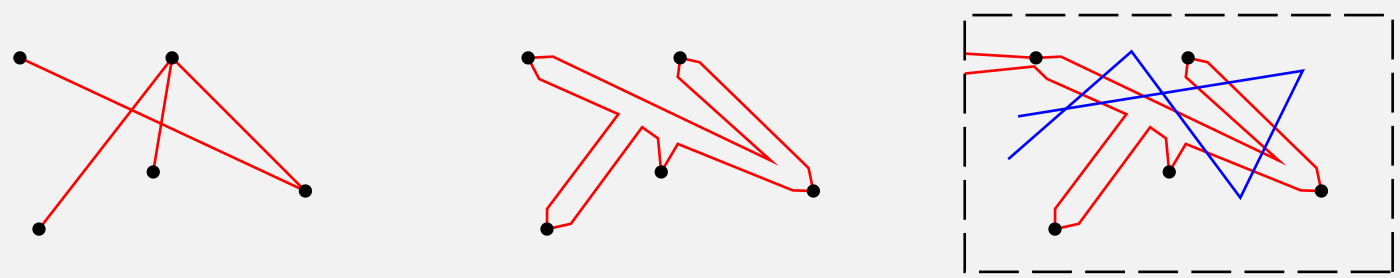
Algorithmical issues (Non-simple)

Theorem: We can decide in $O(n^{3/2}\log n)$ time if α and β are homotopic in $\mathbb{R}^2 \setminus P$.

Proof:

1. compute spanning tree of P with crossing number $O(\sqrt{n})$
2. compute simple path d with crossing number $O(\sqrt{n})$
3. compute the sequences $\alpha \cap d$ and $\beta \cap d$, simplify them, and compare them

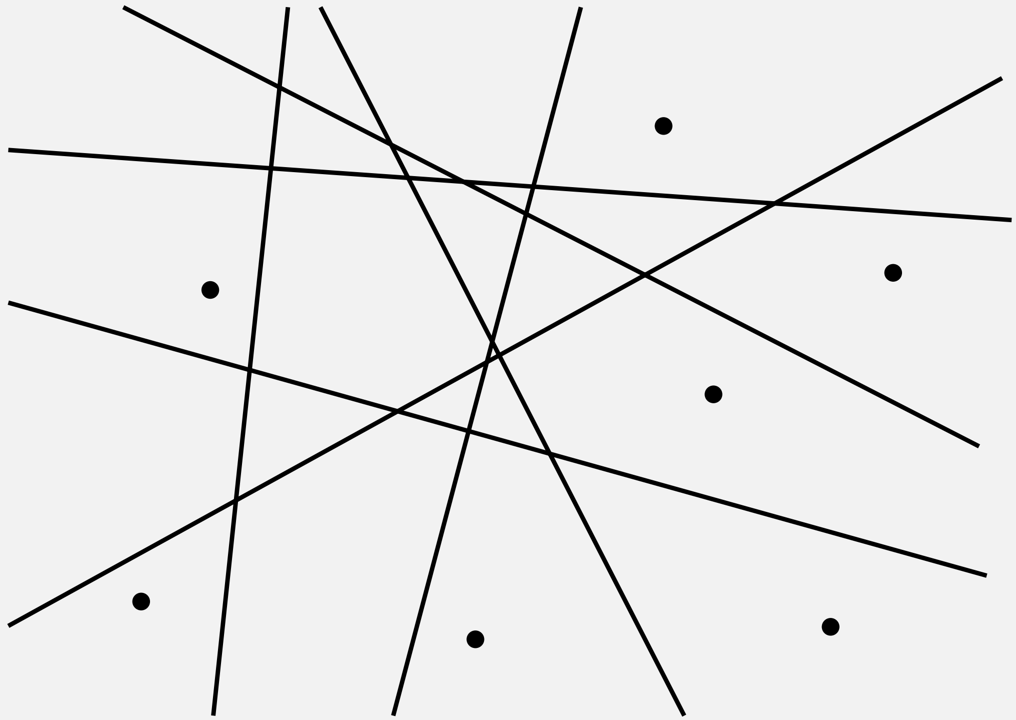
$O(n^{1+\epsilon}) + nO(\log^2 n) + O(n) + O(1) + 2O(n\sqrt{n}\log n)$



Lower bound (Non-simple)

We will use Hopcroft's problem

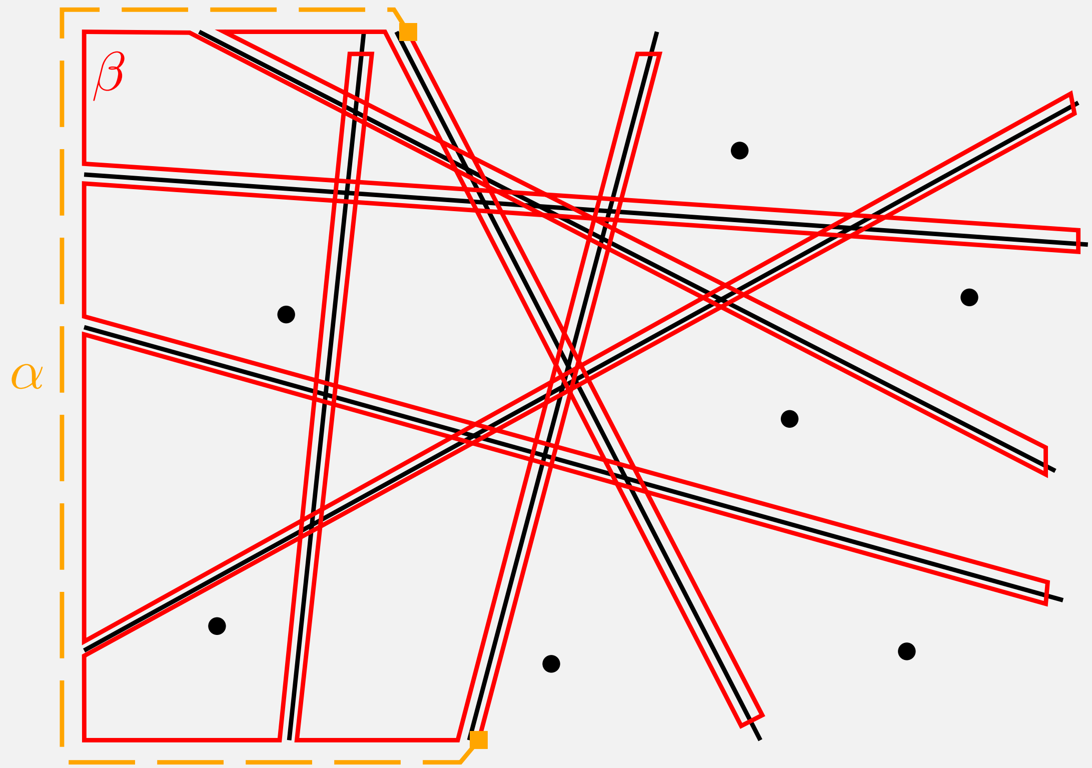
Given n points and n lines, does any point lie in any line?



Within *partition algorithms* $\Rightarrow \Omega(n^{4/3})$

Lower bound (Non-simple)

Reduction from Hopcroft's problem



Incidence point-line $\Leftrightarrow$ α, β no homotopic.

Our problem has a $\Omega(n^{4/3})$ lower bound.

What did I explain?

- concepts around homotopic paths
- testing homotopy for simple paths
  - solvable in $O(n \log n)$ time
  - lower bound $\Omega(n \log n)$
- testing homotopy for non-simple paths
  - solvable in $O(n^{3/2} \log n)$ time
  - lower bound $\Omega(n^{4/3})$