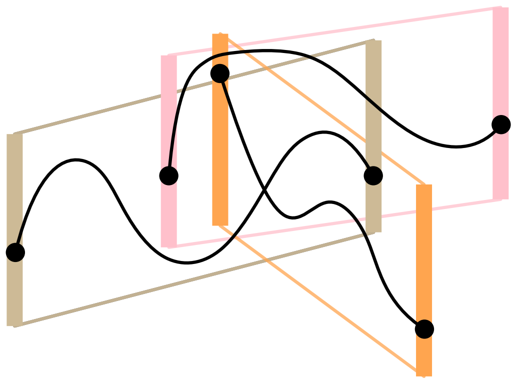
Non-crossing paths with geographic constraints

Rodrigo I. Silveira

UPC Barcelona

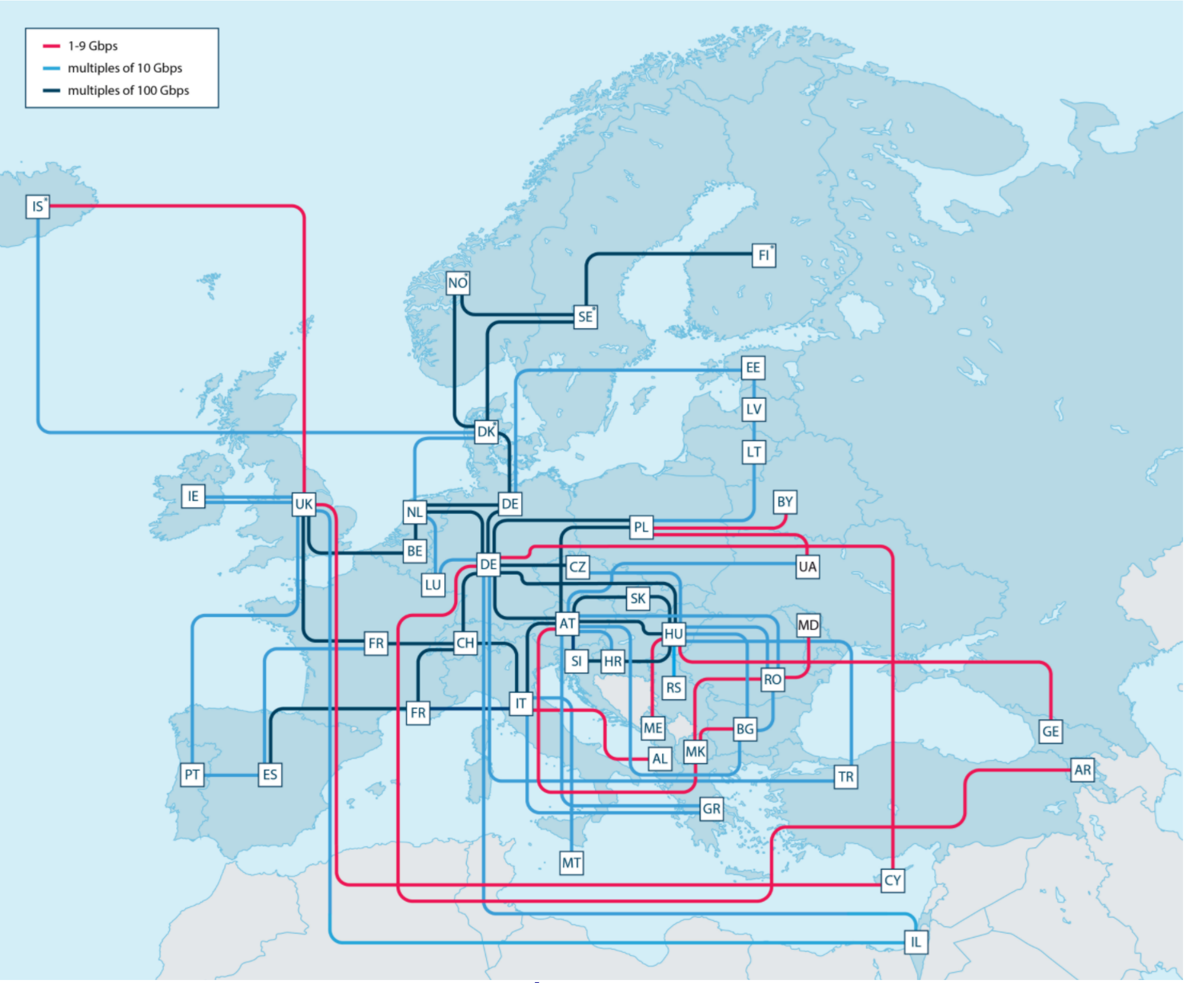


Bettina Speckmann

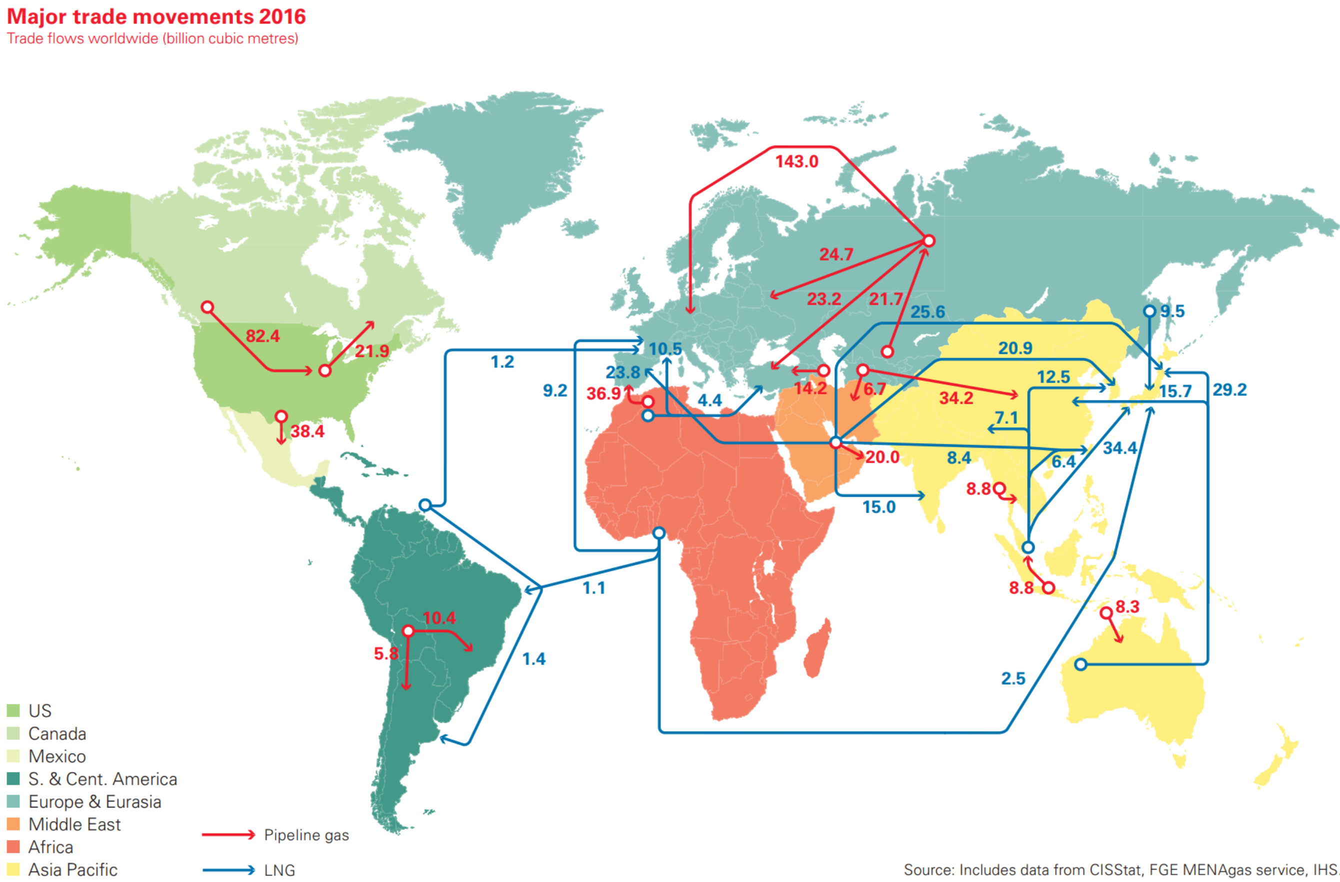
Kevin Verbeek

TU Eindhoven

Geographic networks



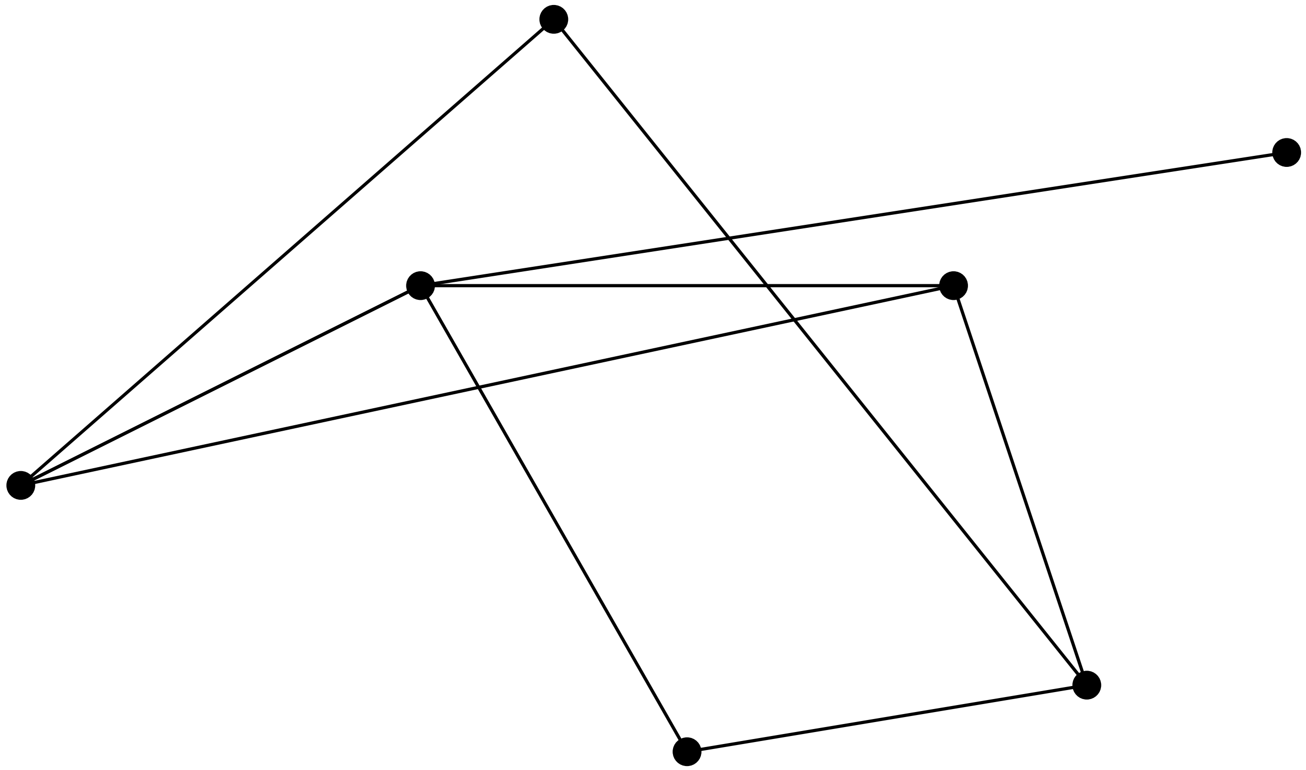
Topology map of GÉANT (pan-European research and education network) [geant.org]



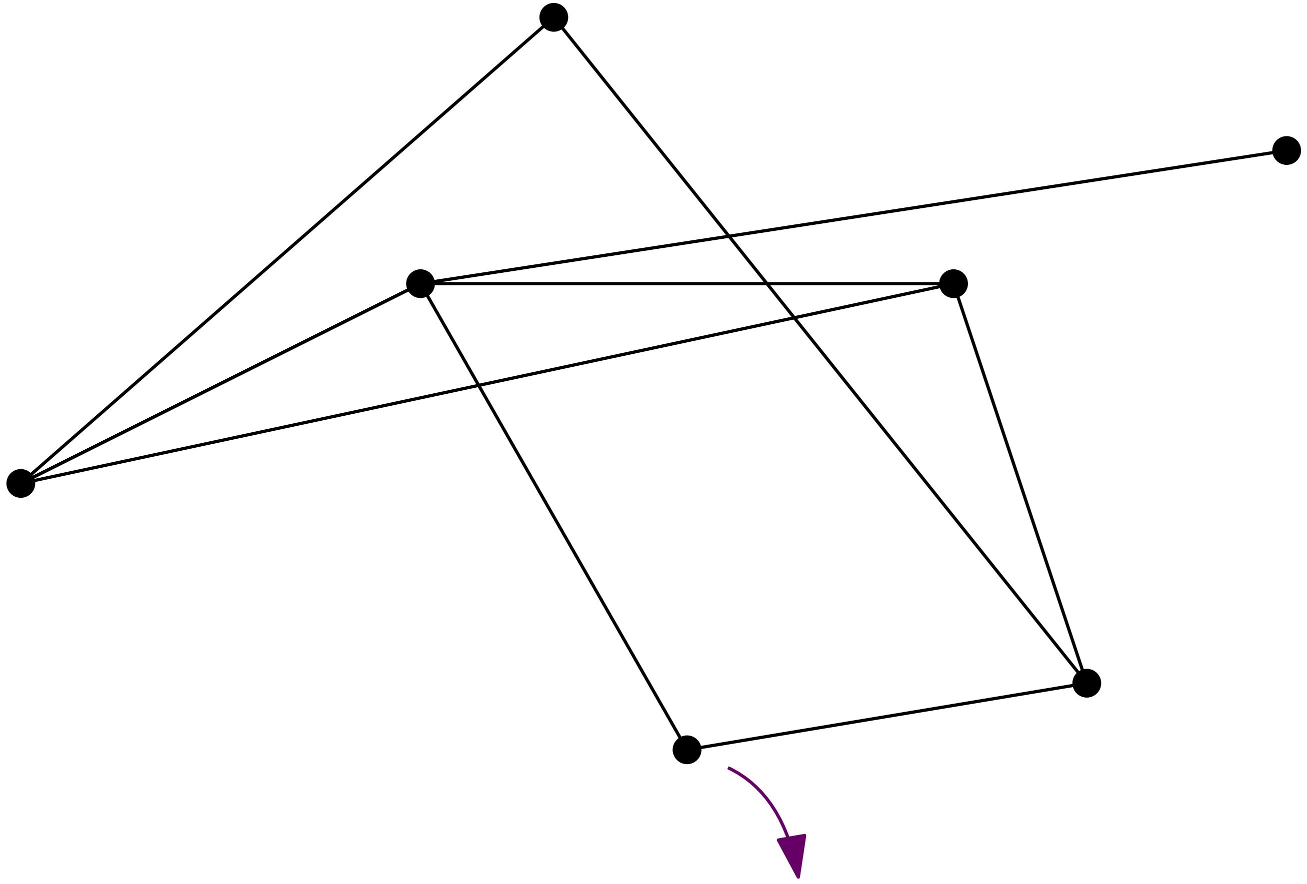
Gas trade map [BP Statistical Review of World Energy '17]

Drawing geographic networks

Drawing geographic networks

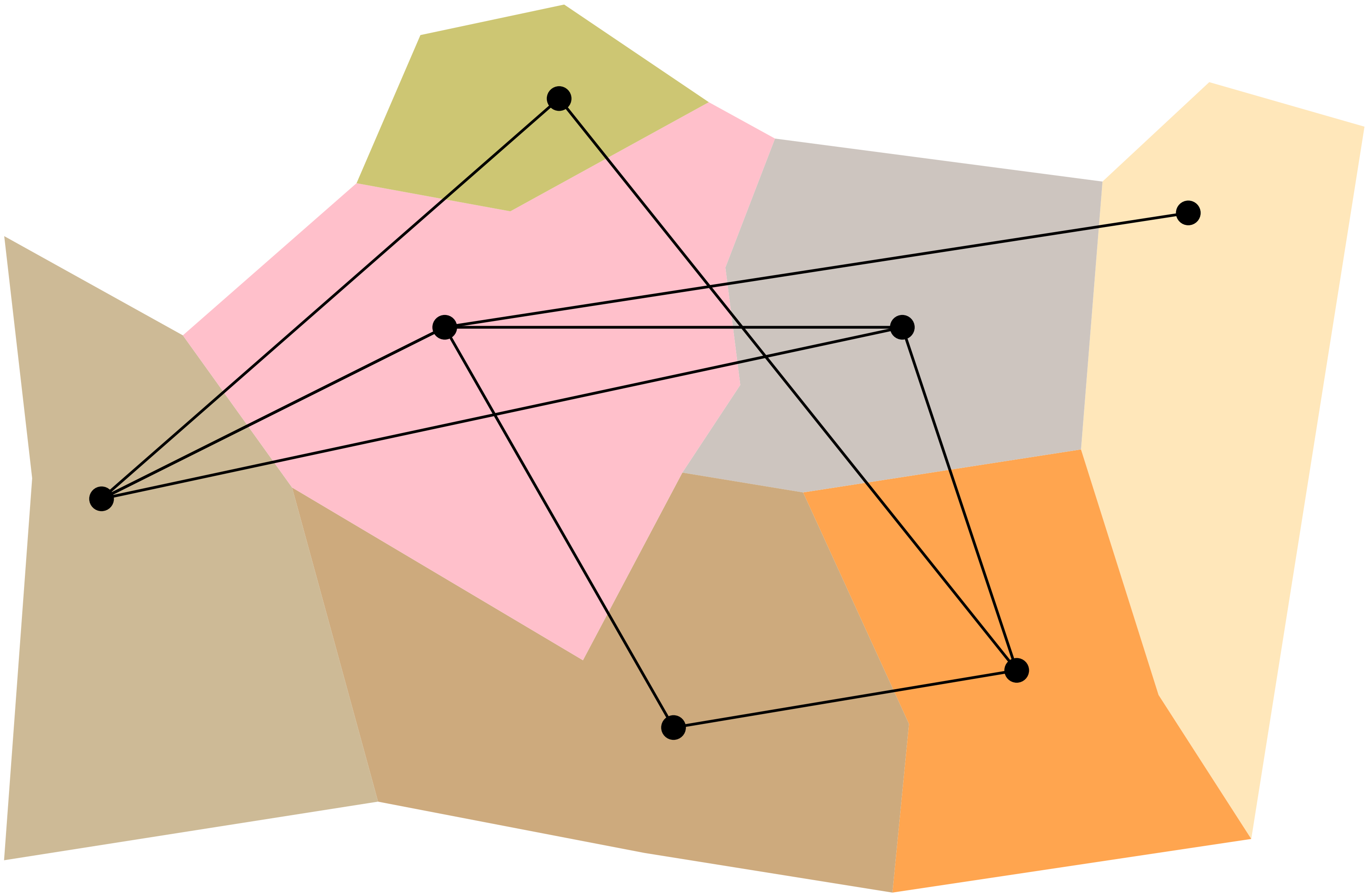


Drawing geographic networks



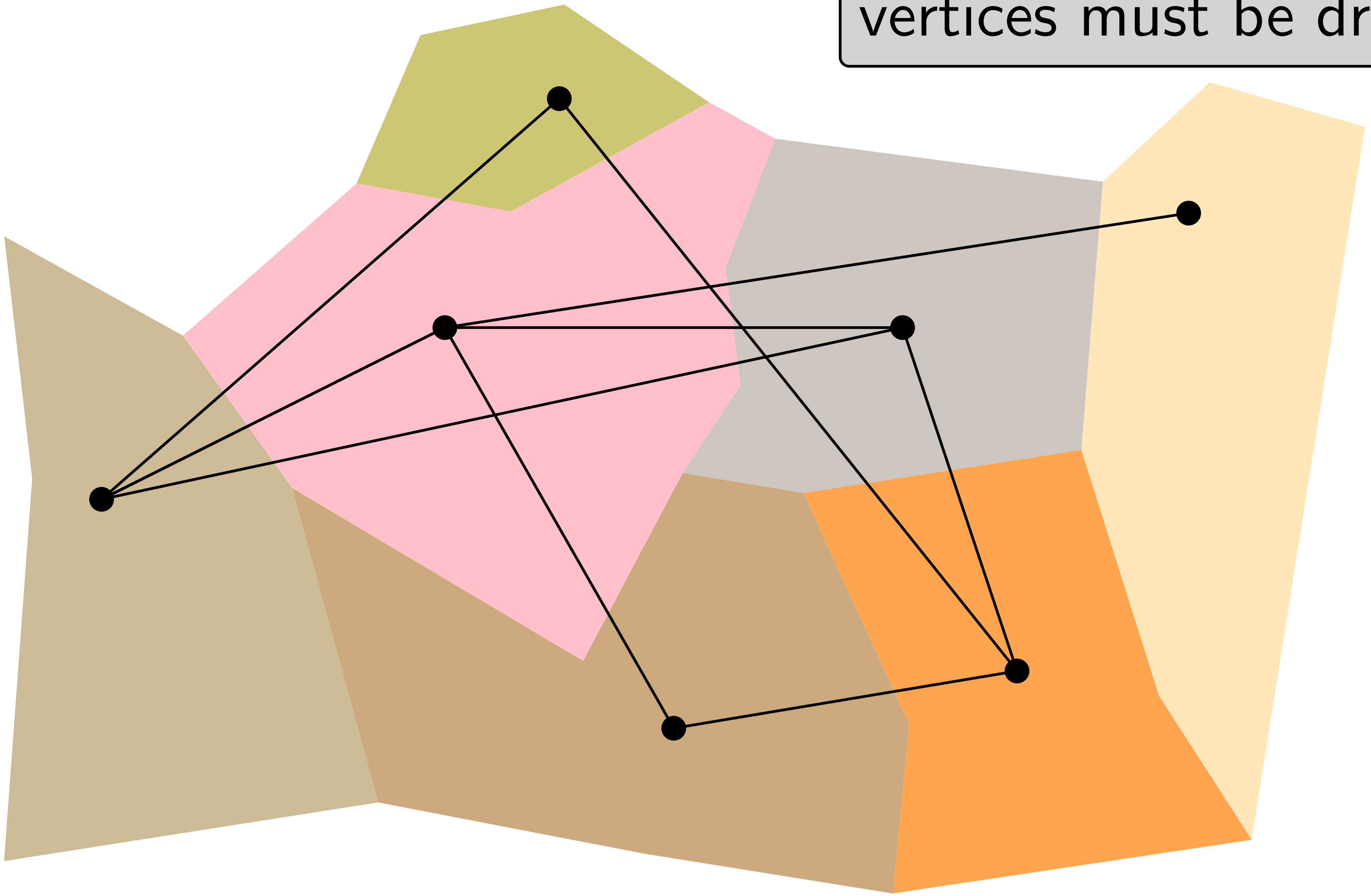
vertices: geographical regions

Drawing geographic networks

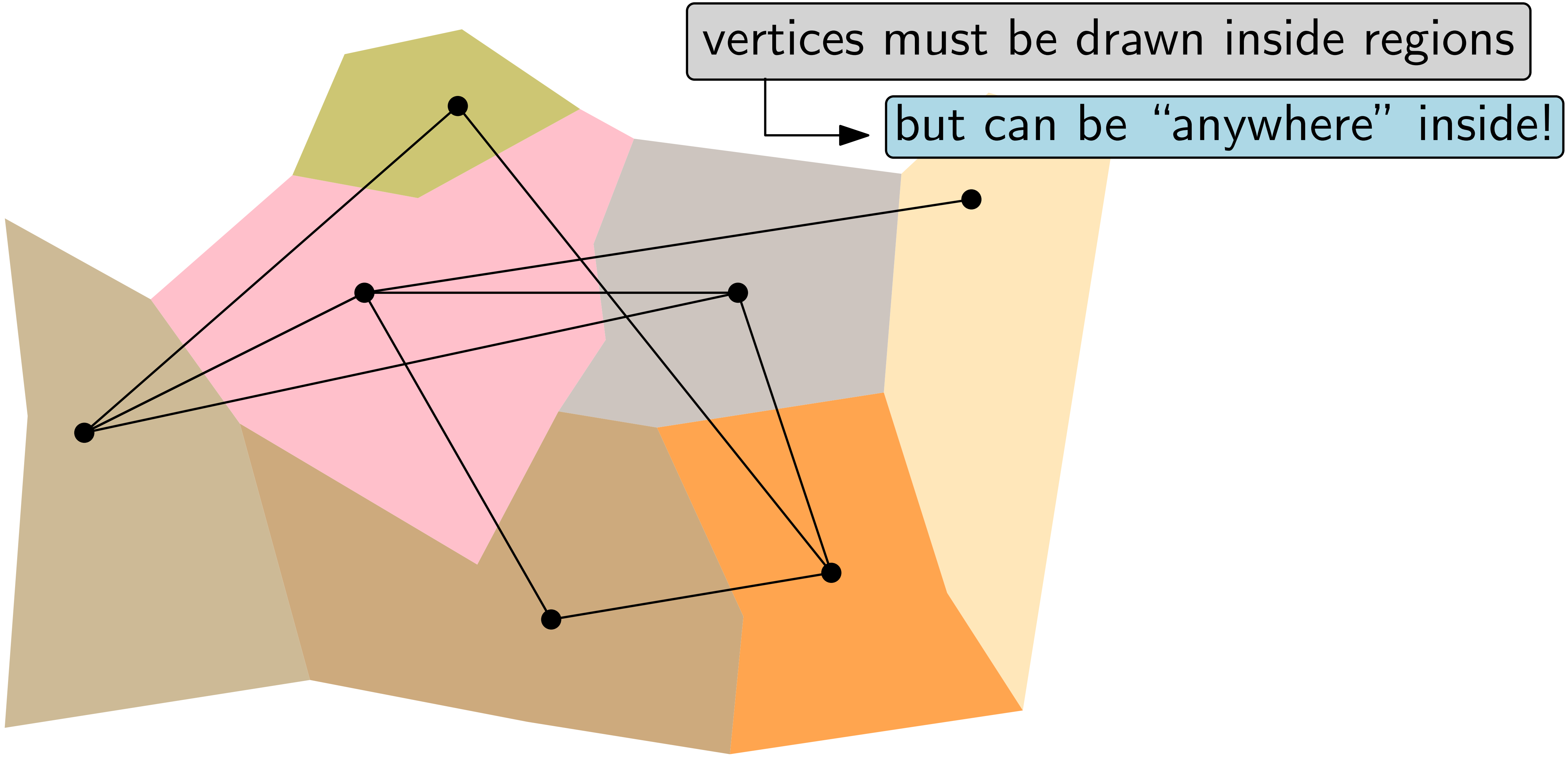


Drawing geographic networks

vertices must be drawn inside regions



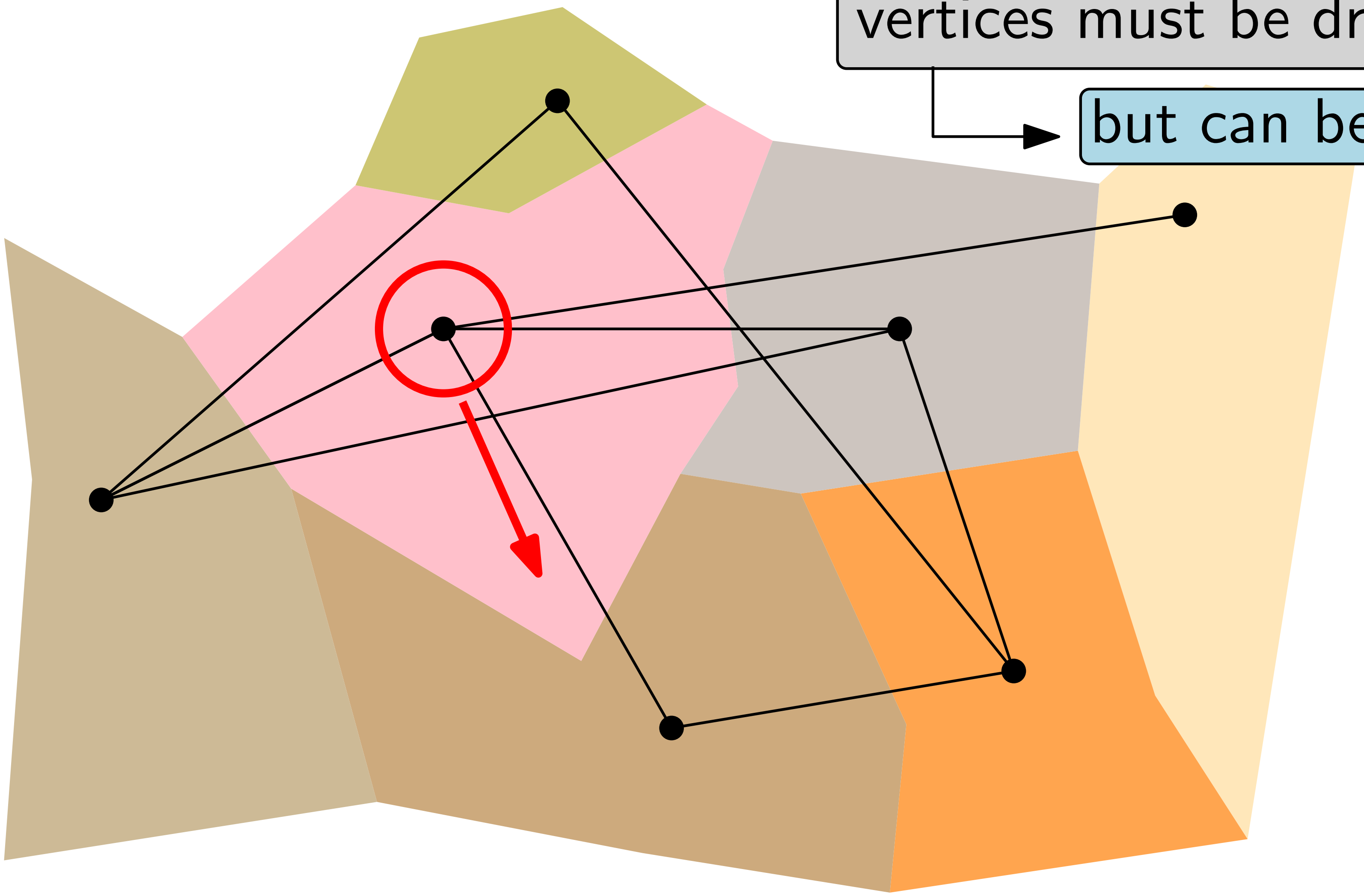
Drawing geographic networks



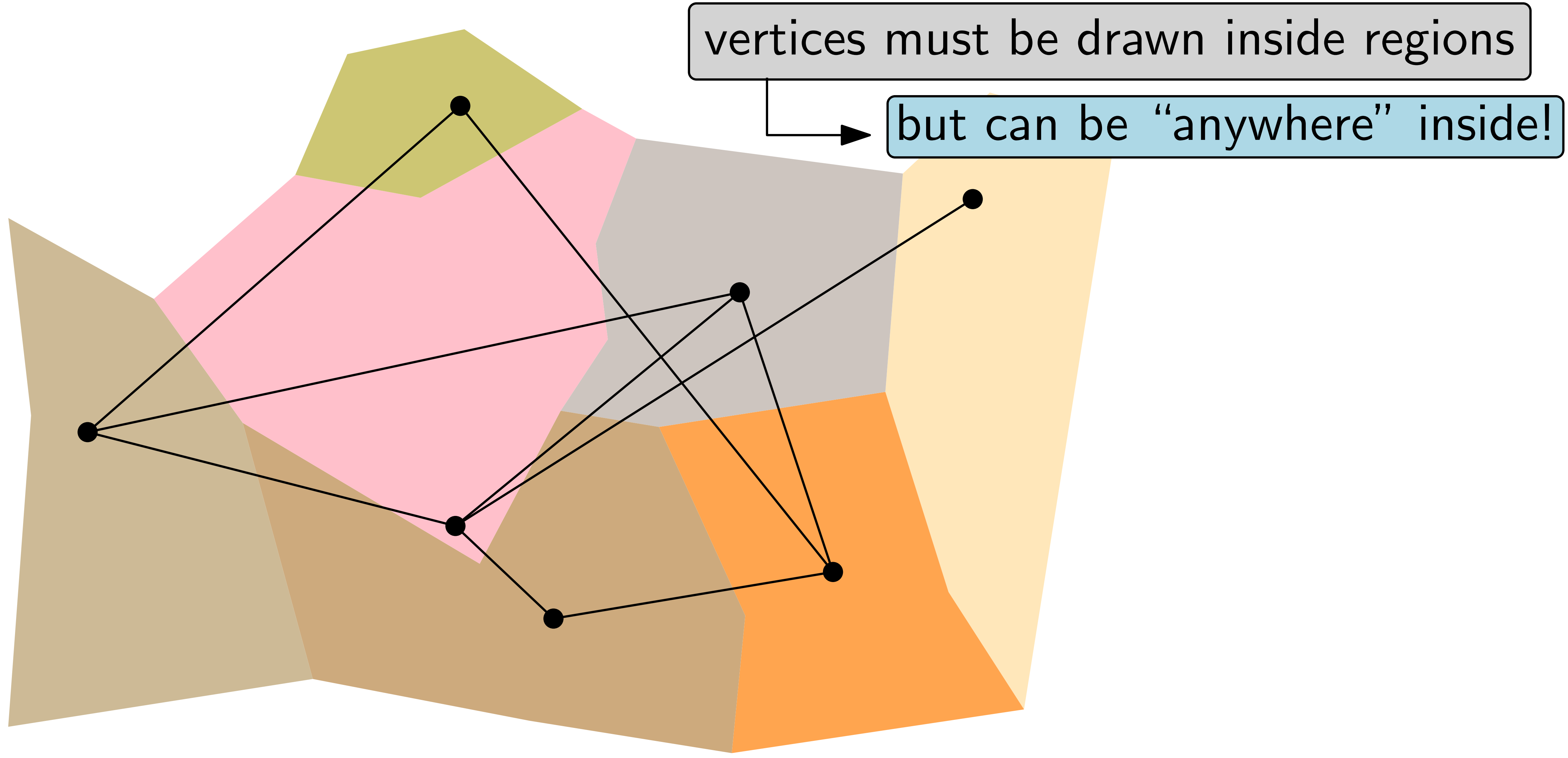
Drawing geographic networks

vertices must be drawn inside regions

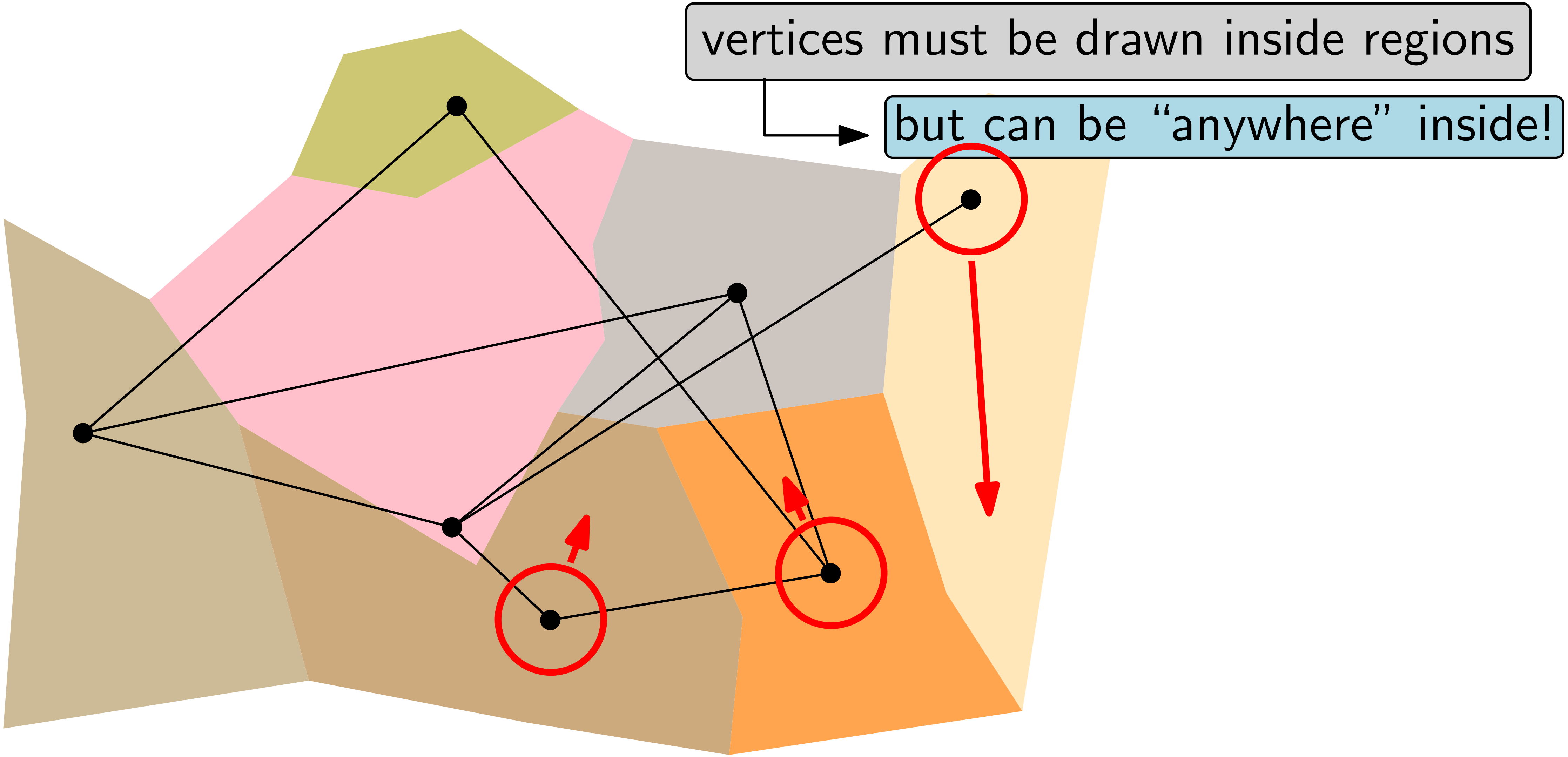
but can be "anywhere" inside!



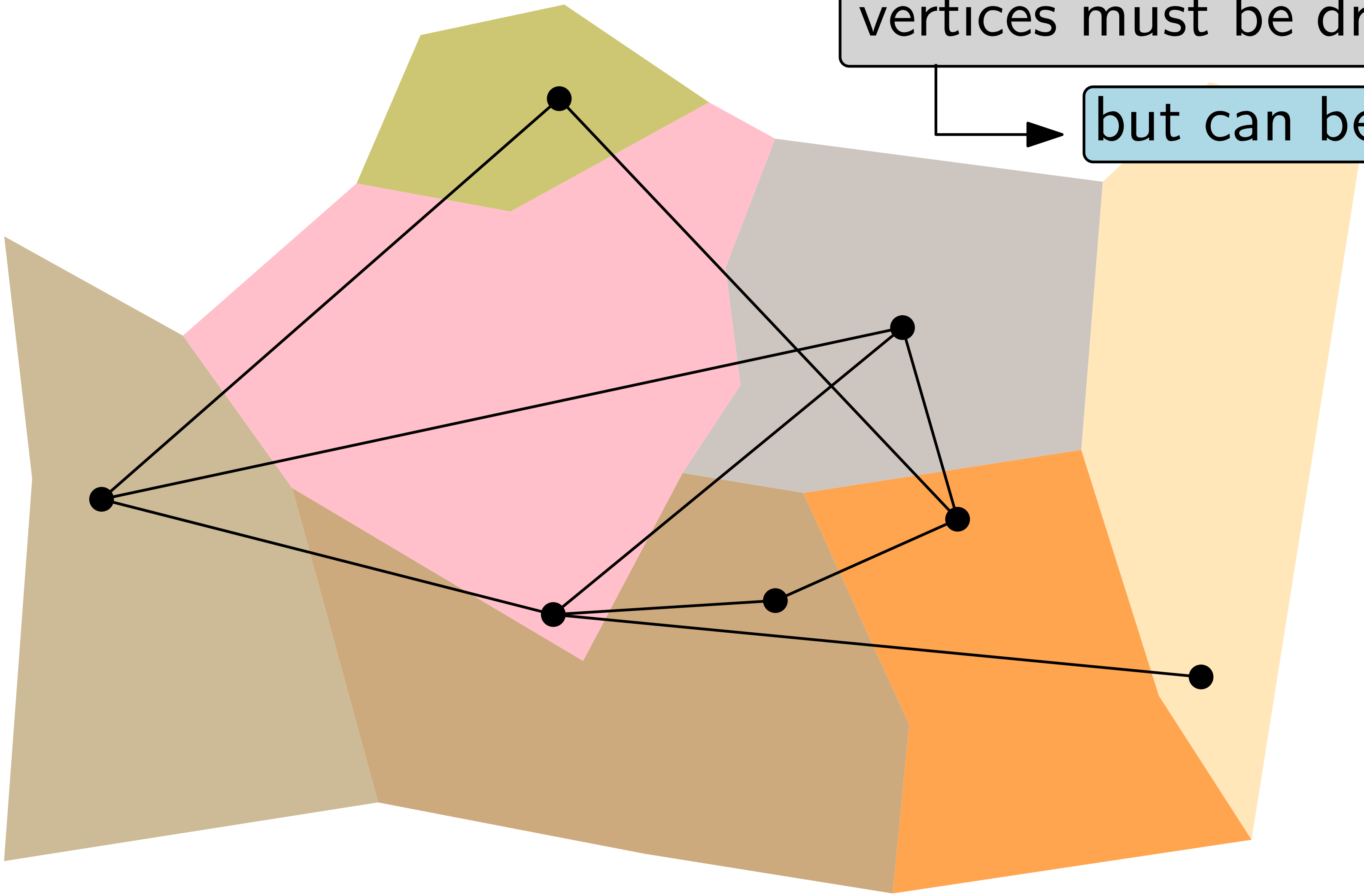
Drawing geographic networks



Drawing geographic networks



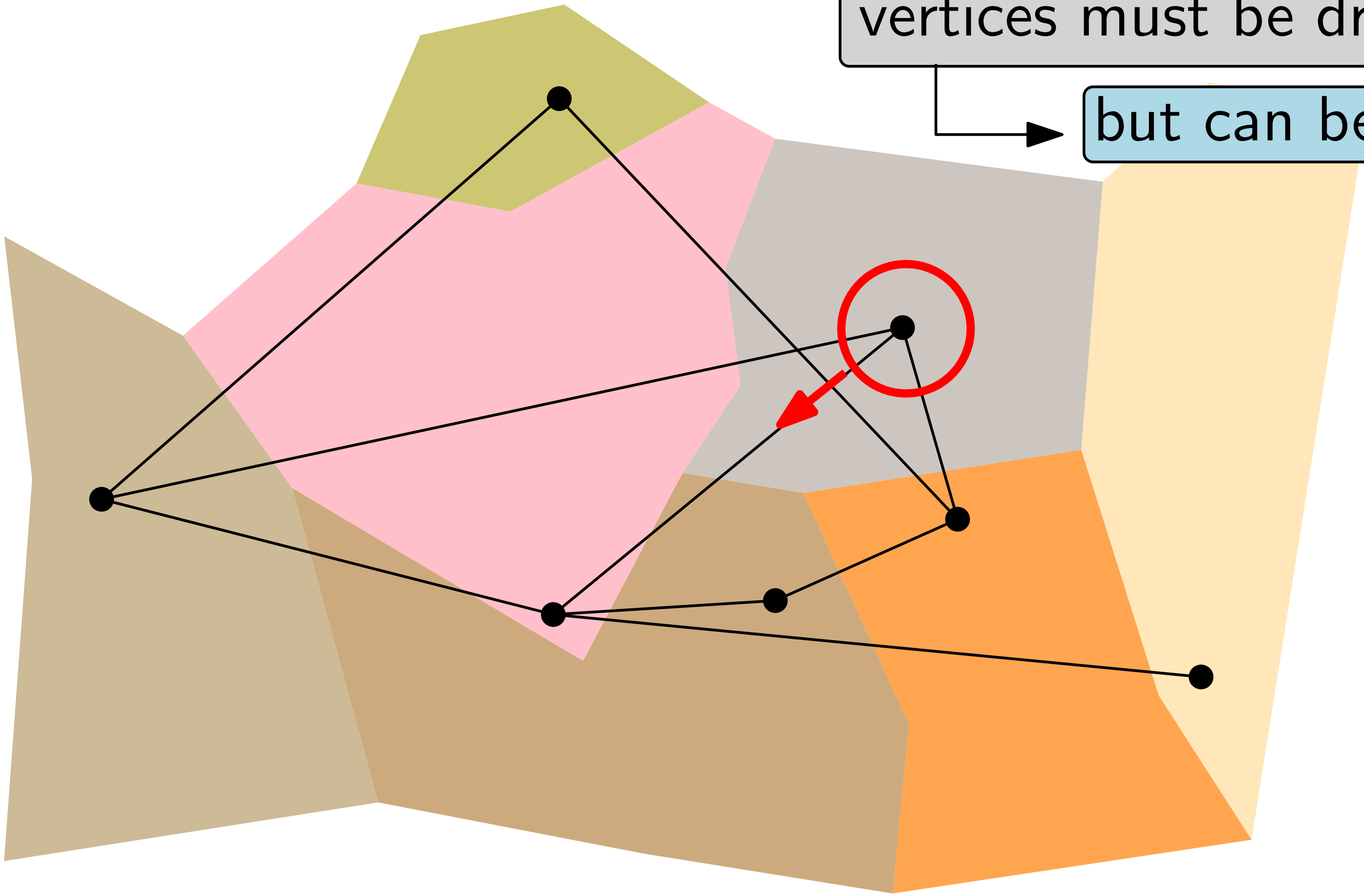
Drawing geographic networks



vertices must be drawn inside regions

but can be "anywhere" inside!

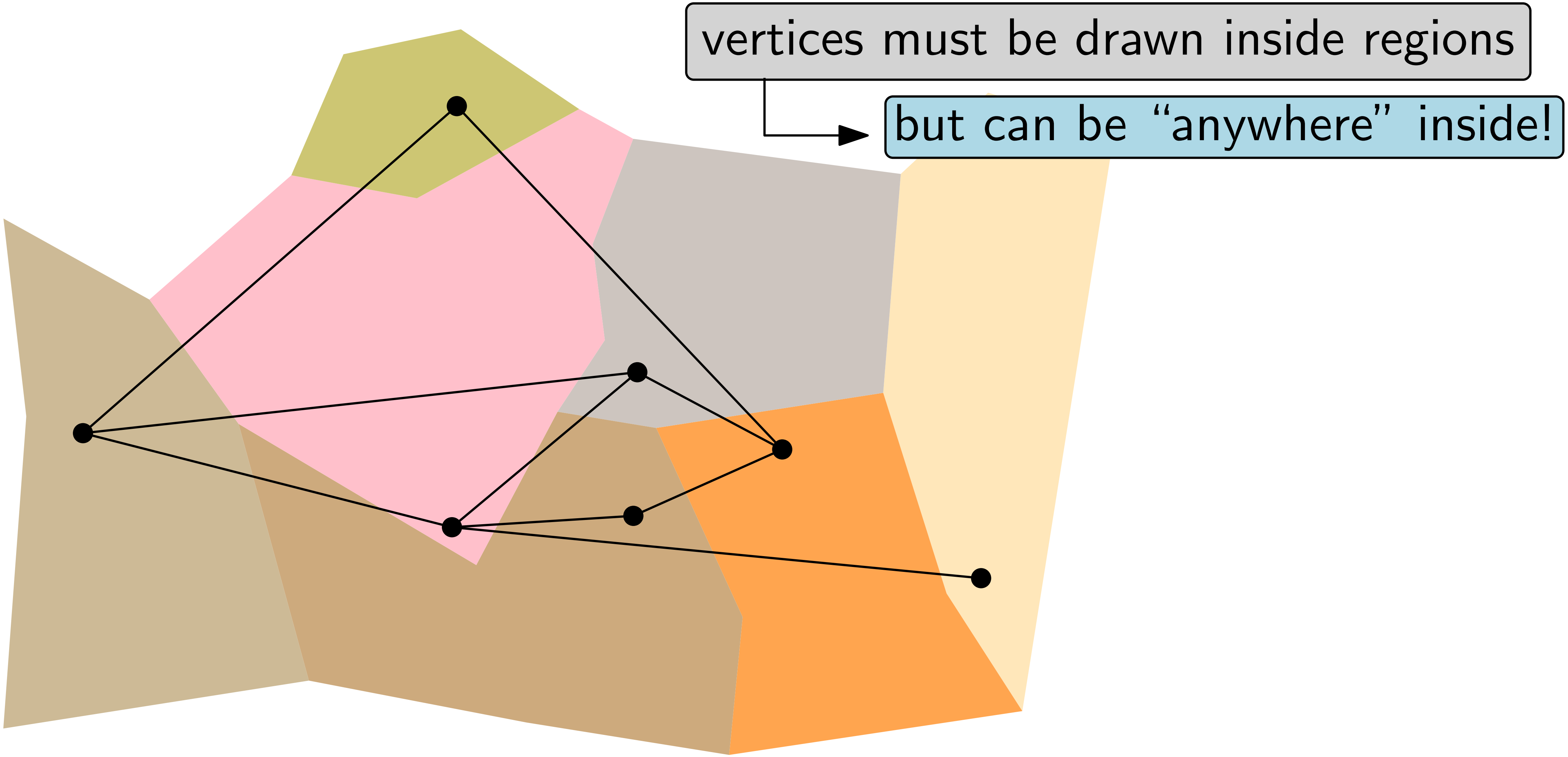
Drawing geographic networks



vertices must be drawn inside regions

but can be "anywhere" inside!

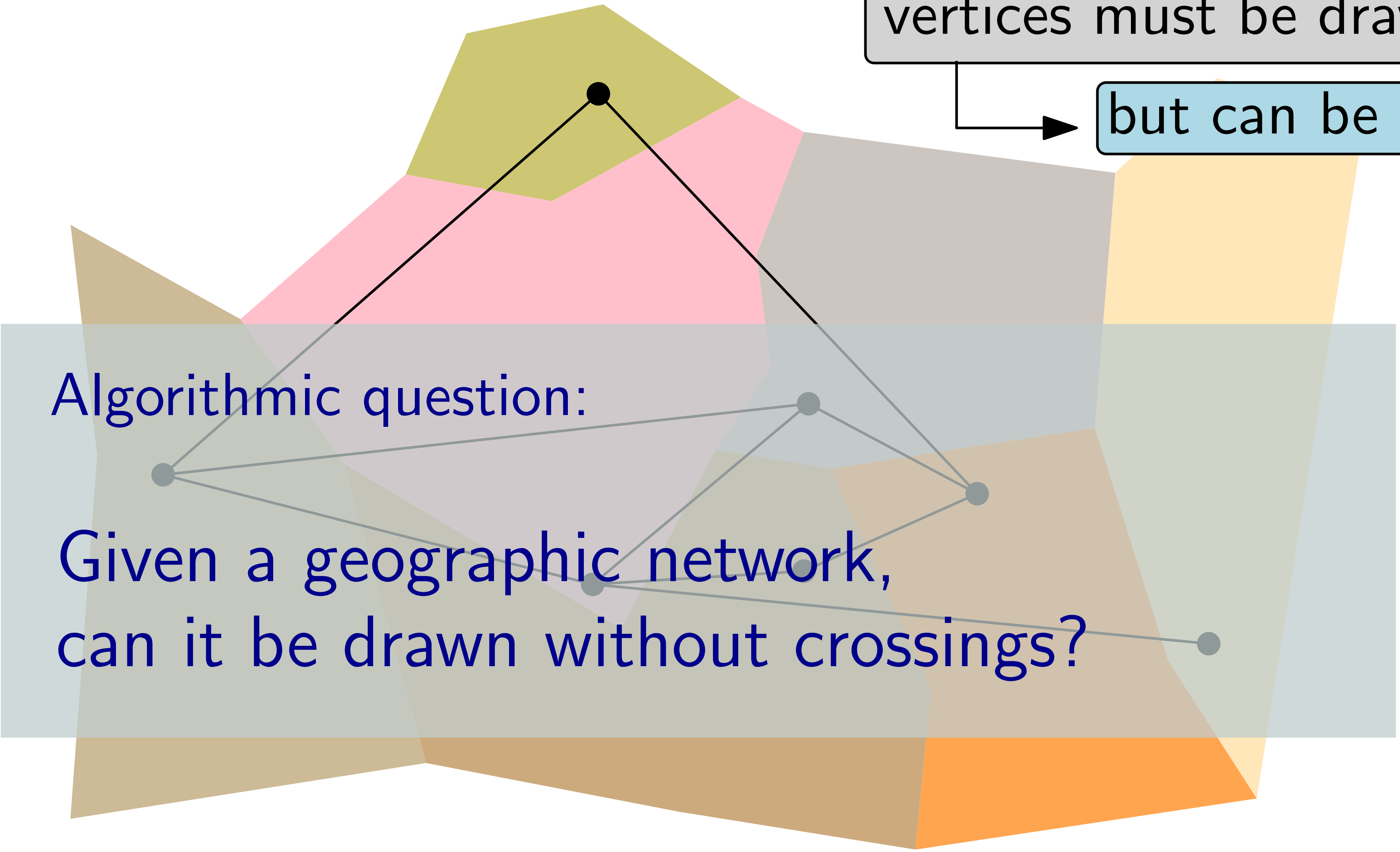
Drawing geographic networks



Drawing geographic networks

vertices must be drawn inside regions

but can be "anywhere" inside!

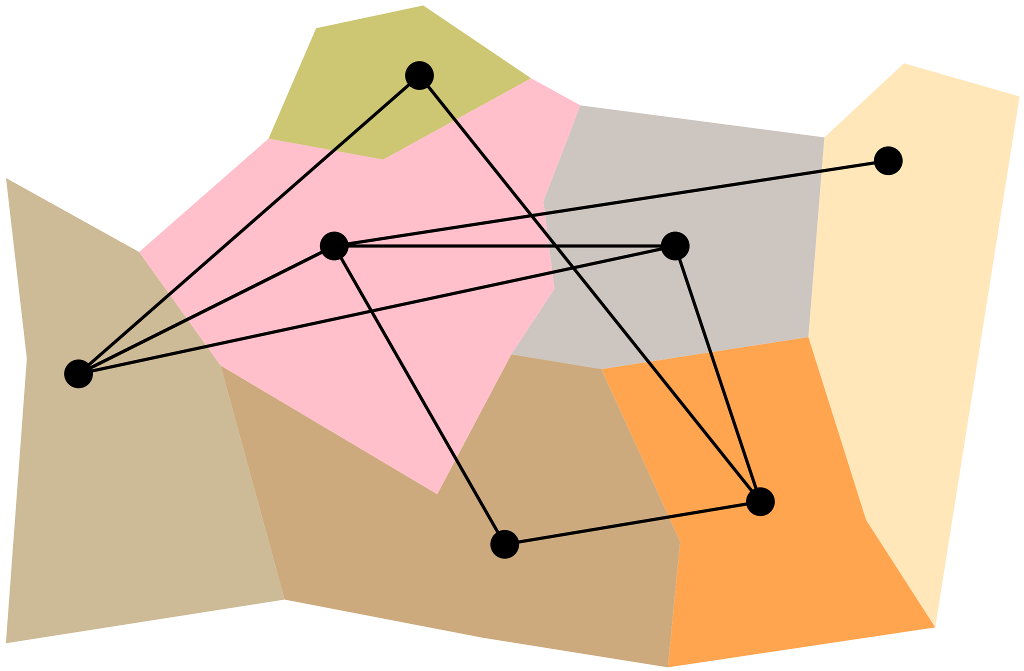


Algorithmic question:

Given a geographic network,
can it be drawn without crossings?

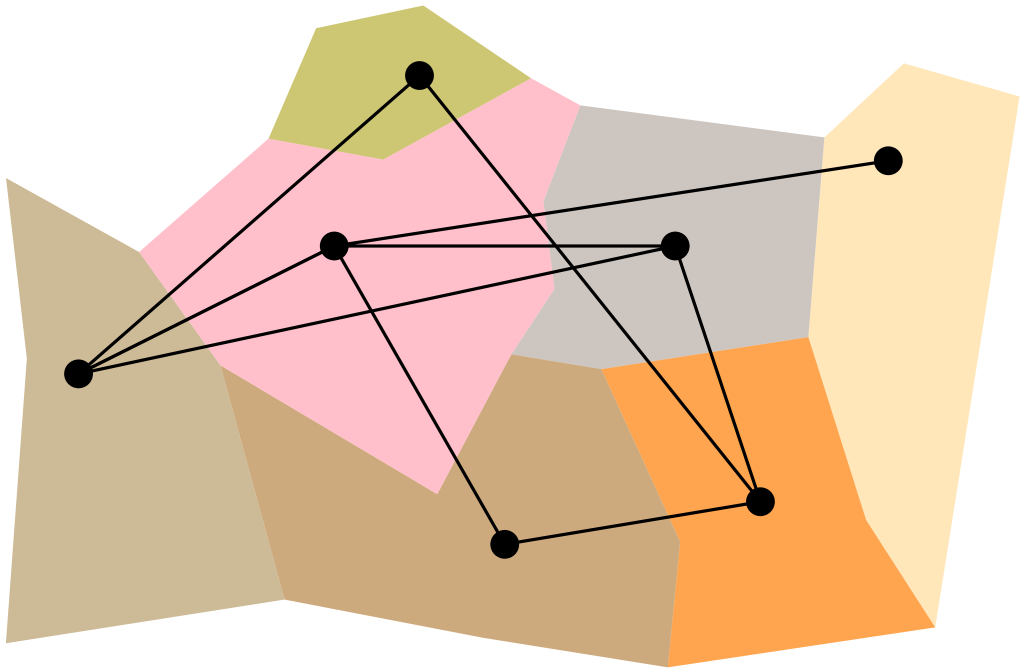
The big problem

Given a geographic network, can it be drawn without crossings?



The big problem

Given a geographic network, can it be drawn without crossings?

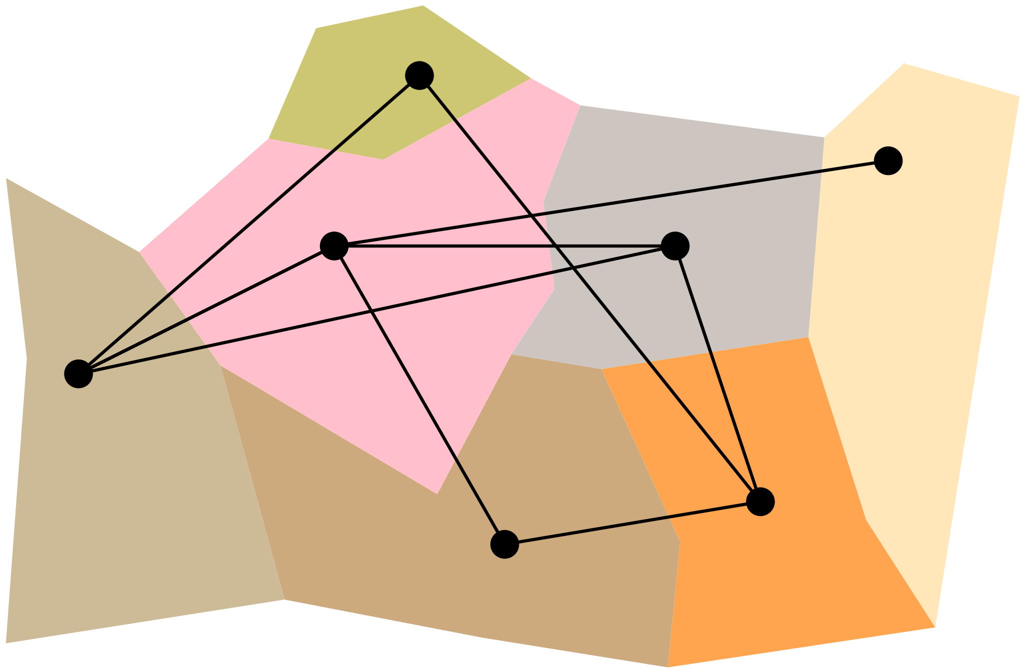


Need to precise:

- Type of graph
- Regions
- Curves to draw edges

Our problem

Given a geographic network, can it be drawn without crossings?

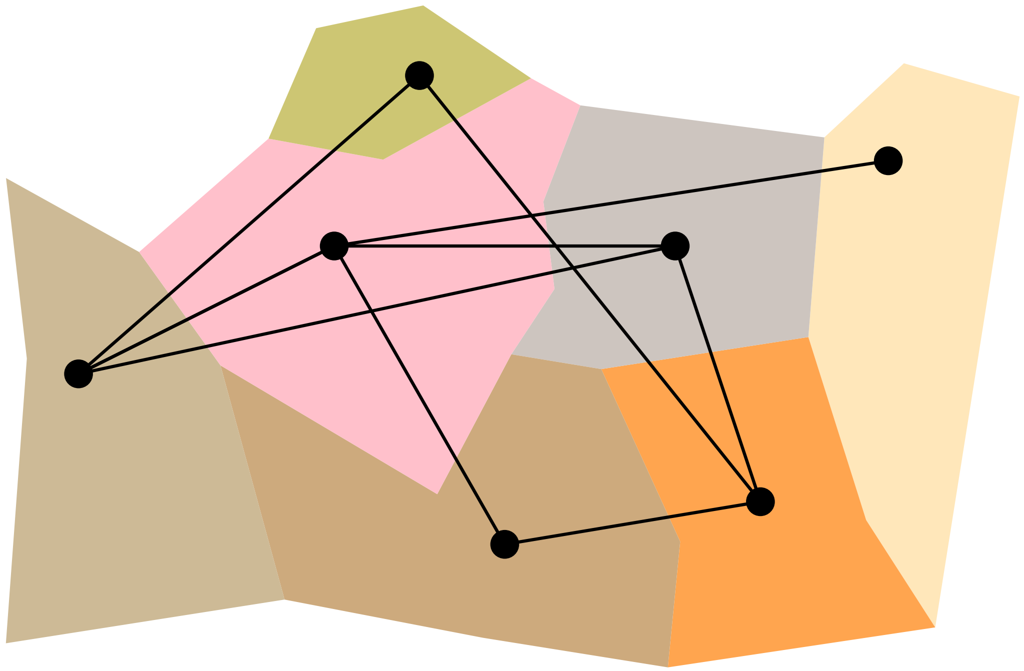


Need to precise:

- Type of graph
- Regions
- Curves to draw edges

Our problem

Given a geographic network, can it be drawn without crossings?

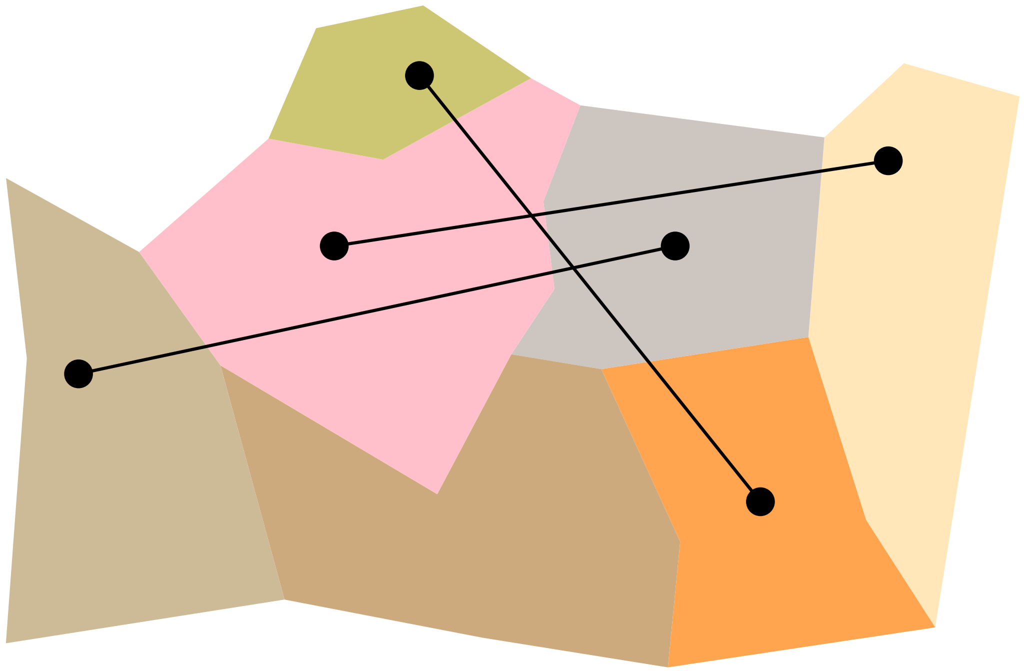


Need to precise:

- Type of graph ← matching
- Regions
- Curves to draw edges

Our problem

Given a geographic network, can it be drawn without crossings?

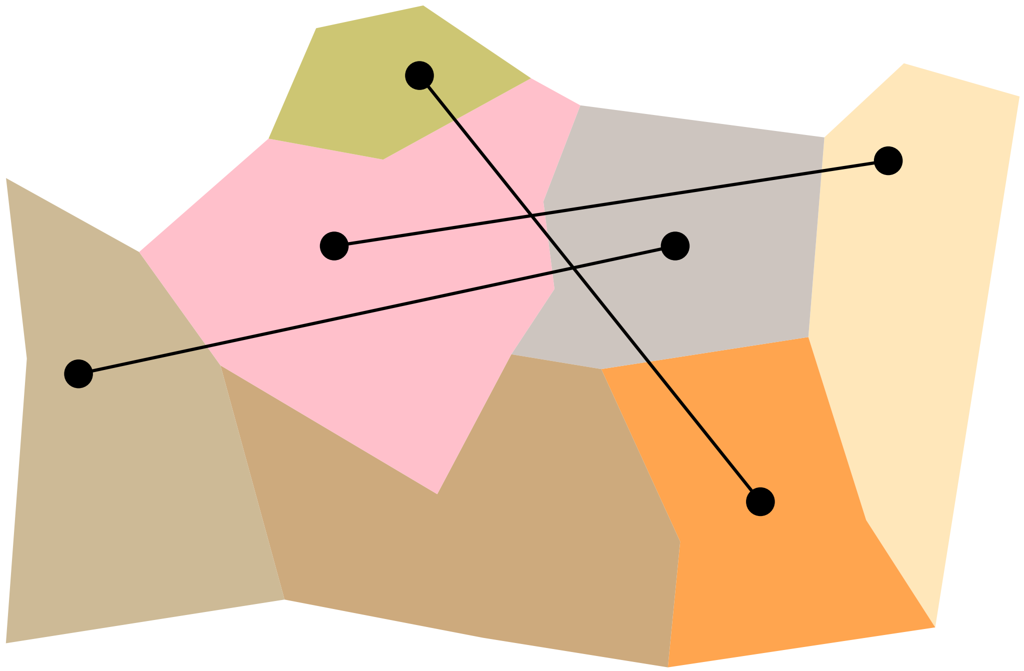


Need to precise:

- Type of graph ← matching
- Regions
- Curves to draw edges

Our problem

Given a geographic network, can it be drawn without crossings?

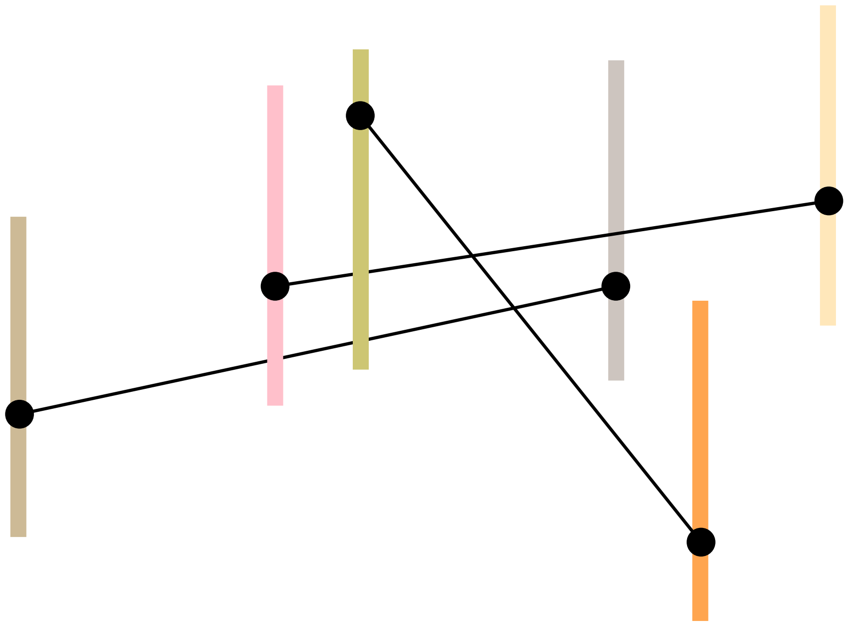


Need to precise:

- Type of graph $\leftarrow$ matching
- Regions $\leftarrow$ unit vertical segments
- Curves to draw edges

Our problem

Given a geographic network, can it be drawn without crossings?

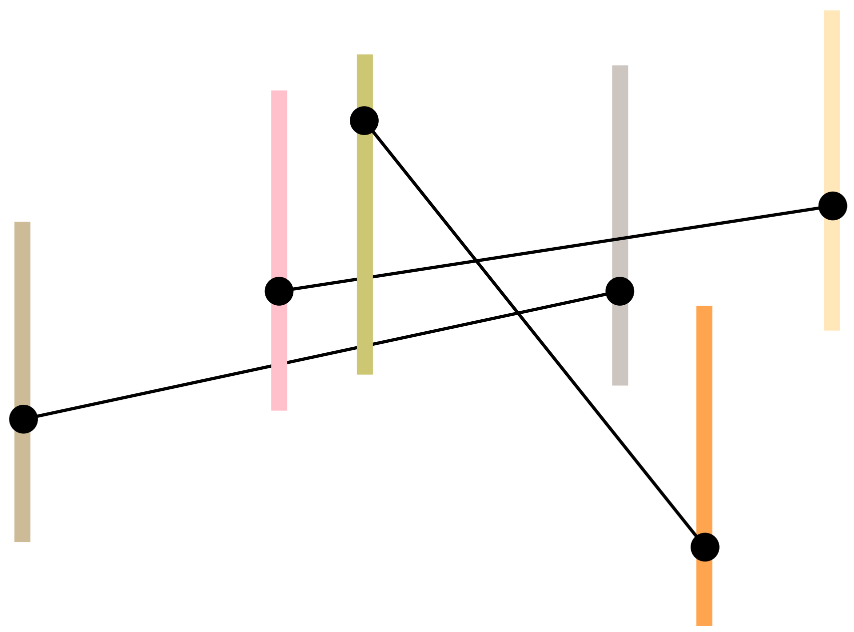


Need to precise:

- Type of graph $\leftarrow$ matching
- Regions $\leftarrow$ unit vertical segments
- Curves to draw edges

Our problem

Given a geographic network, can it be drawn without crossings?

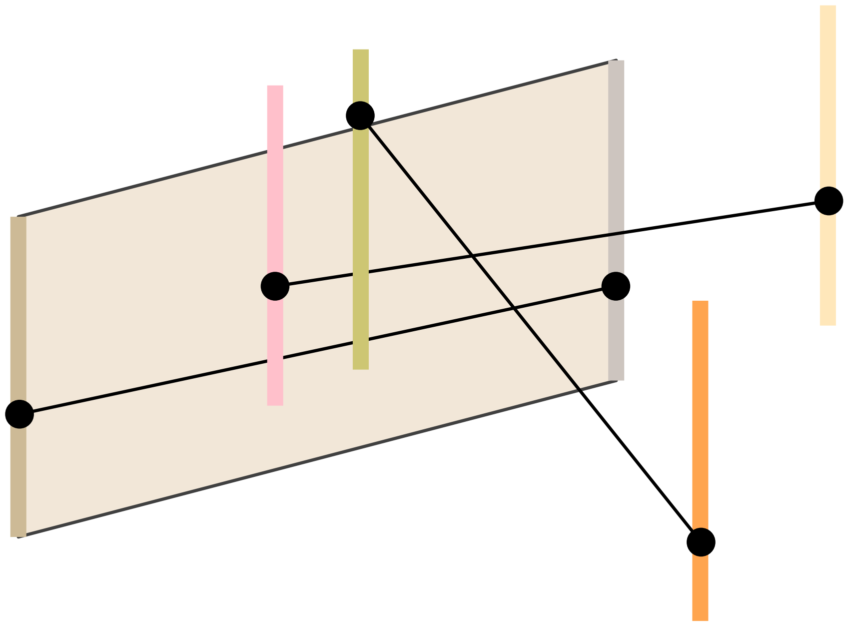


Need to precise:

- Type of graph ← matching
- Regions ← unit vertical segments
- Curves to draw edges
  ← drawn inside convex hulls of edge vertical segments (*tubes*)

Our problem

Given a geographic network, can it be drawn without crossings?



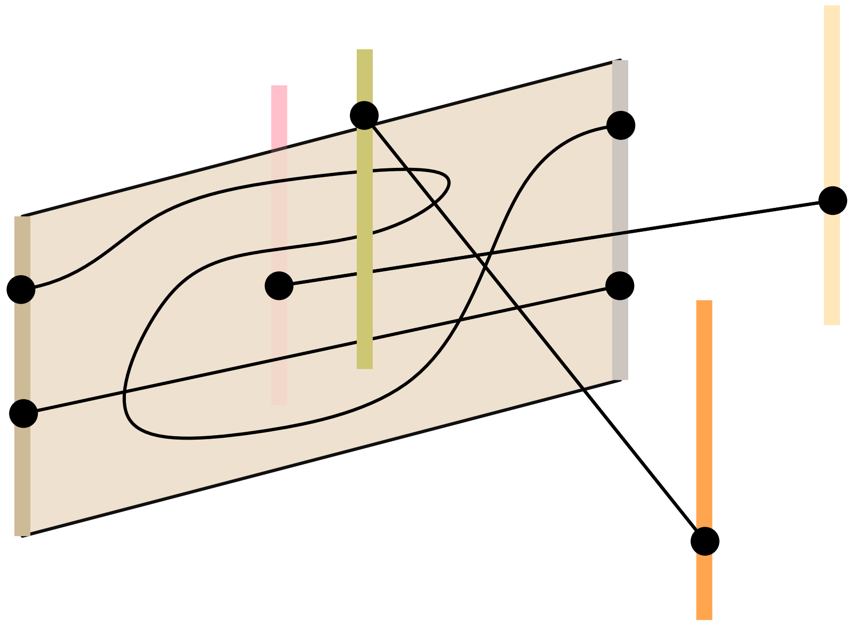
Need to precise:

- Type of graph ← matching
- Regions ← unit vertical segments
- Curves to draw edges

  ← drawn inside convex hulls of edge vertical segments (*tubes*)

Our problem

Given a geographic network, can it be drawn without crossings?

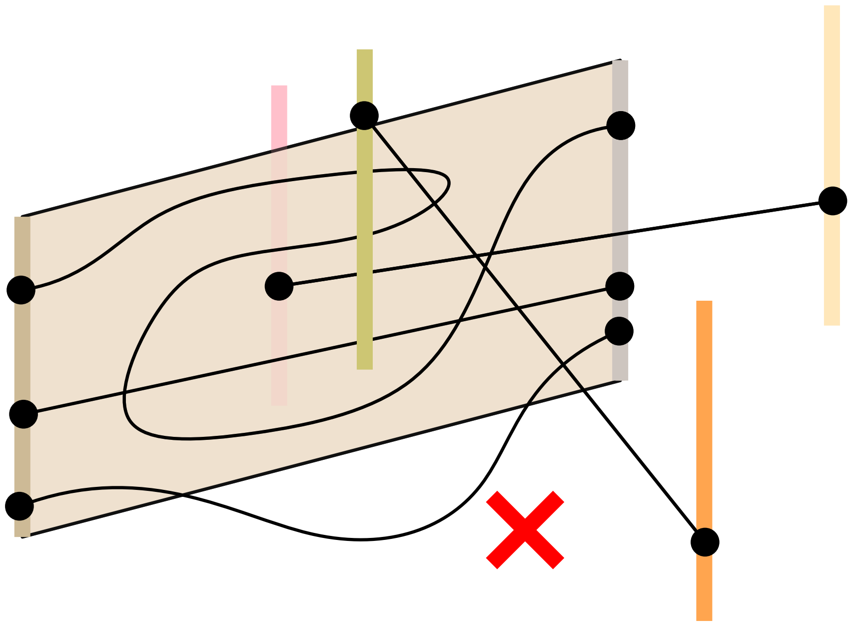


Need to precise:

- Type of graph ← matching
- Regions ← unit vertical segments
- Curves to draw edges
  ← drawn inside convex hulls of edge vertical segments (*tubes*)

Our problem

Given a geographic network, can it be drawn without crossings?

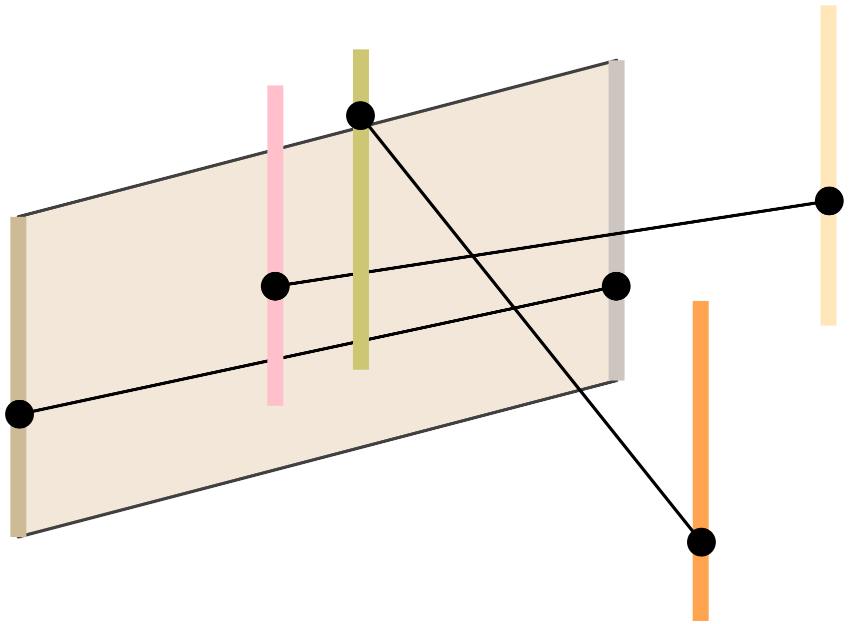


Need to precise:

- Type of graph ← matching
- Regions ← unit vertical segments
- Curves to draw edges
  ← drawn inside convex hulls of edge vertical segments (*tubes*)

Our problem

Given a geographic network, can it be drawn without crossings?

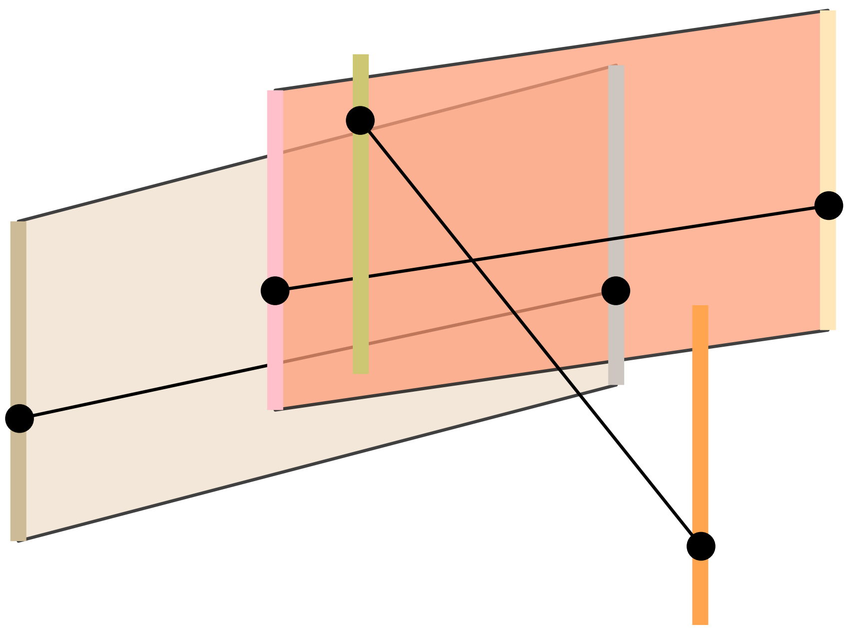


Need to precise:

- Type of graph $\leftarrow$ matching
- Regions $\leftarrow$ unit vertical segments
- Curves to draw edges
  $\leftarrow$ drawn inside convex hulls of edge vertical segments (*tubes*)

Our problem

Given a geographic network, can it be drawn without crossings?

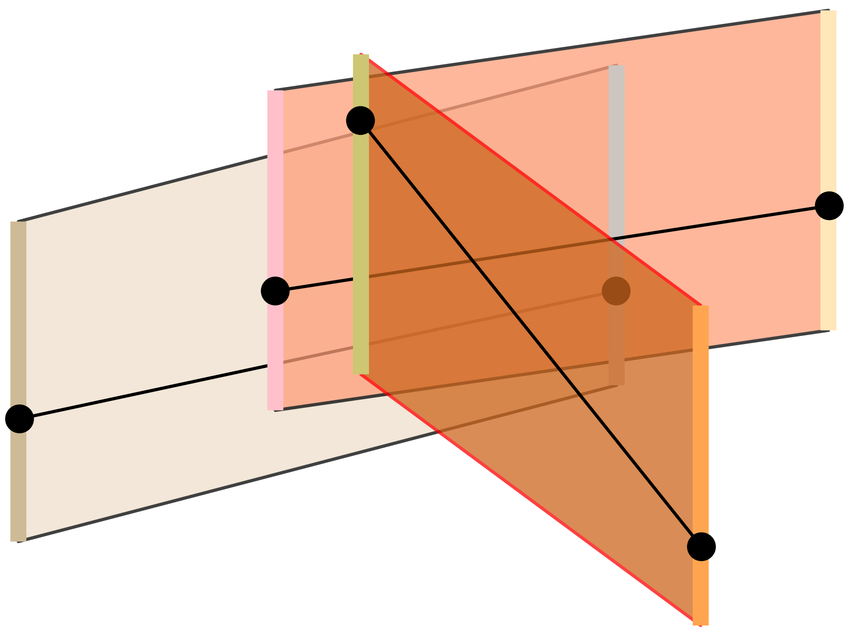


Need to precise:

- Type of graph ← matching
- Regions ← unit vertical segments
- Curves to draw edges
  ← drawn inside convex hulls of edge vertical segments (*tubes*)

Our problem

Given a geographic network, can it be drawn without crossings?

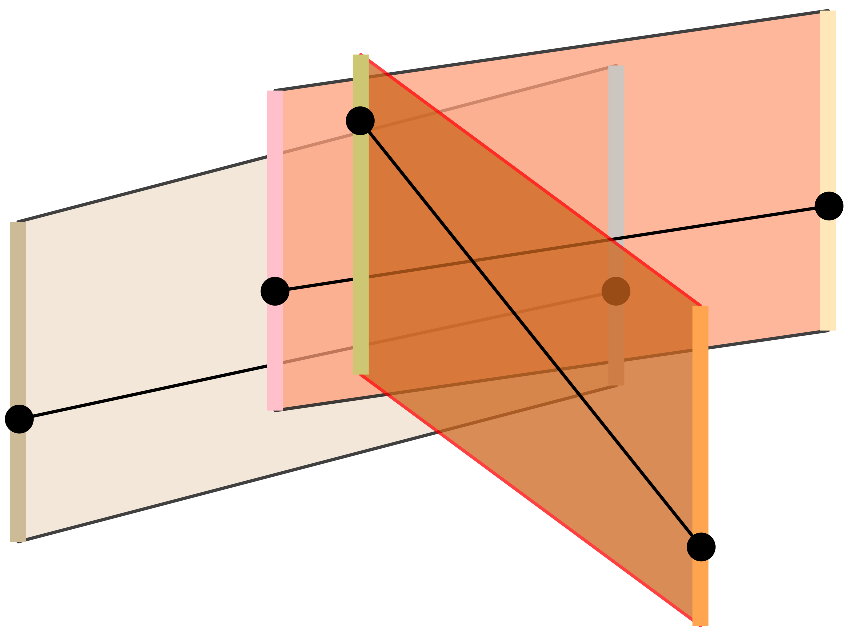


Need to precise:

- Type of graph ← matching
- Regions ← unit vertical segments
- Curves to draw edges
  ← drawn inside convex hulls of edge vertical segments (*tubes*)

Our problem

Given a geographic network, can it be drawn without crossings?



Need to precise:

- Type of graph ← matching
- Regions ← unit vertical segments
- Curves to draw edges

  ← drawn inside convex hulls of edge vertical segments (*tubes*)

  ← three ways to draw edges

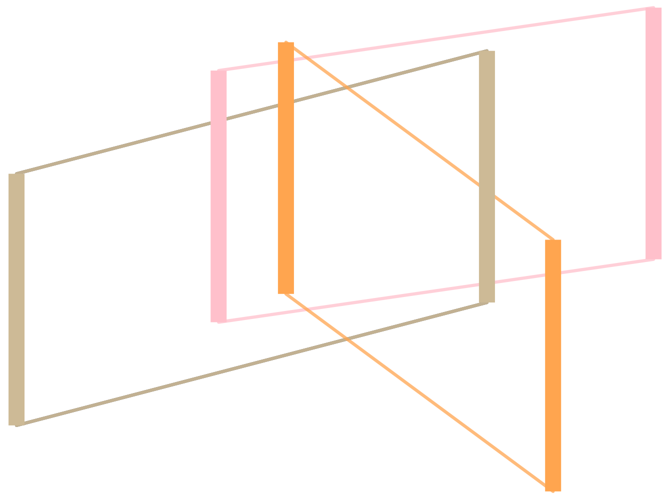
Our problem

Given a matching, where the endpoint of each edge must lie on a unit vertical segment, can it be drawn without crossings?

Our problem

Given a matching, where the endpoint of each edge must lie on a unit vertical segment, can it be drawn without crossings?

Three ways to draw edges inside tubes:

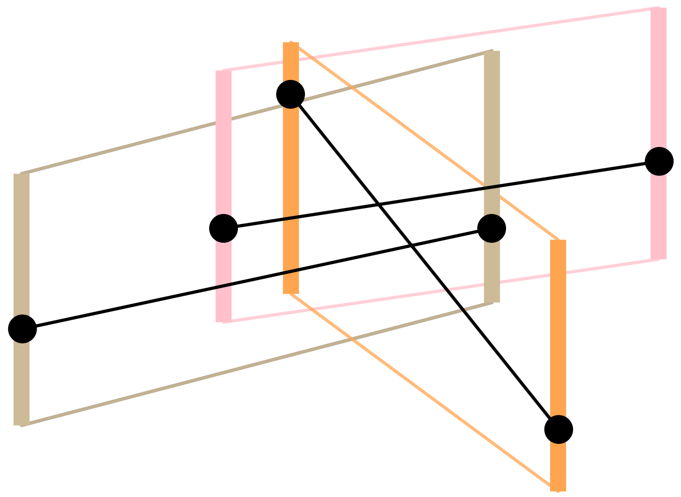


Our problem

Given a matching, where the endpoint of each edge must lie on a unit vertical segment, can it be drawn without crossings?

Three ways to draw edges inside tubes:

straight-line segment

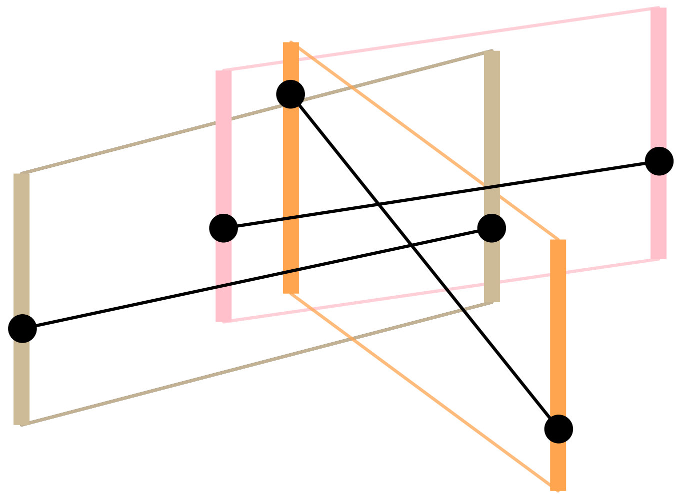


Our problem

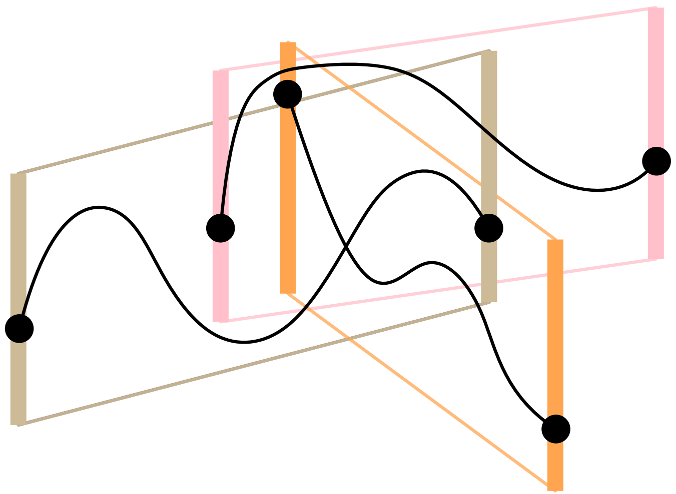
Given a matching, where the endpoint of each edge must lie on a unit vertical segment, can it be drawn without crossings?

Three ways to draw edges inside tubes:

straight-line segment



(x-)monotone paths

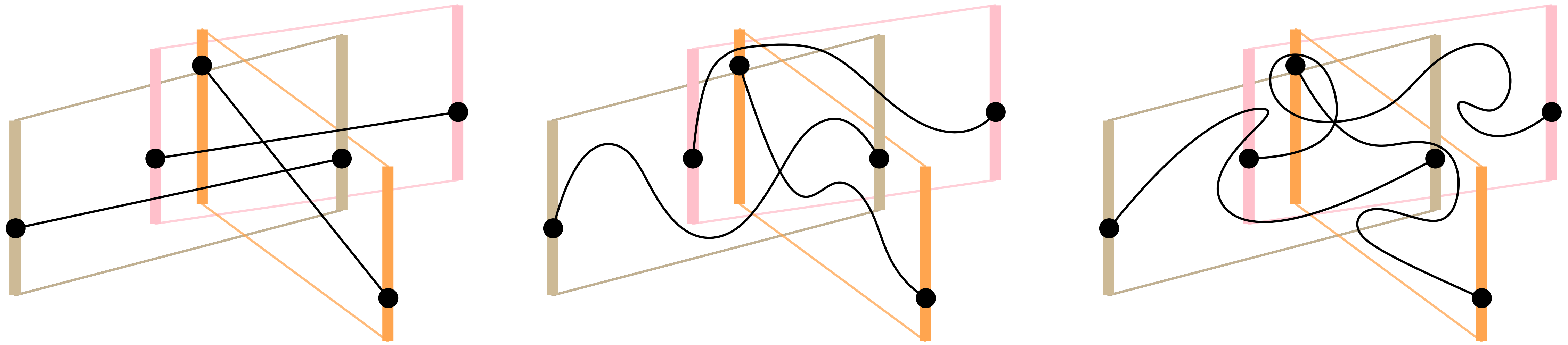


Our problem

Given a matching, where the endpoint of each edge must lie on a unit vertical segment, can it be drawn without crossings?

Three ways to draw edges inside tubes:

straight-line segment | (x-)monotone paths | arbitrary paths

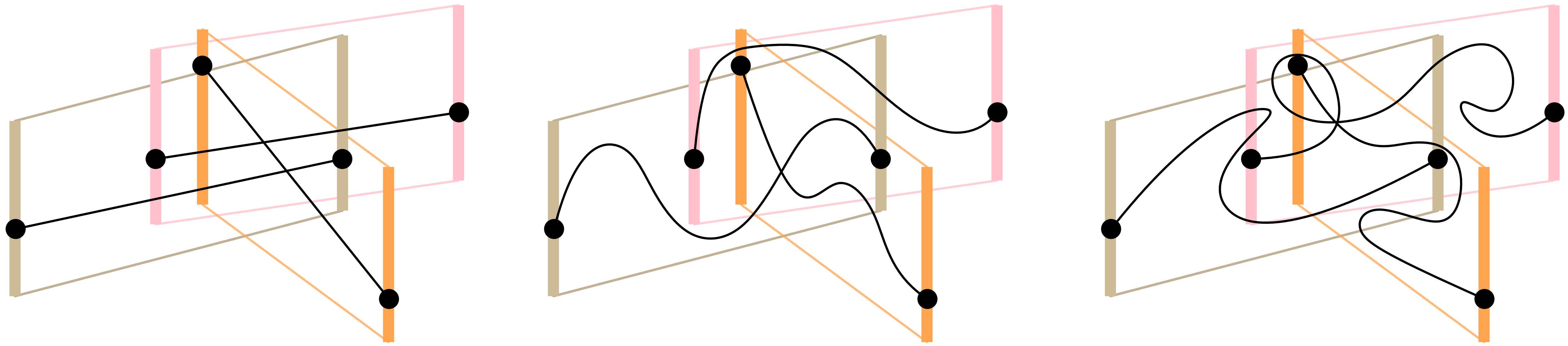


Our problem

Given a matching, where the endpoint of each edge must lie on a unit vertical segment, can it be drawn without crossings?

Three ways to draw edges inside tubes:

straight-line segment | (x-)monotone paths | arbitrary paths

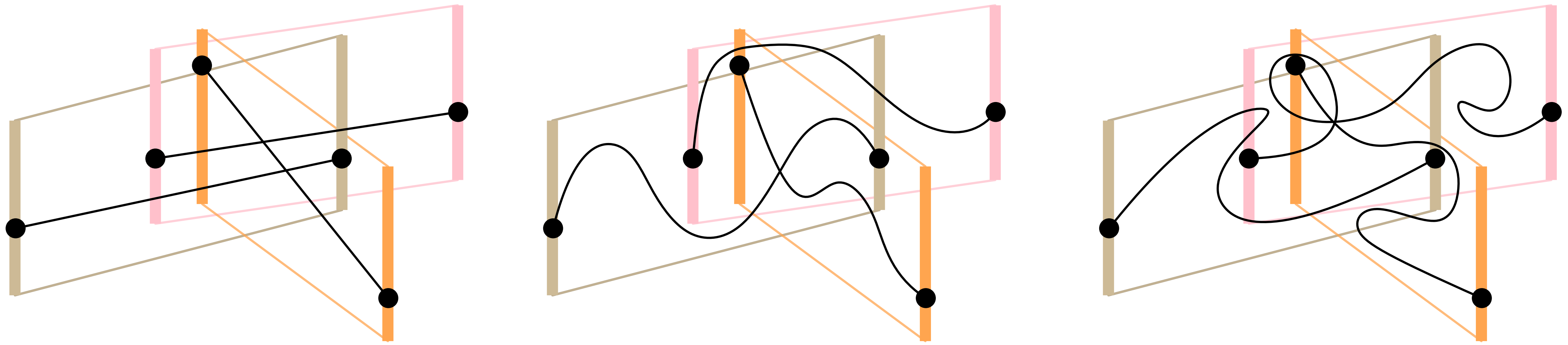


Recall: endpoints can be anywhere on the vertical line segments

Our results

Given a matching, where the endpoint of each edge must lie on a unit vertical segment, can it be drawn without crossings?

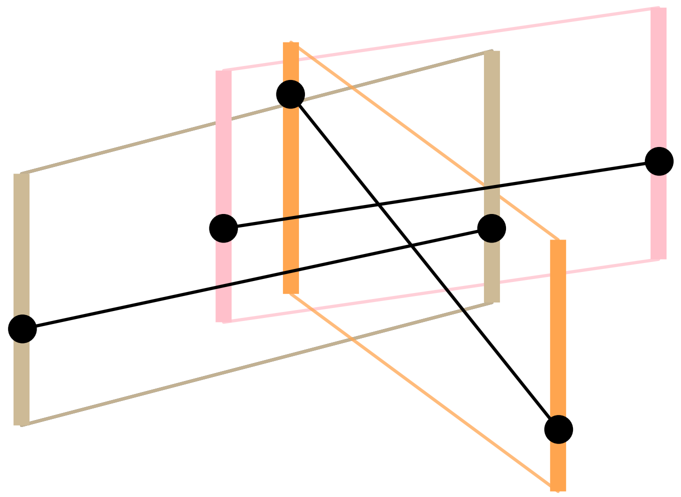
straight-line paths | (x-)monotone paths | arbitrary paths



Our results

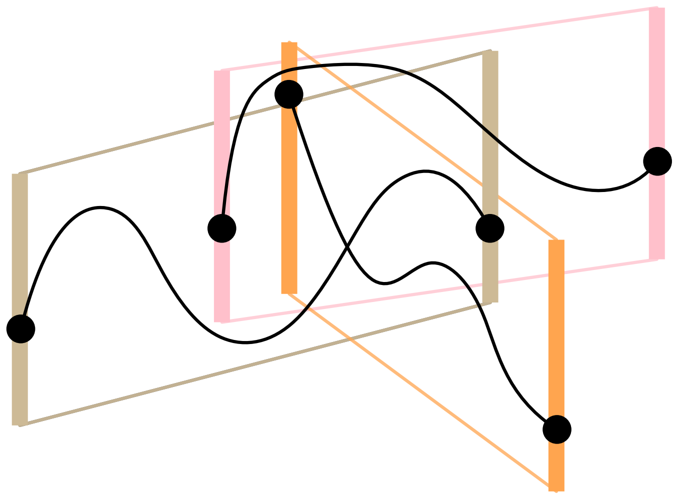
Given a matching, where the endpoint of each edge must lie on a unit vertical segment, can it be drawn without crossings?

straight-line paths

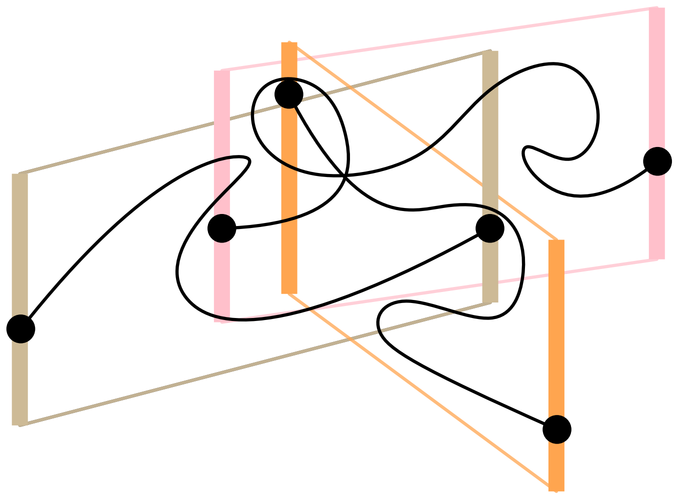


NP-complete

(x-)monotone paths



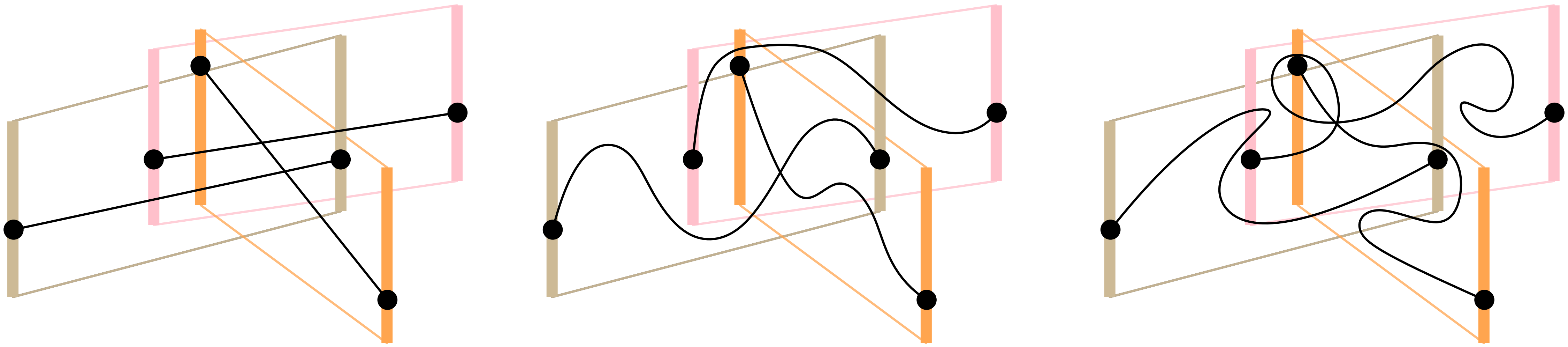
arbitrary paths



Our results

Given a matching, where the endpoint of each edge must lie on a unit vertical segment, can it be drawn without crossings?

straight-line paths | (x-)monotone paths | arbitrary paths

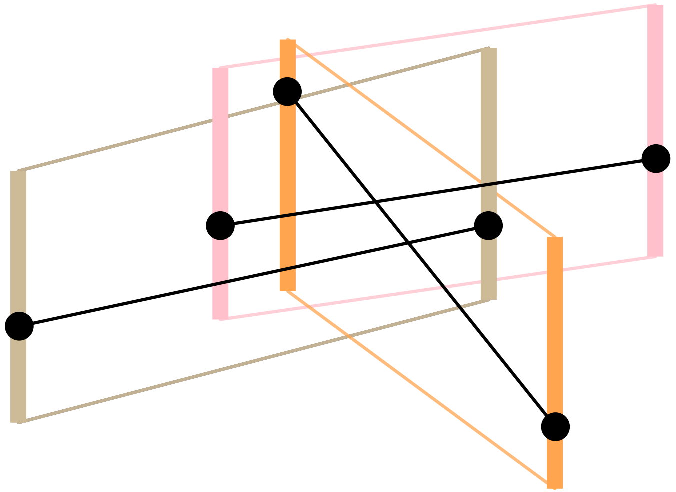


NP-complete | Polynomial

Our results

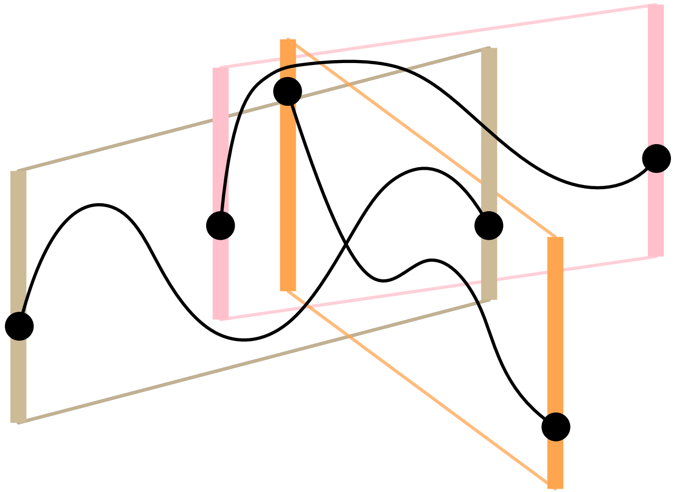
Given a matching, where the endpoint of each edge must lie on a unit vertical segment, can it be drawn without crossings?

straight-line paths



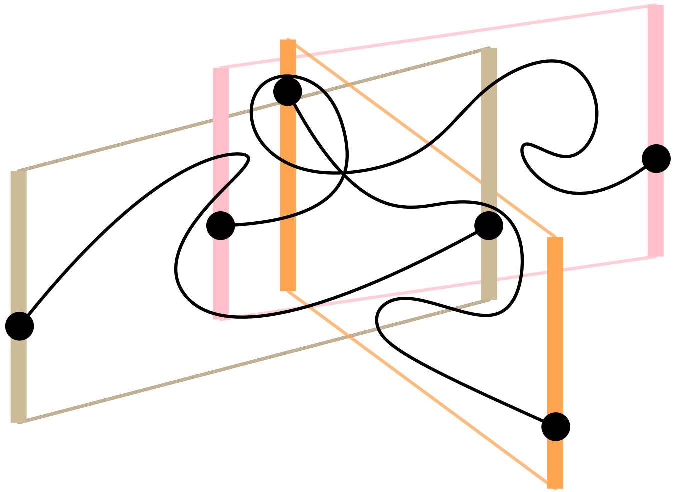
NP-complete

$(x\text{-})$monotone paths



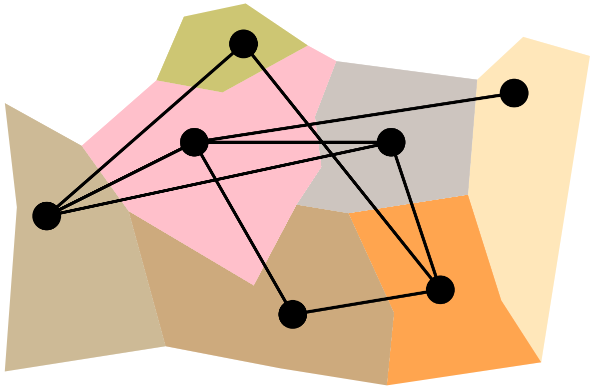
Polynomial

arbitrary paths



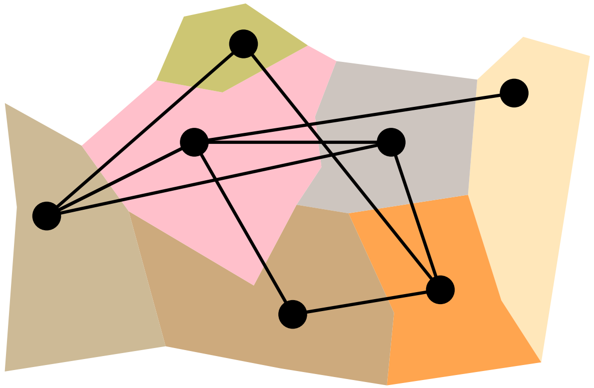
Polynomial under certain assumptions

Related work



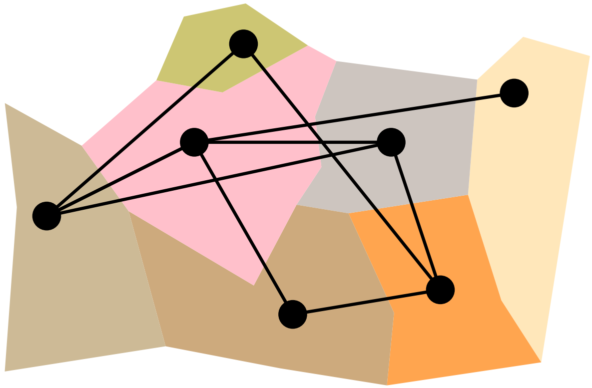
Related work

- Force-directed approach for general problem [Abellanas et al., 2005]



Related work

- Force-directed approach for general problem [Abellanas et al., 2005]
- Straight-line case shown NP-hard for:
  - Cycle graphs when regions are vertical segments [Löffler, 2011]
  - Matchings when regions are vertical segments [Aloupis et al., 2015; Verbeek, 2008]
  - General graphs when regions are unit squares [Angelini et al., 2014]

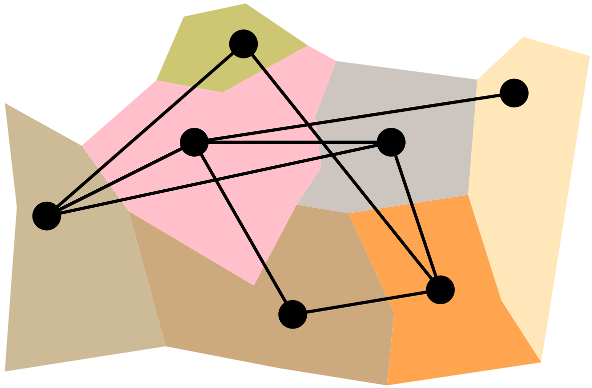


Related work

- Force-directed approach for general problem [Abellanas et al., 2005]

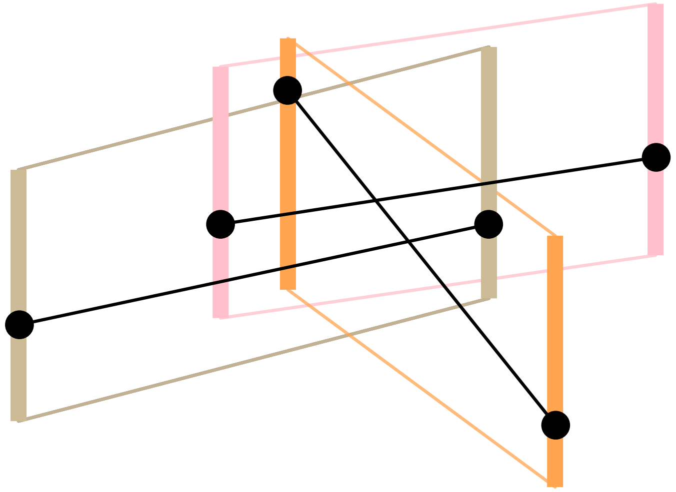
- Straight-line case shown NP-hard for:
  - Cycle graphs when regions are vertical segments [Löffler, 2011]
  - Matchings when regions are vertical segments [Aloupis et al., 2015; Verbeek, 2008]
  - General graphs when regions are unit squares [Angelini et al., 2014]

- Several other related problems:
  - Fitting planar graphs to planar maps [Alam et al., 2015]
  - c-planarity / ordered-level planarity [Feng et al., 1995, Klemz and Rote, 2017]
  - Manhattan geodesic planarity [Katz et al., 2009]
  - Non-crossing connectors in the plane [Kratochvíl and Ueckerdt, 2013]



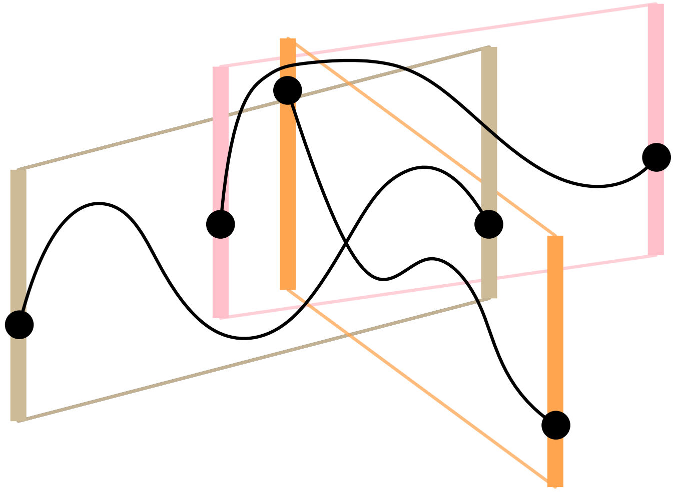
Our results

straight line paths



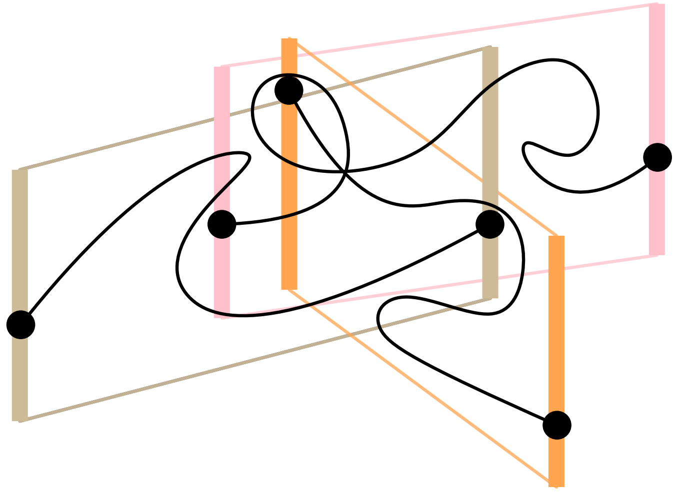
NP-complete

(*x*-)monotone paths



Polynomial

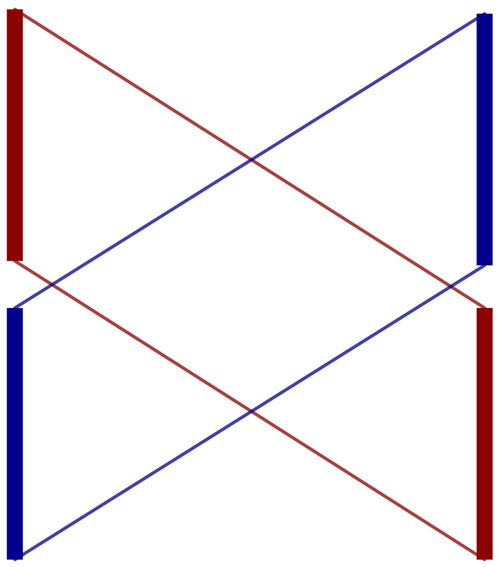
arbitrary paths



Polynomial under certain assumptions

How can two tubes intersect?

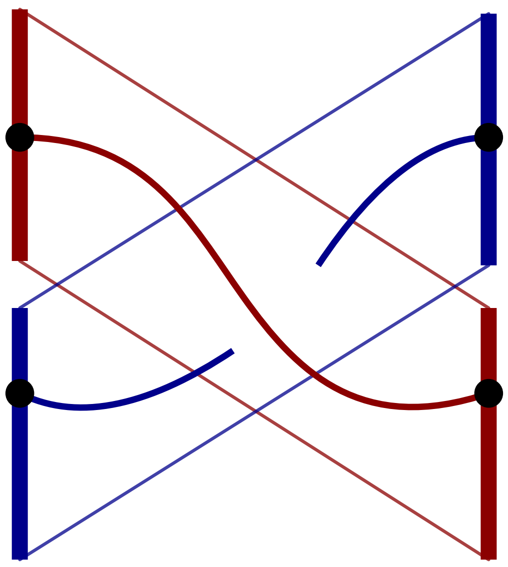
How can two tubes intersect?



full crossing

(no solution)

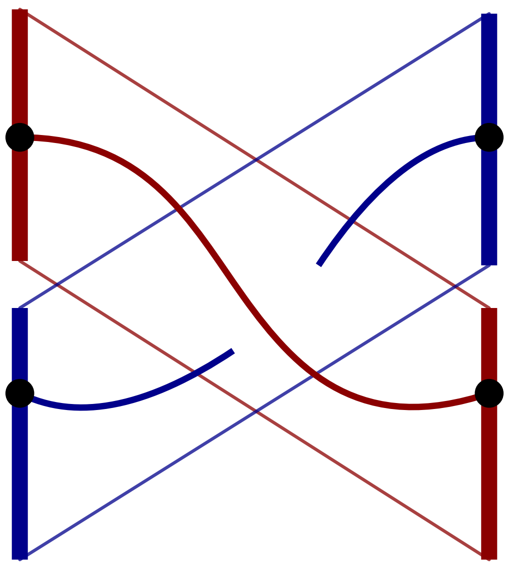
How can two tubes intersect?



full crossing

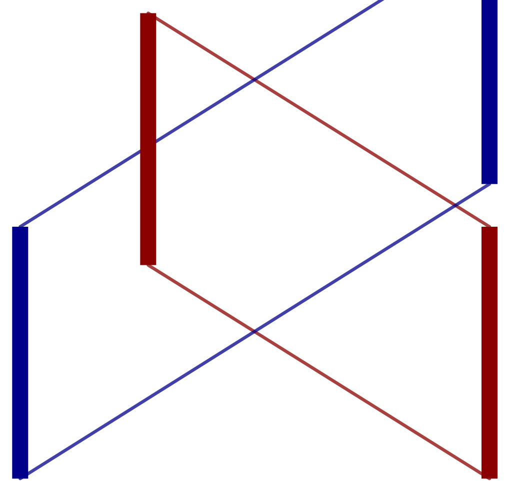
(no solution)

How can two tubes intersect?



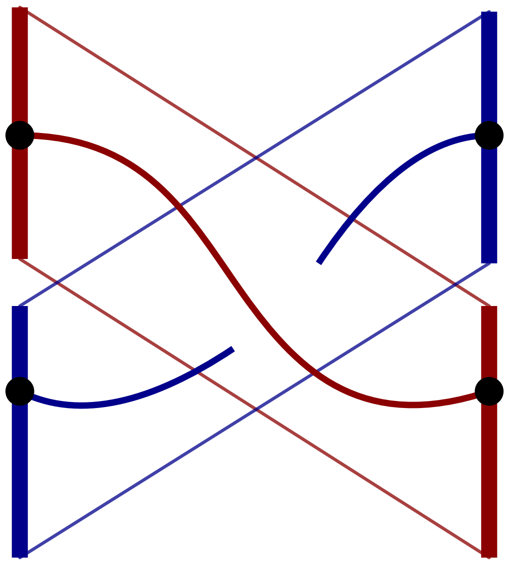
full crossing

(no solution)



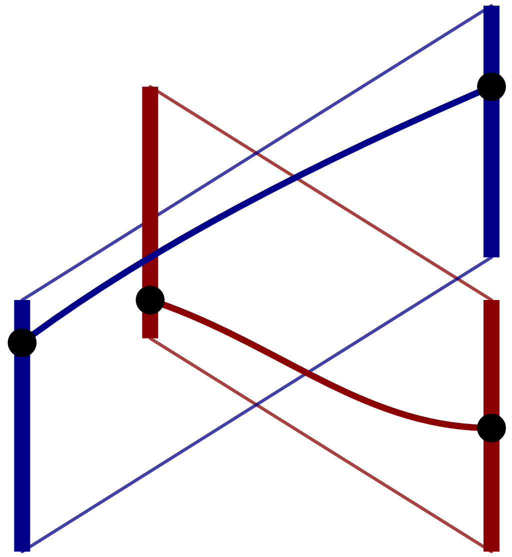
single intersection

How can two tubes intersect?



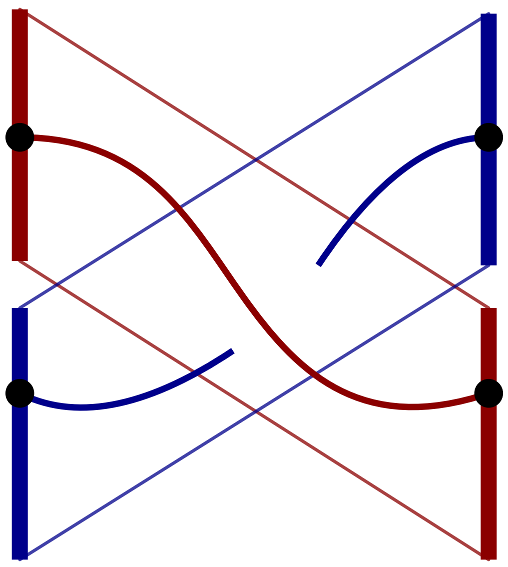
full crossing

(no solution)



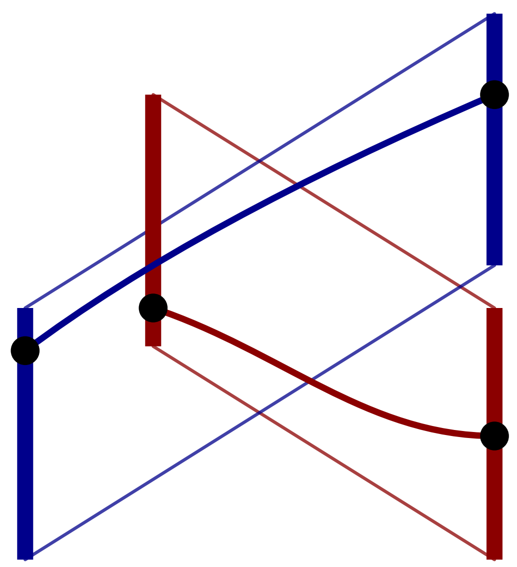
single intersection

How can two tubes intersect?



full crossing

(no solution)

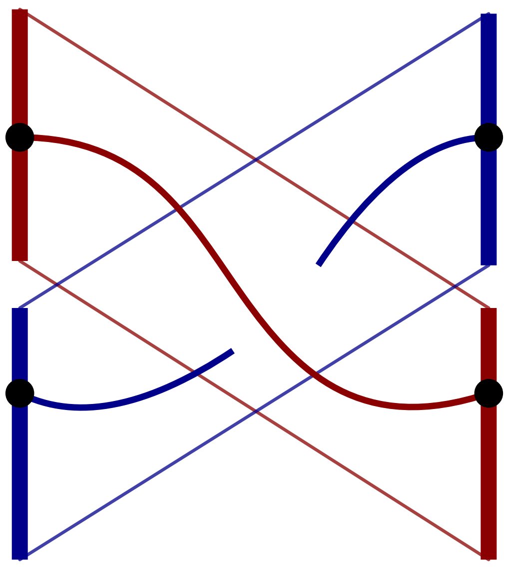


single intersection

induces vertical order between paths

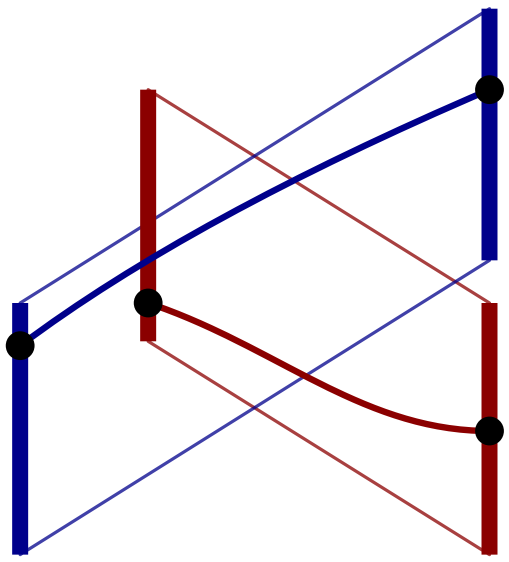
e.g., blue path is "above" red

How can two tubes intersect?



full crossing

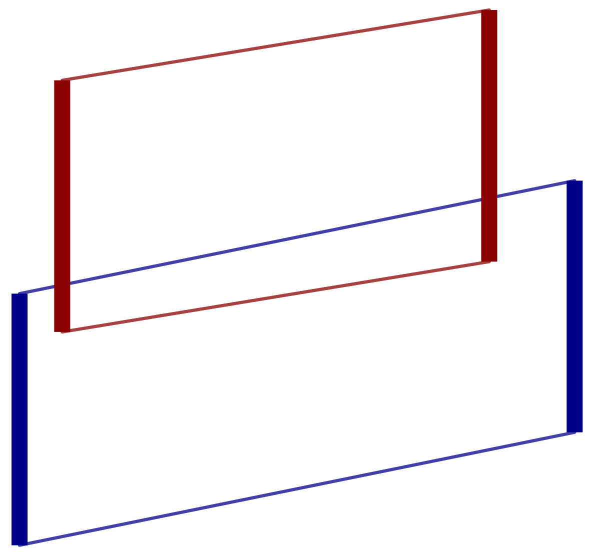
(no solution)



single intersection

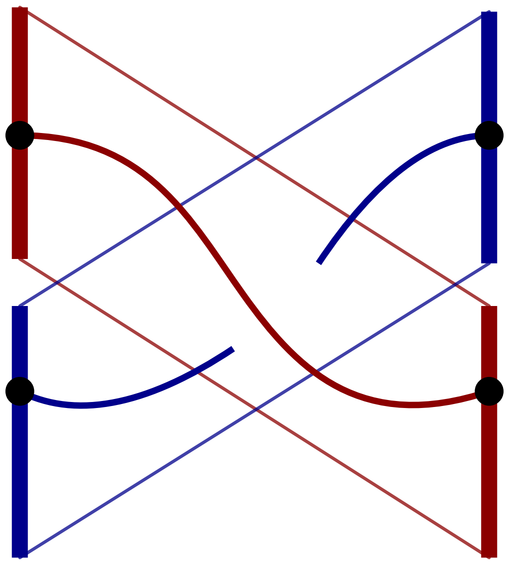
induces vertical order
between paths

e.g., blue path is
“above” red



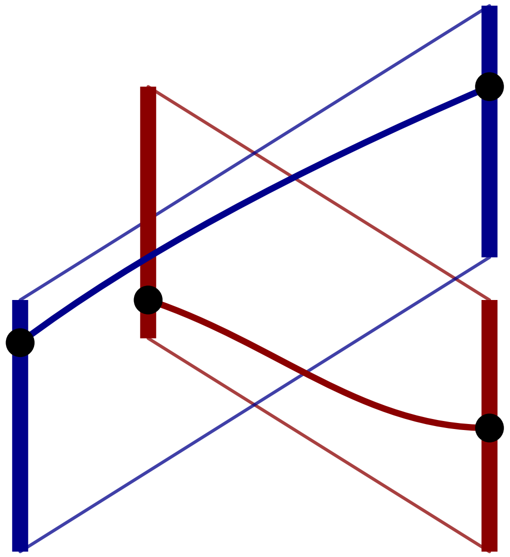
double intersection

How can two tubes intersect?



full crossing

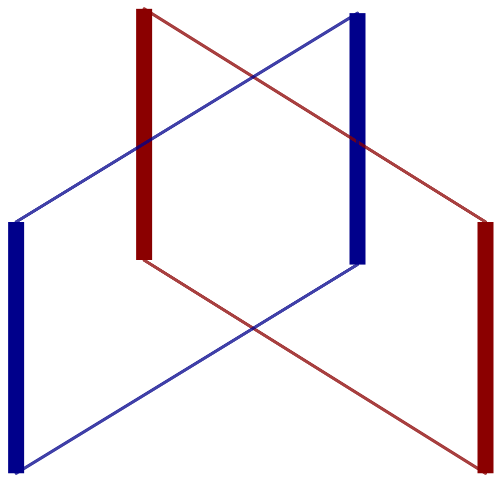
(no solution)



single intersection

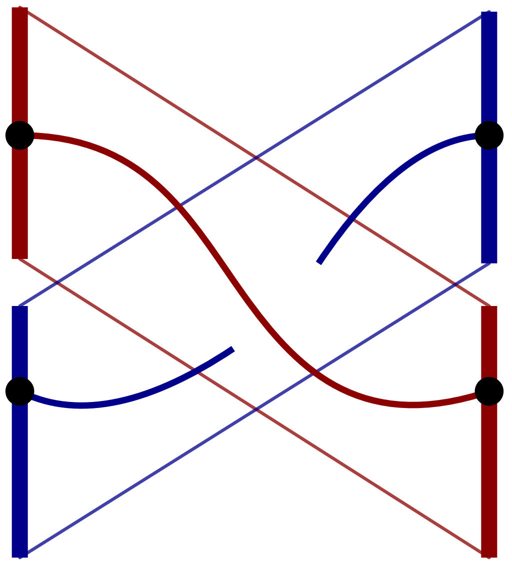
induces vertical order between paths

e.g., blue path is "above" red



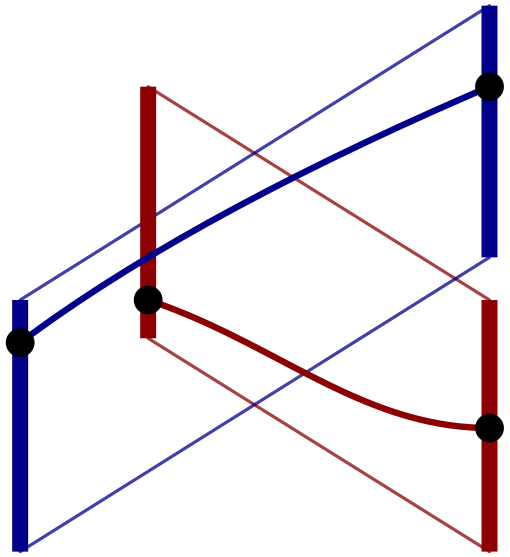
double intersection

How can two tubes intersect?



full crossing

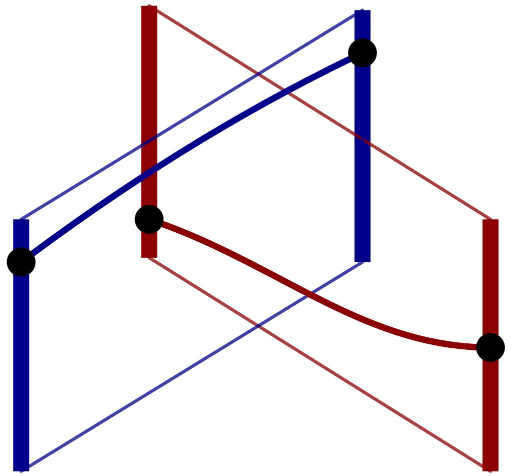
(no solution)



single intersection

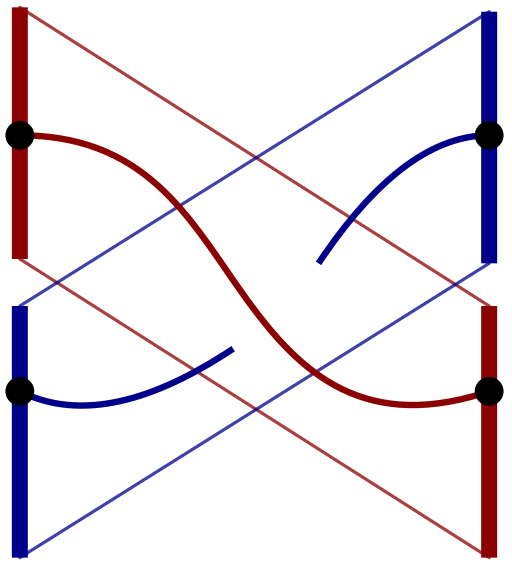
induces vertical order between paths

e.g., blue path is "above" red



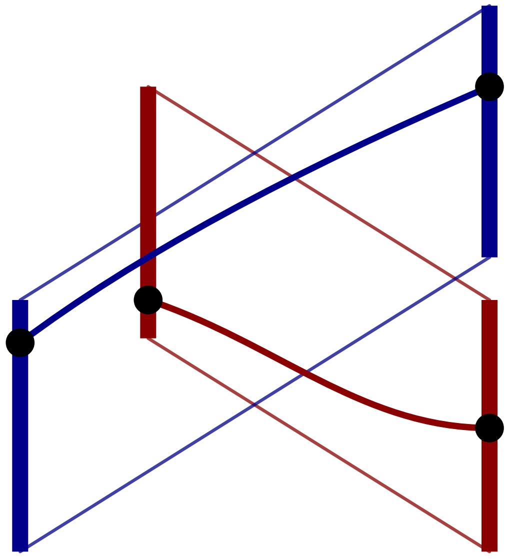
double intersection

How can two tubes intersect?



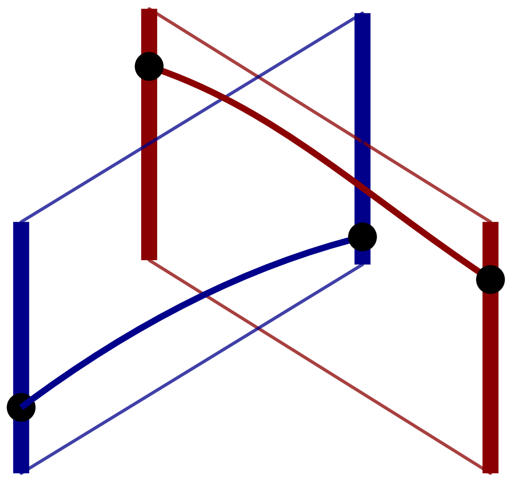
full crossing

(no solution)



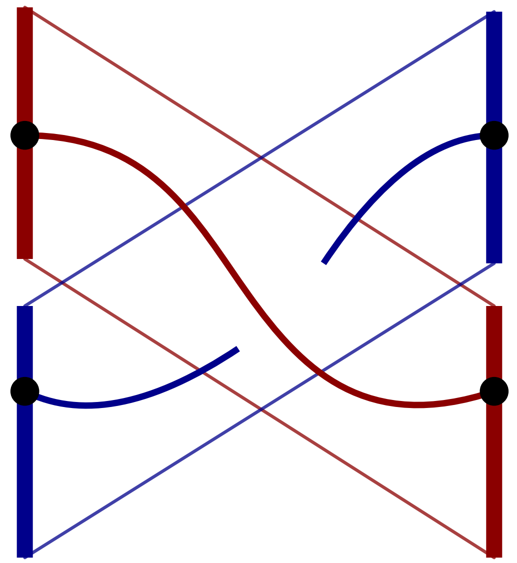
single intersection

induces vertical order
between paths
e.g., blue path is
“above” red



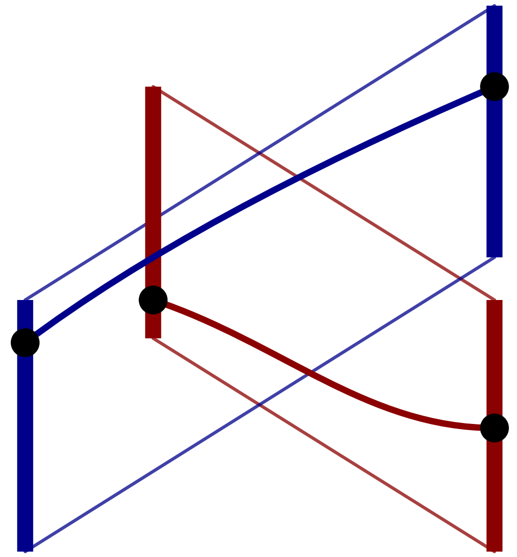
double intersection

How can two tubes intersect?



full crossing

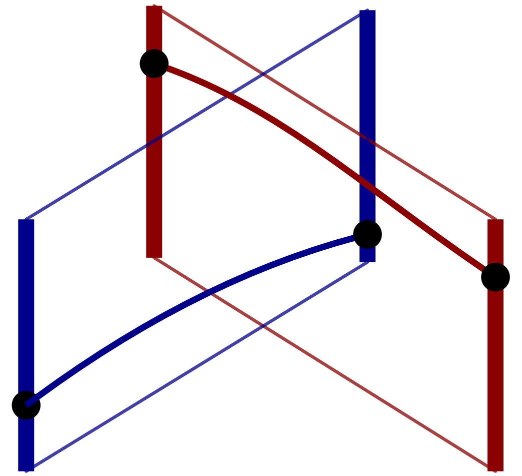
(no solution)



single intersection

induces vertical order
between paths
e.g., blue path is
"above" red

→ we can define an *order graph*

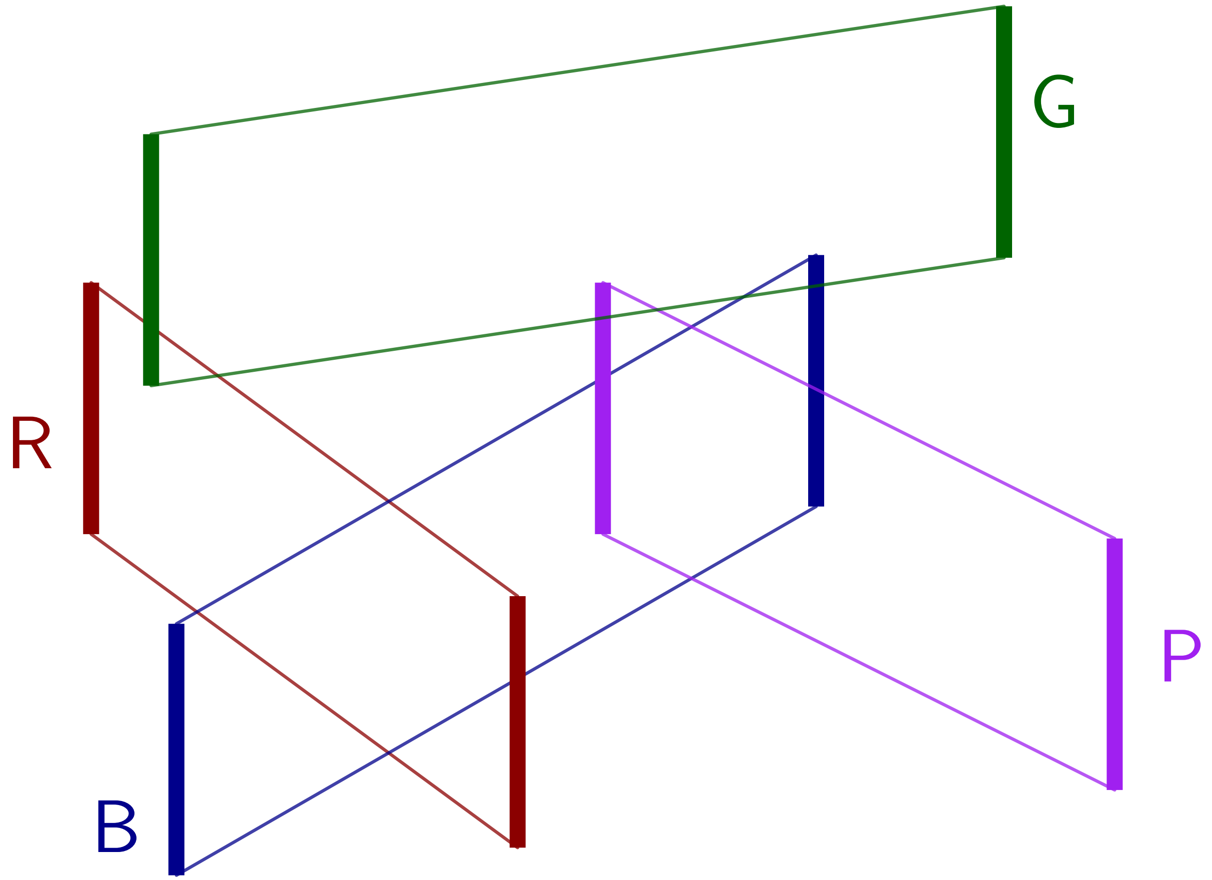


double intersection

Order graph

Order graph

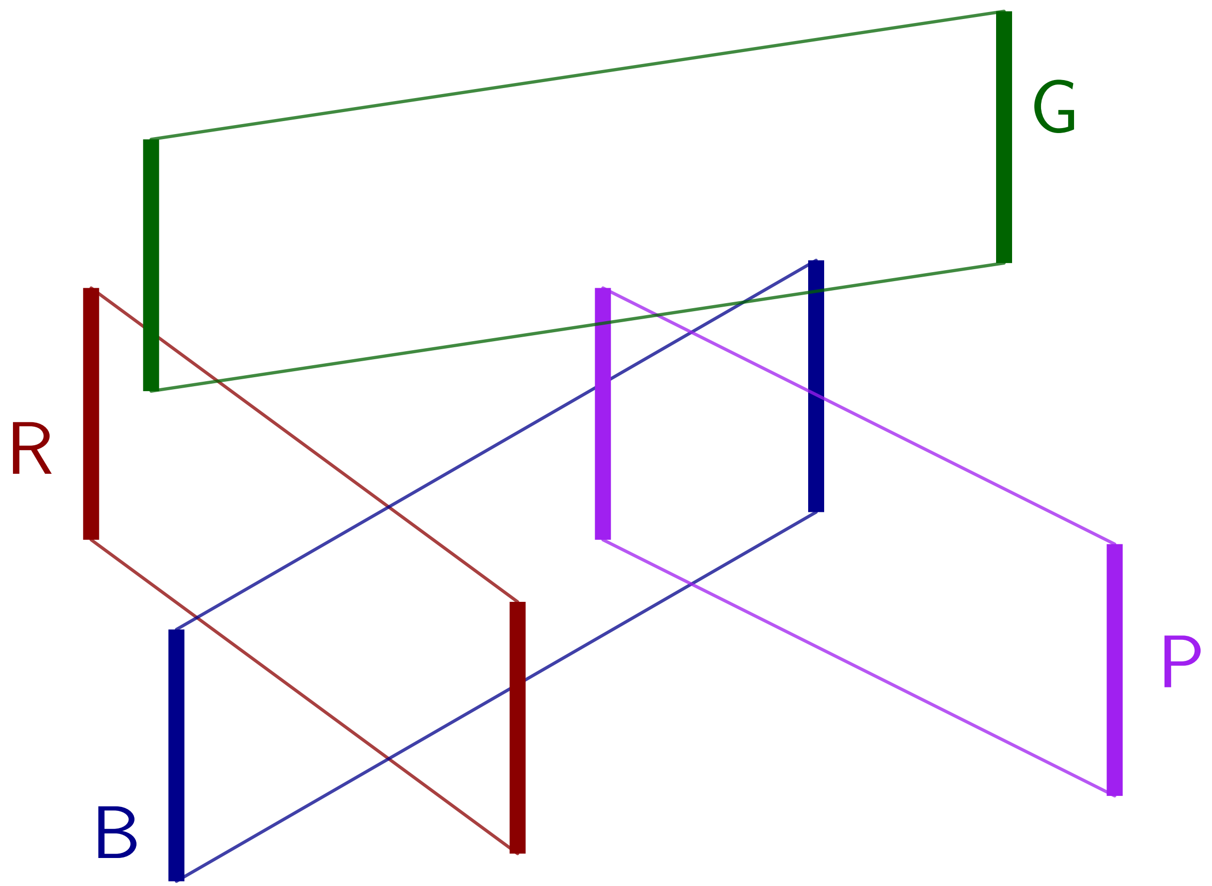
Tubes



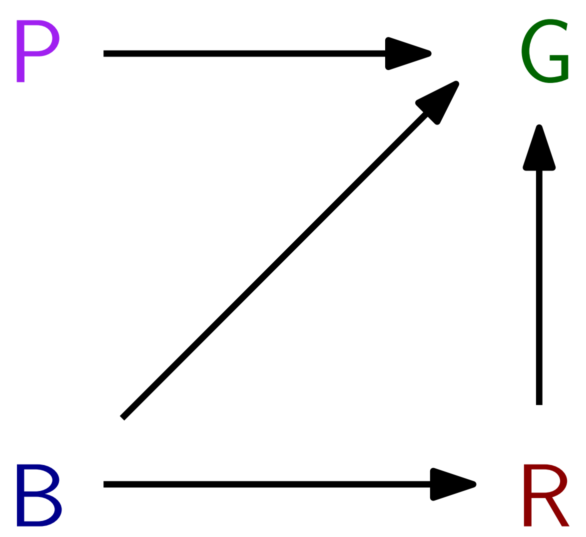
Order graph

Order graph

Tubes

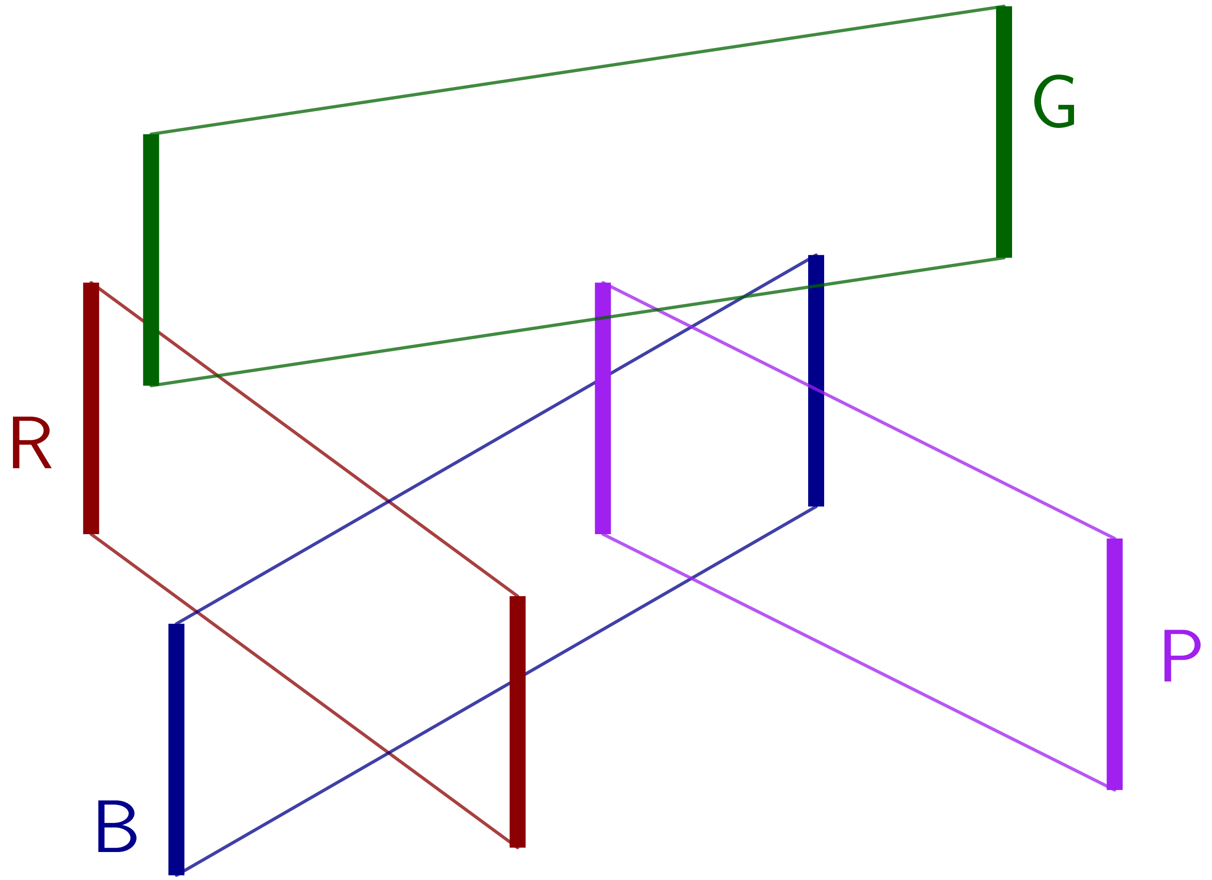


Order graph

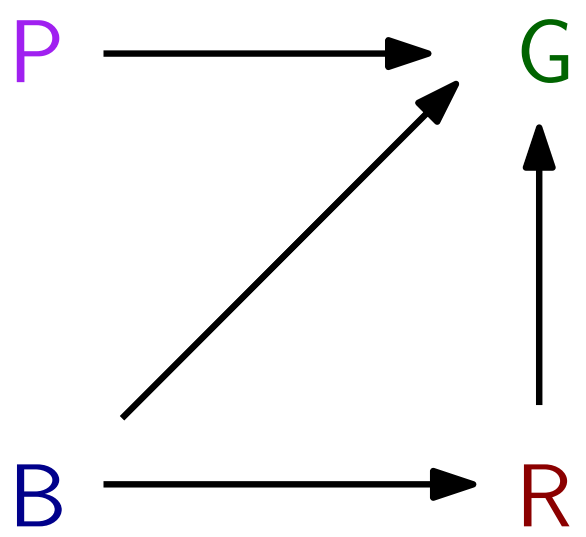


Order graph

Tubes



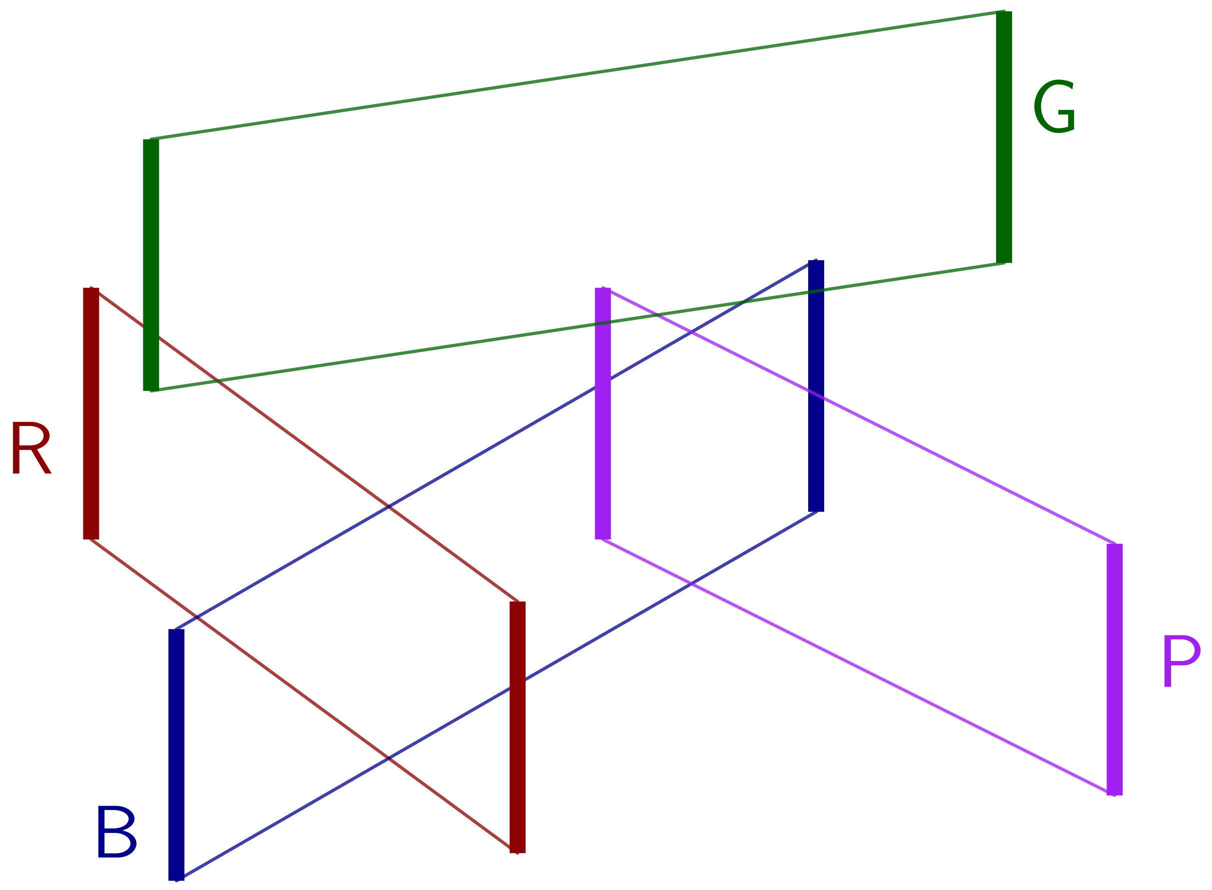
Order graph



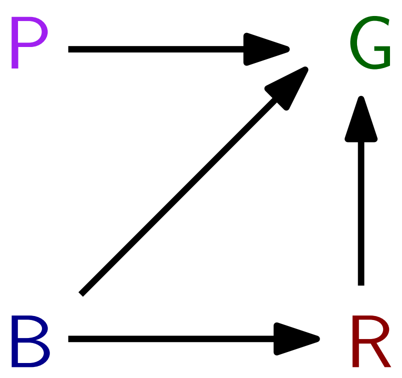
There is a solution if and only if the order graph has no directed cycles

Drawing the tubes in order

Tubes



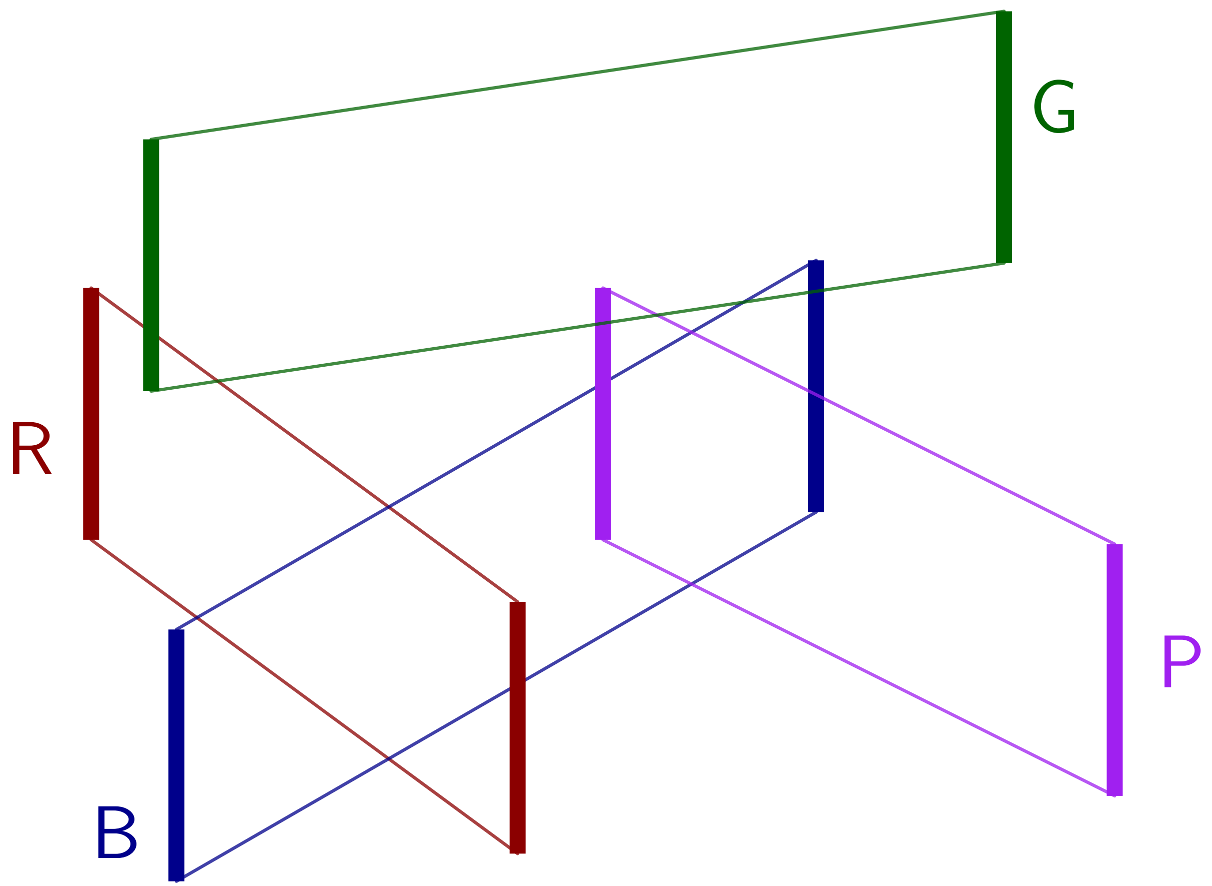
Order graph



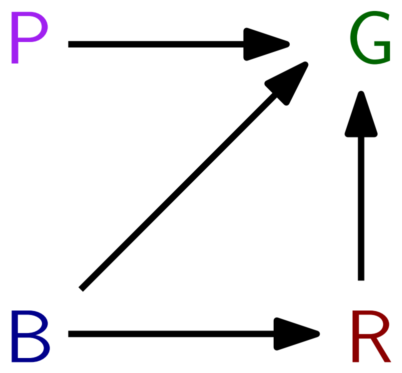
If no directed cycles:

Drawing the tubes in order

Tubes



Order graph

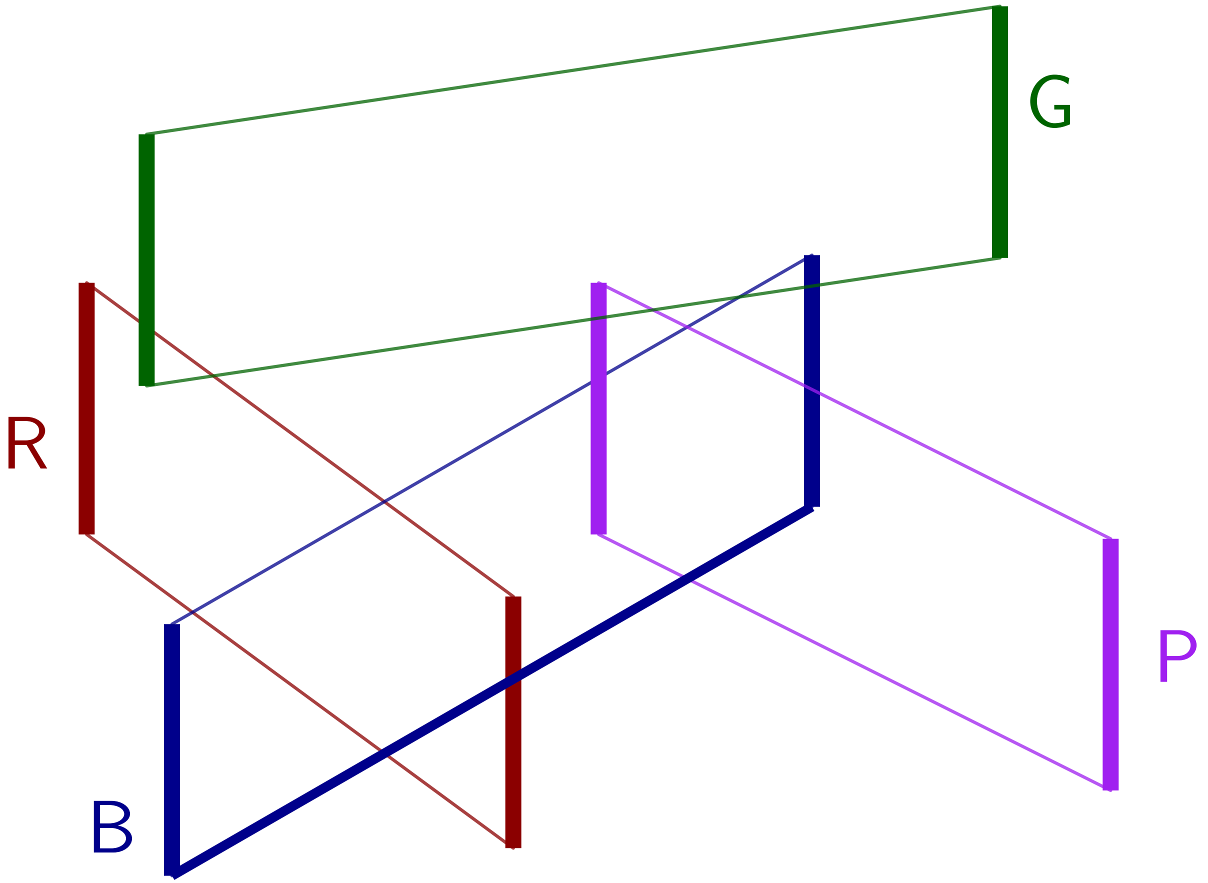


If no directed cycles:

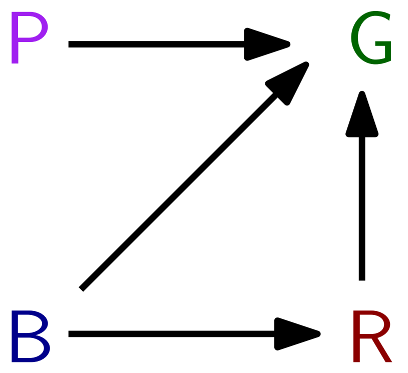
- extract total order
  e.g., B R P G
- follow that order, drawing paths as low as possible

Drawing the tubes in order

Tubes



Order graph

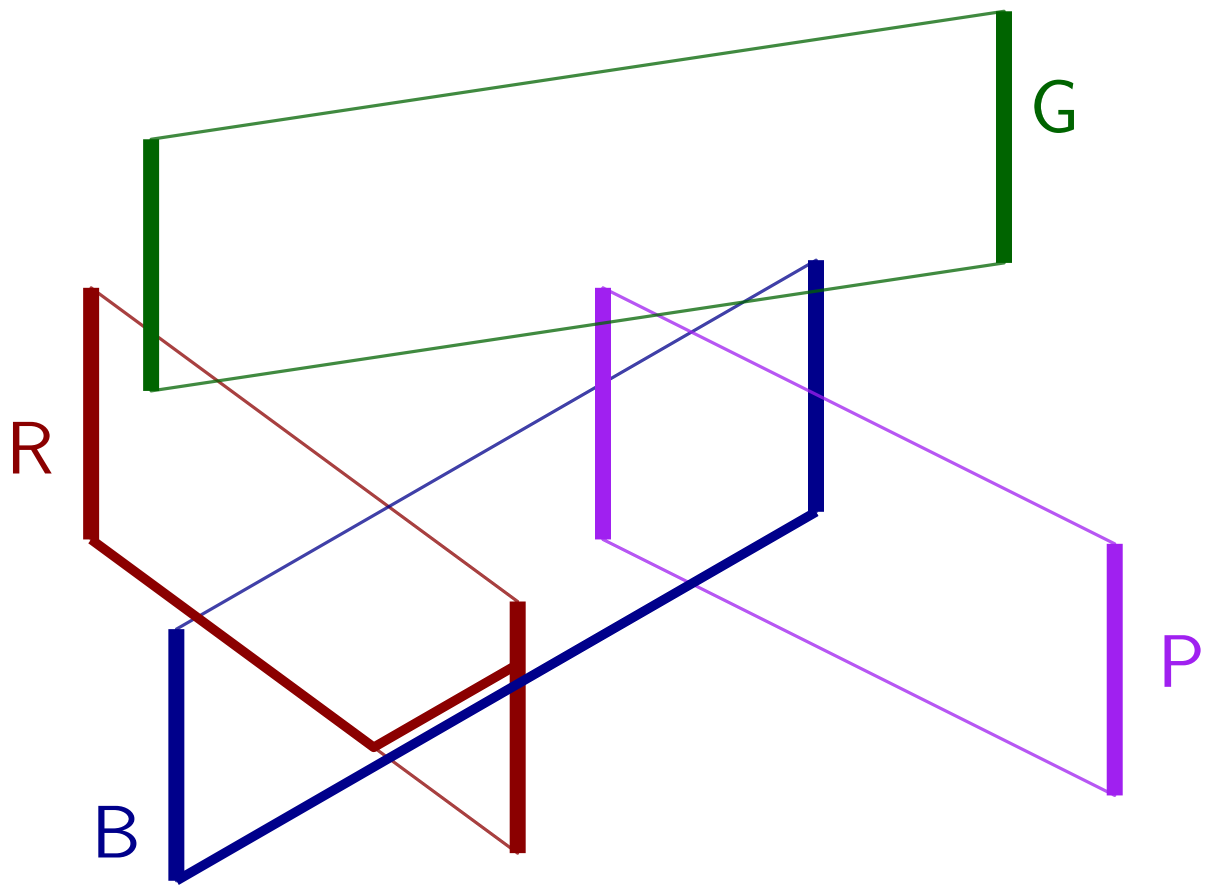


If no directed cycles:

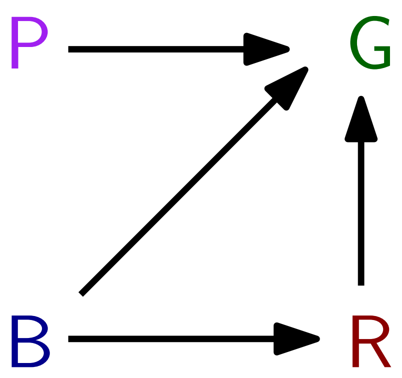
- extract total order
  e.g., B R P G
- follow that order, drawing paths as low as possible

Drawing the tubes in order

Tubes



Order graph

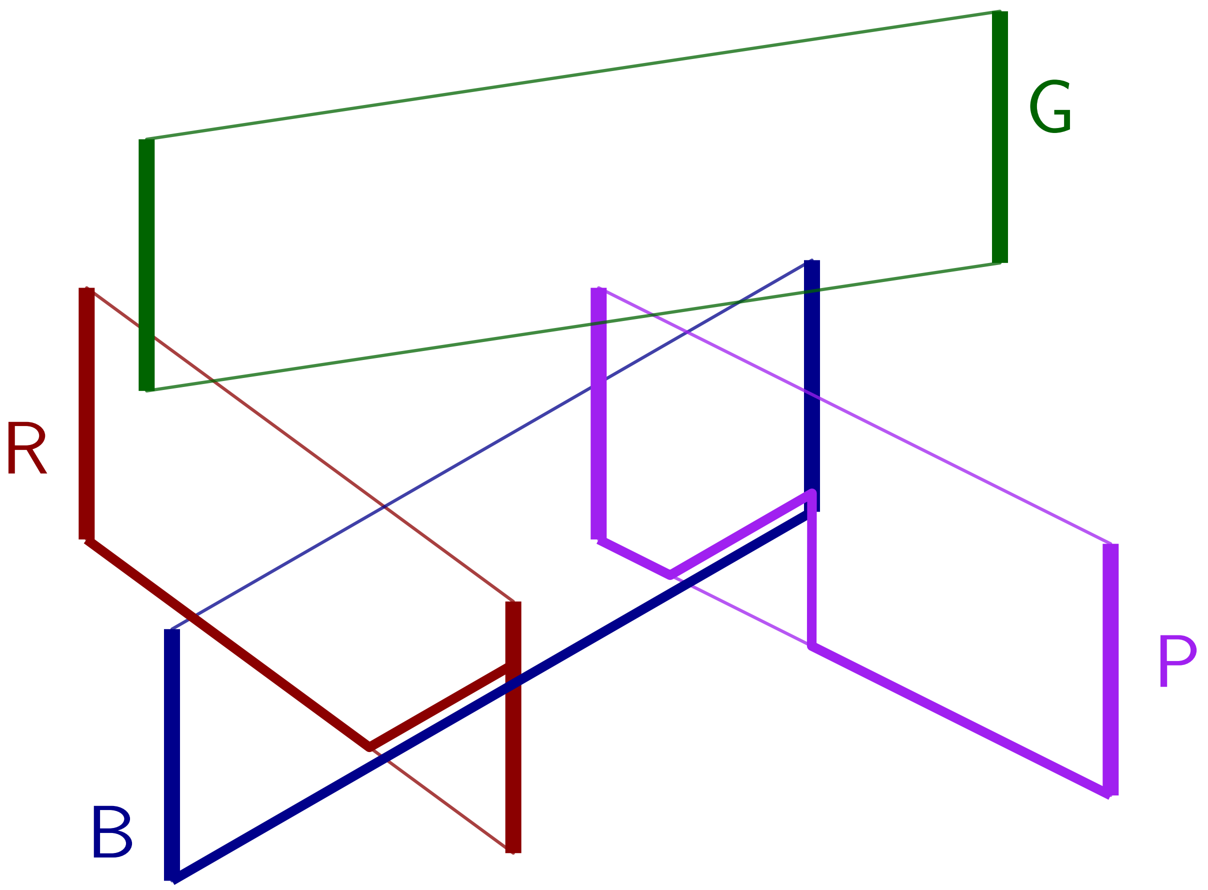


If no directed cycles:

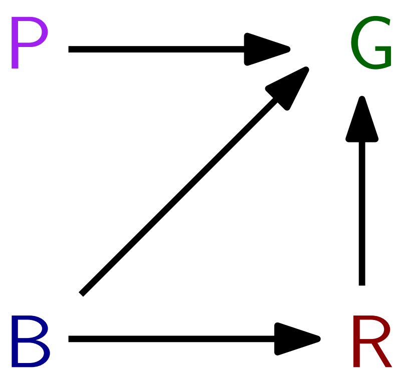
- extract total order
  e.g., B R P G
- follow that order, drawing paths as low as possible

Drawing the tubes in order

Tubes



Order graph

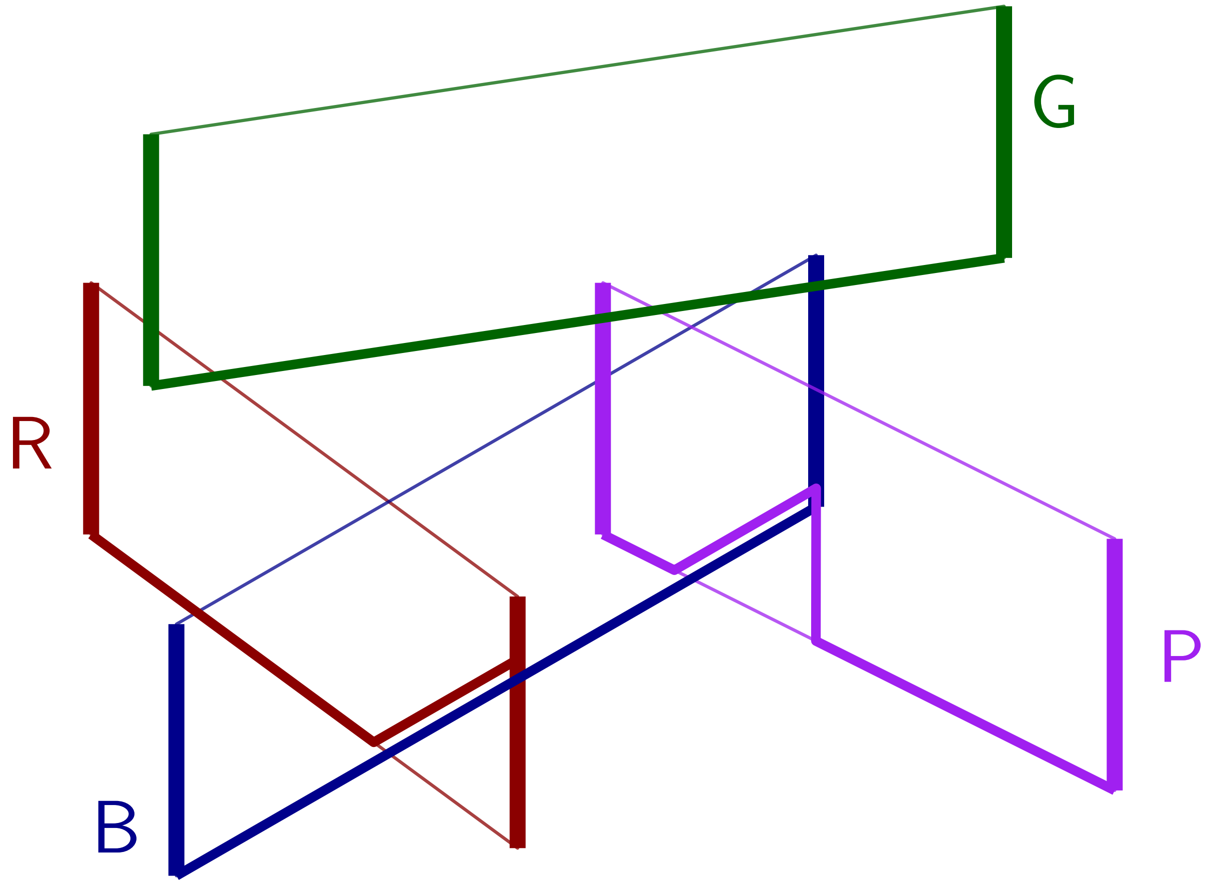


If no directed cycles:

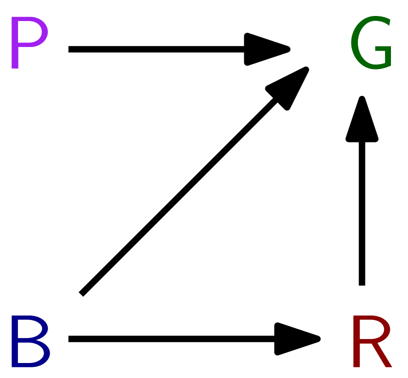
- extract total order
  e.g., B R P G
- follow that order, drawing paths as low as possible

Drawing the tubes in order

Tubes



Order graph

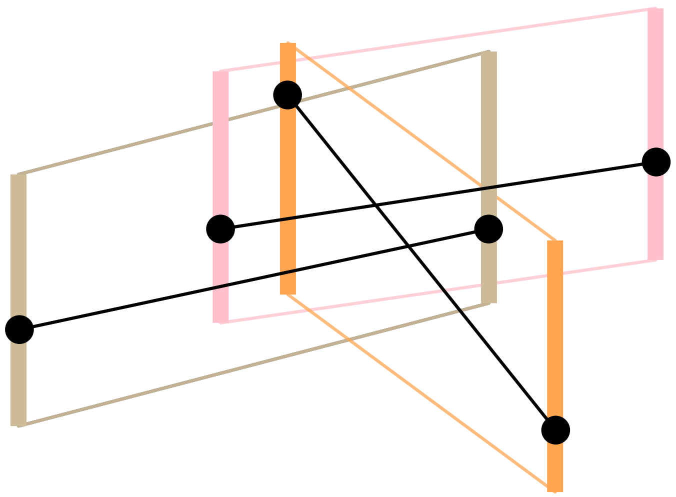


If no directed cycles:

- extract total order
  e.g., B R P G
- follow that order, drawing paths as low as possible

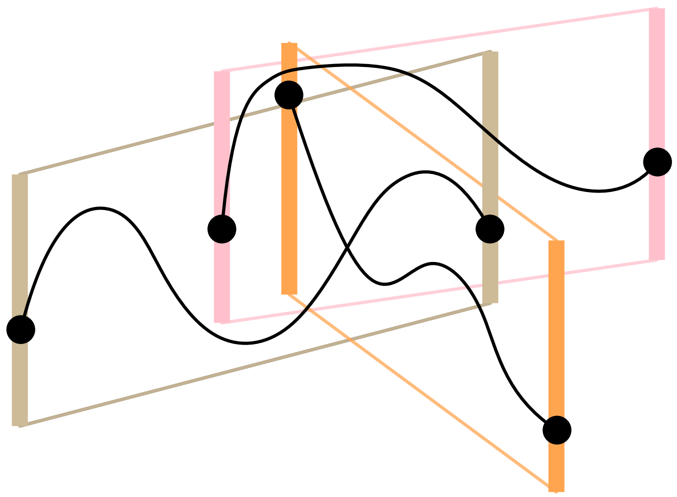
Our results

straight line paths



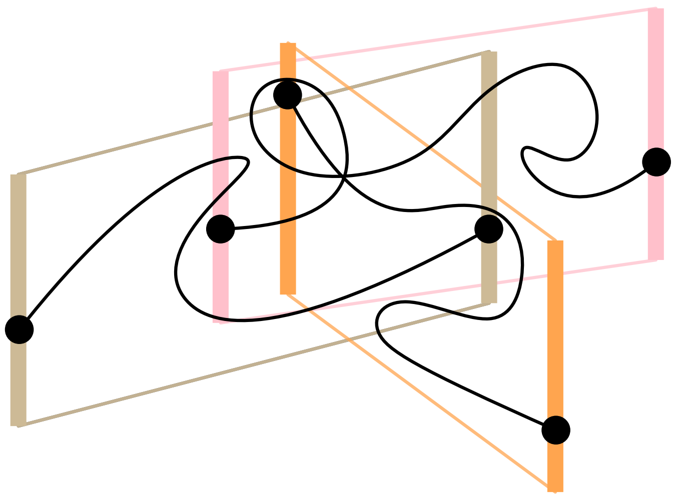
NP-complete

(x-)monotone paths



Polynomial

arbitrary paths

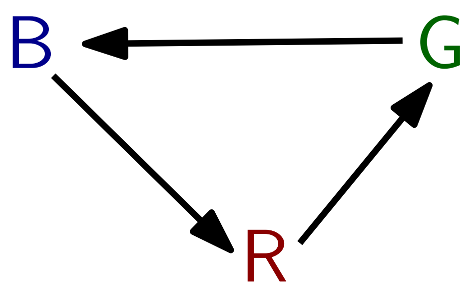


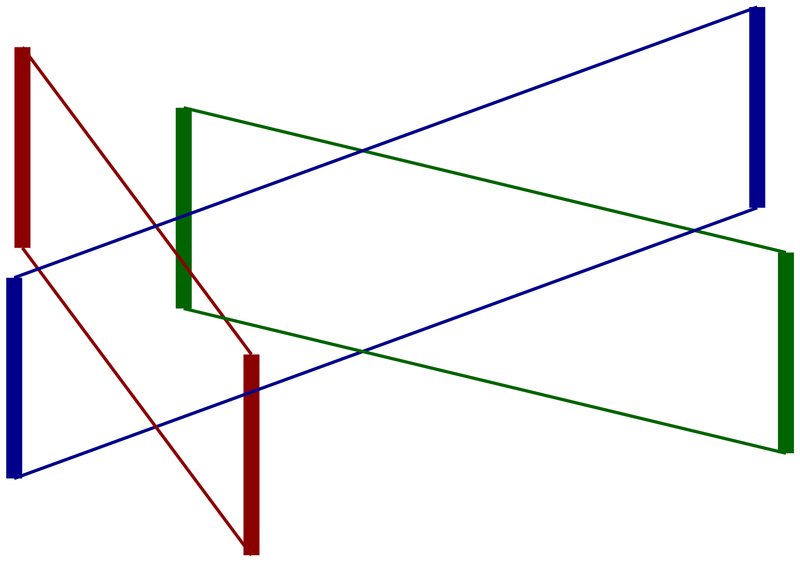
Polynomial under certain assumptions

Arbitrary paths

Arbitrary paths

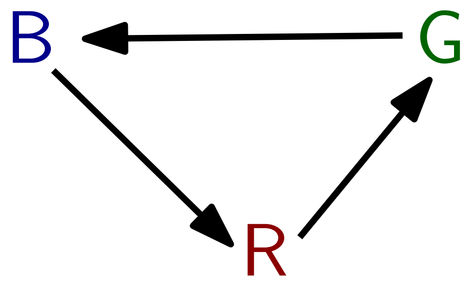
Now, some cycles of directed edges can be solved:

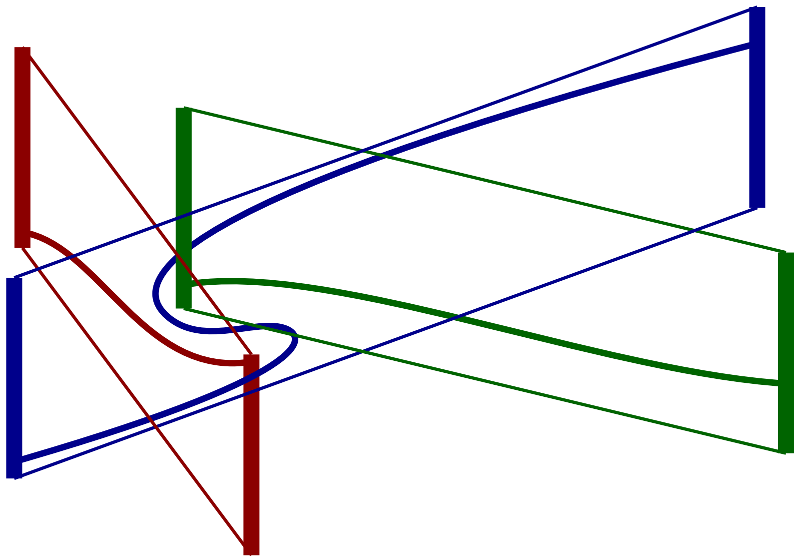




Arbitrary paths

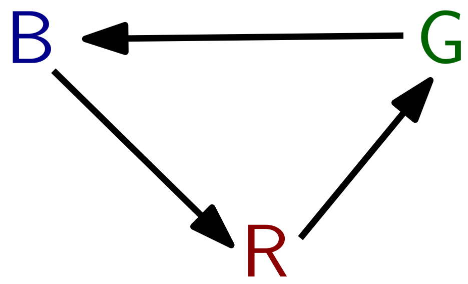
Now, some cycles of directed edges can be solved:

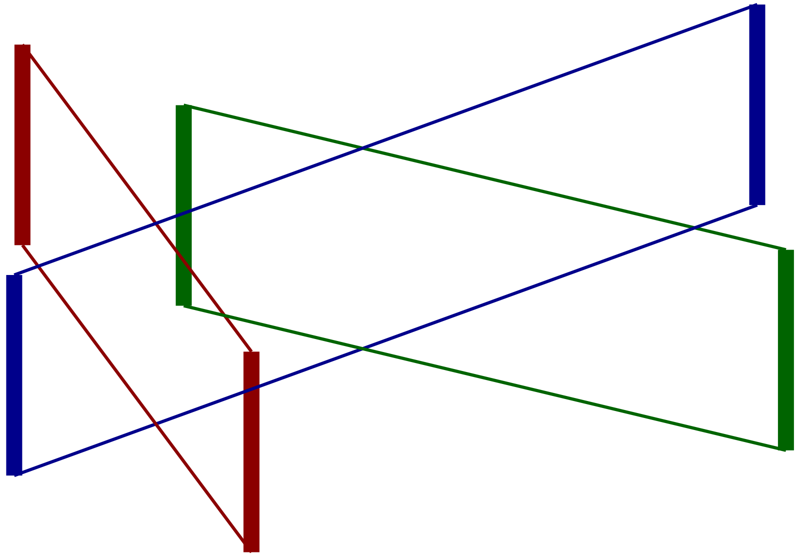




Arbitrary paths

Now, some cycles of directed edges can be solved:

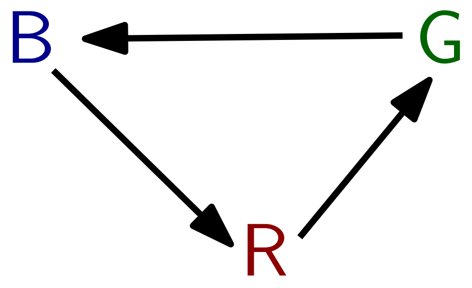


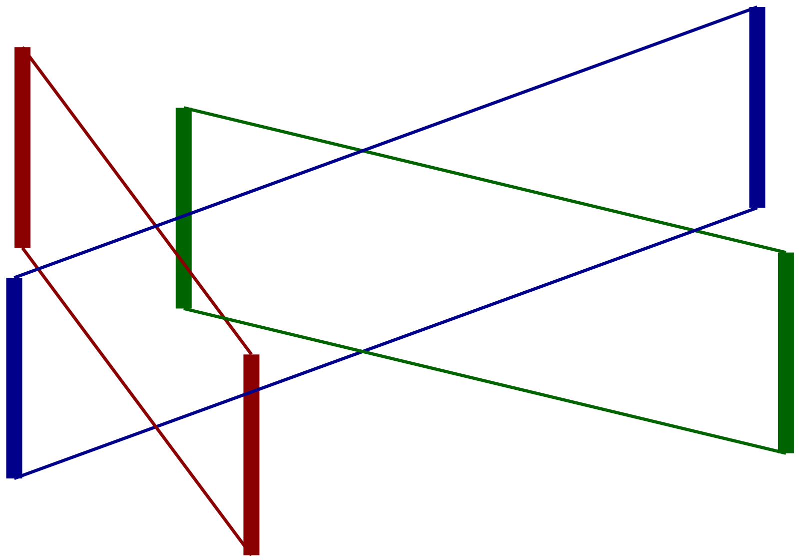


Related problem: *Non-crossing connectors* [Kratochvíl and Ueckerdt, 2013]

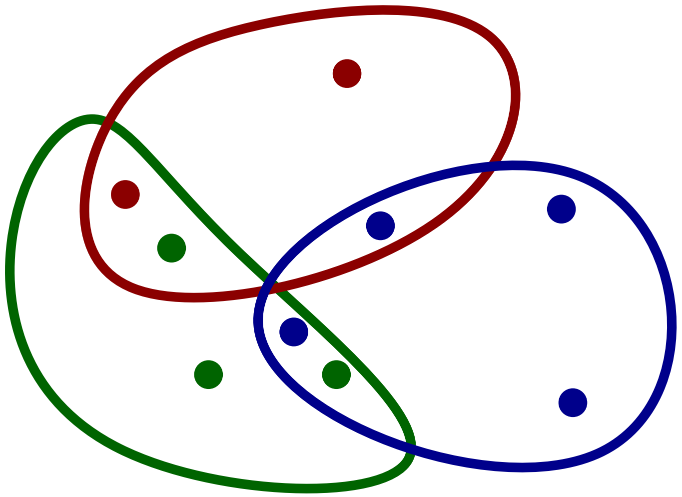
Arbitrary paths

Now, some cycles of directed edges
can be solved:



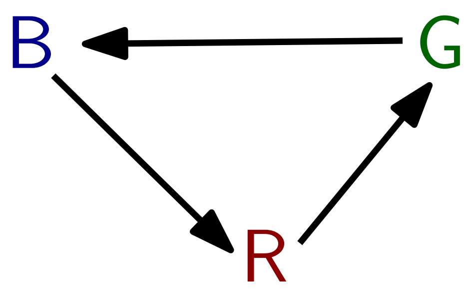


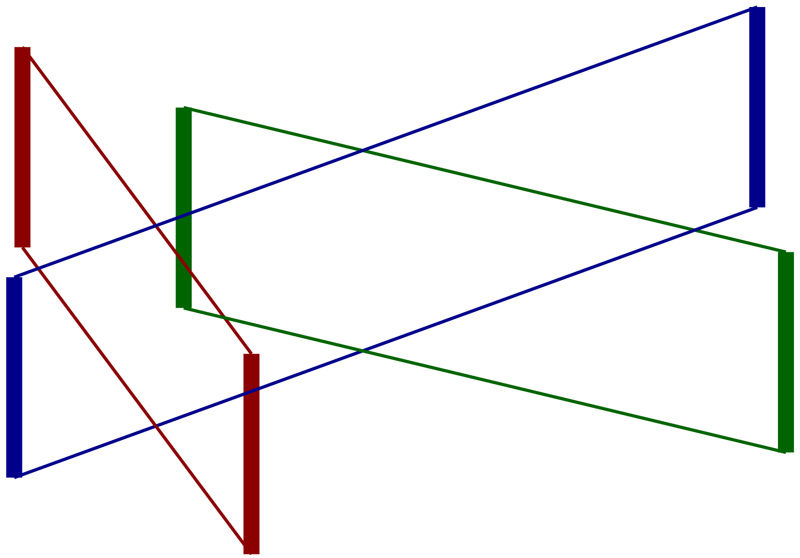
Related problem: *Non-crossing connectors* [Kratochvíl and Ueckerdt, 2013]



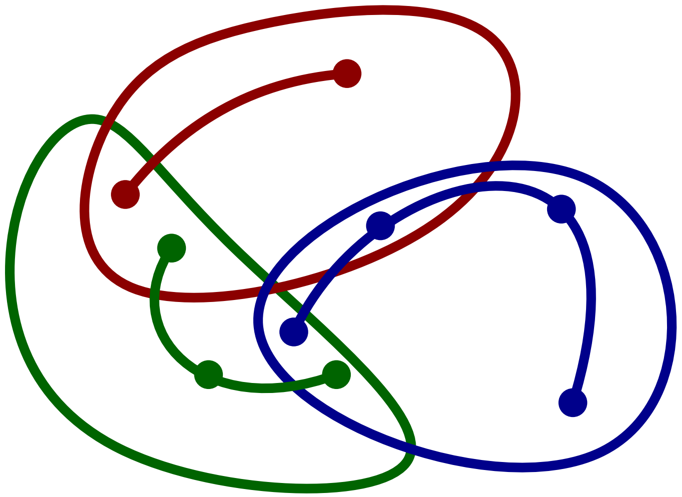
Arbitrary paths

Now, some cycles of directed edges can be solved:



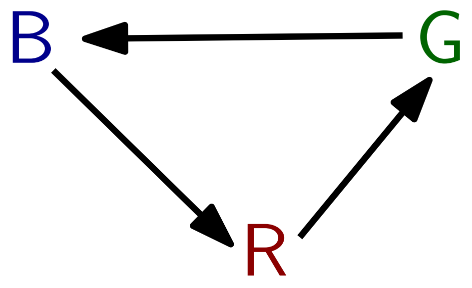


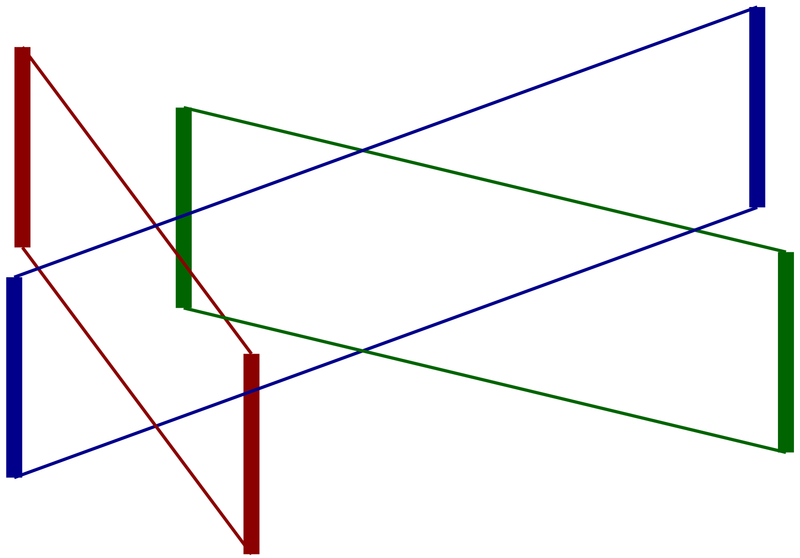
Related problem: *Non-crossing connectors* [Kratochvíl and Ueckerdt, 2013]



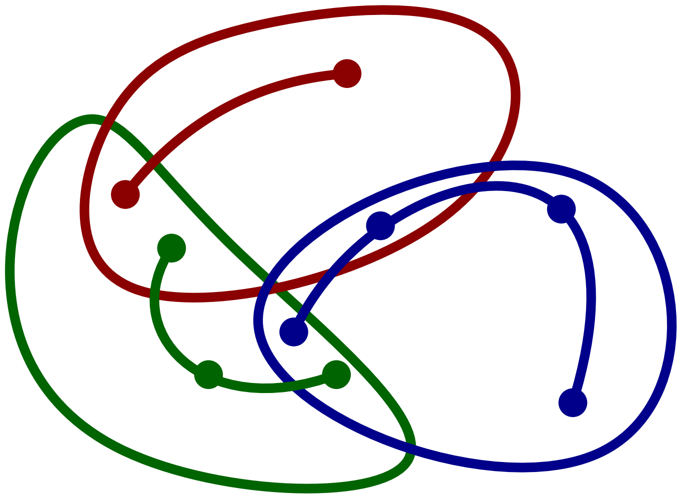
Arbitrary paths

Now, some cycles of directed edges can be solved:



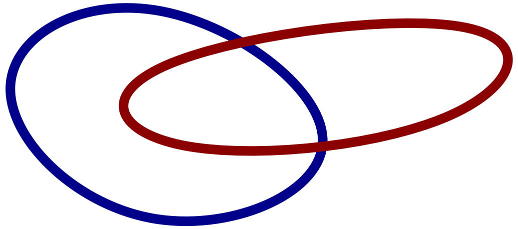


Related problem: *Non-crossing connectors* [Kratochvíl and Ueckerdt, 2013]

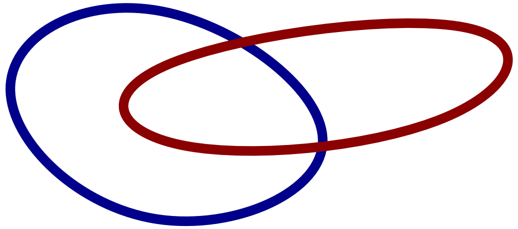


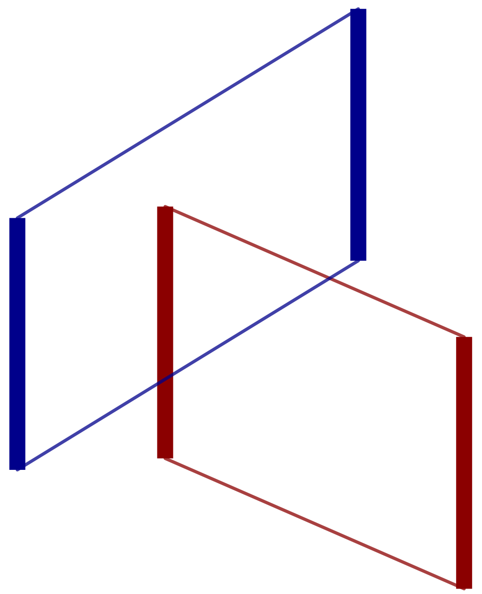
They prove: If the regions are pseudo-disks, there is always a solution

Are our tubes pseudo-disks?



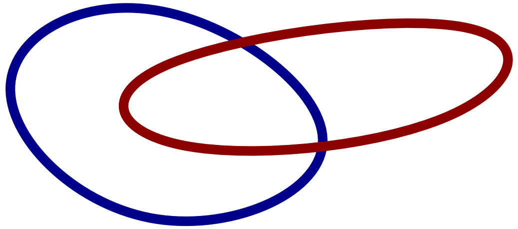
Are our tubes pseudo-disks?

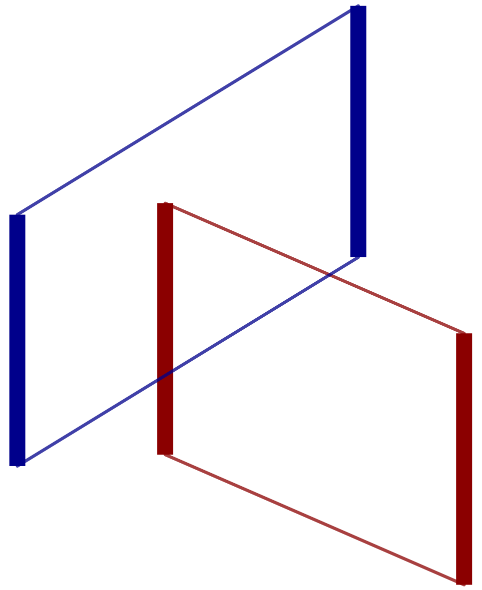




single intersection

Are our tubes pseudo-disks?

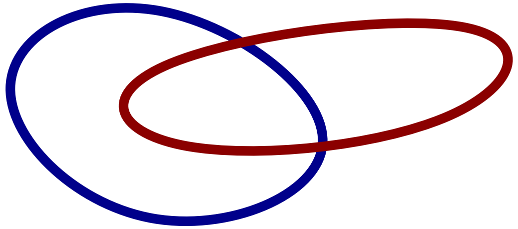


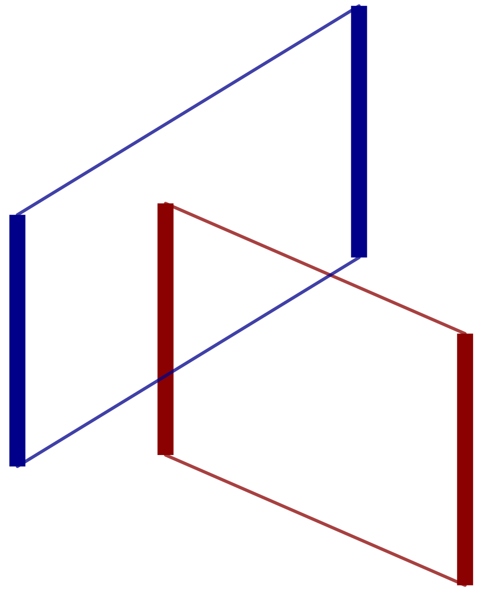


single intersection

pseudo-disks!

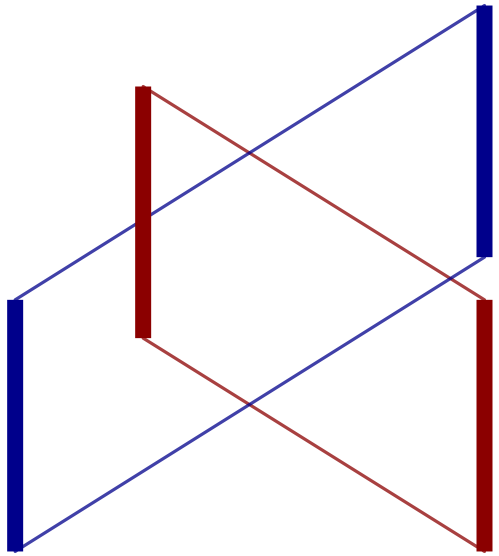
Are our tubes pseudo-disks?





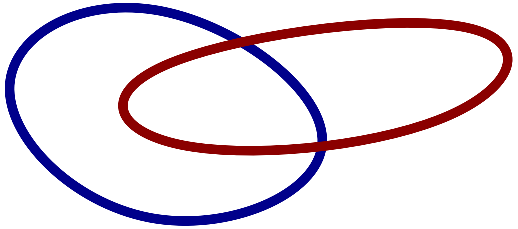
single intersection

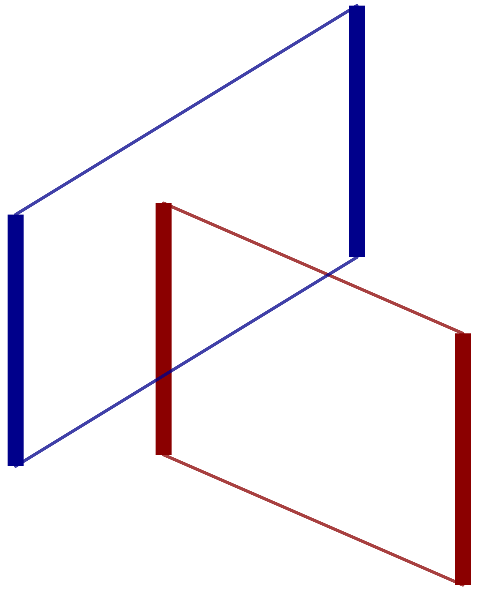
pseudo-disks!



single intersection

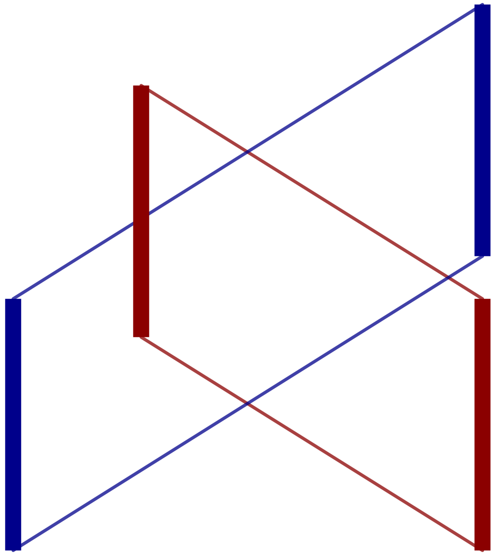
Are our tubes pseudo-disks?





single intersection

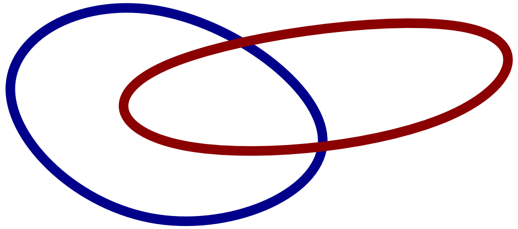
pseudo-disks!

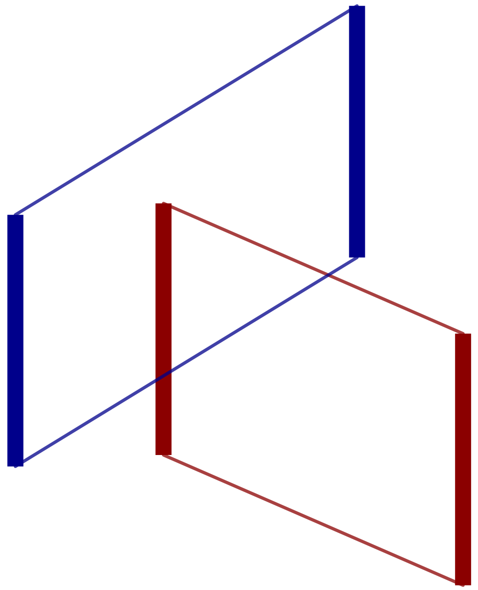


single intersection

not pseudo-disks

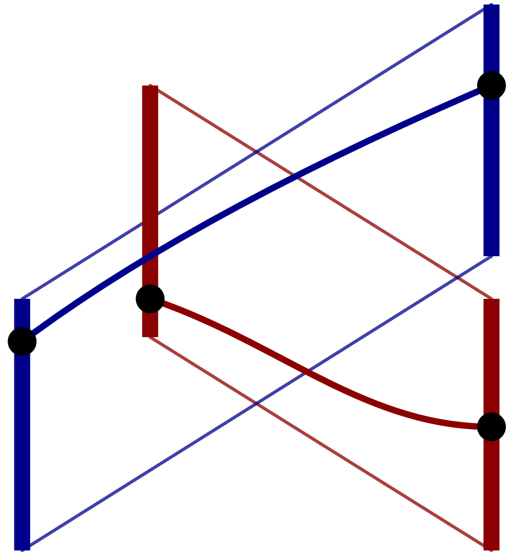
Are our tubes pseudo-disks?





single intersection

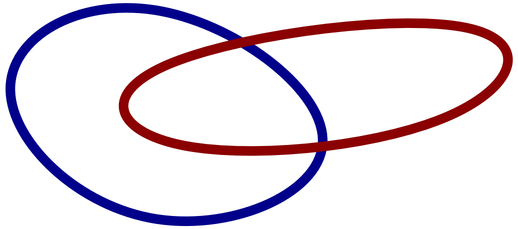
pseudo-disks!

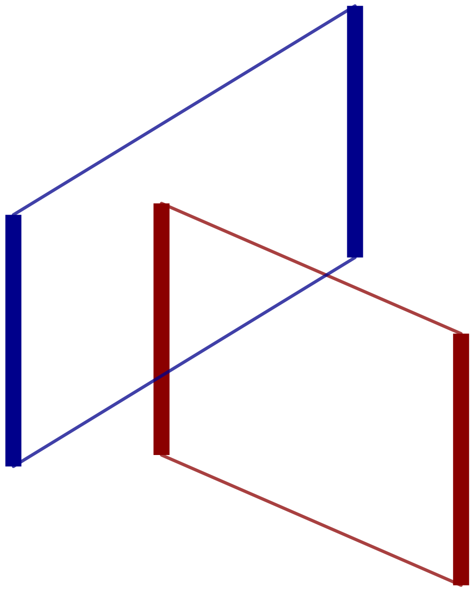


single intersection

not pseudo-disks

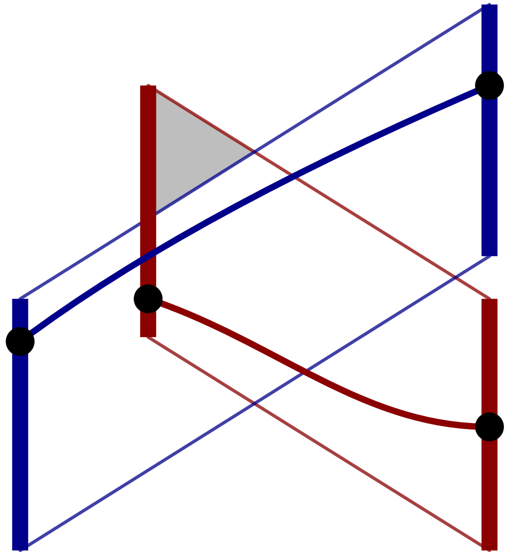
Are our tubes pseudo-disks?





single intersection

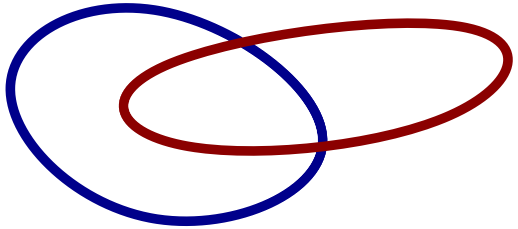
pseudo-disks!

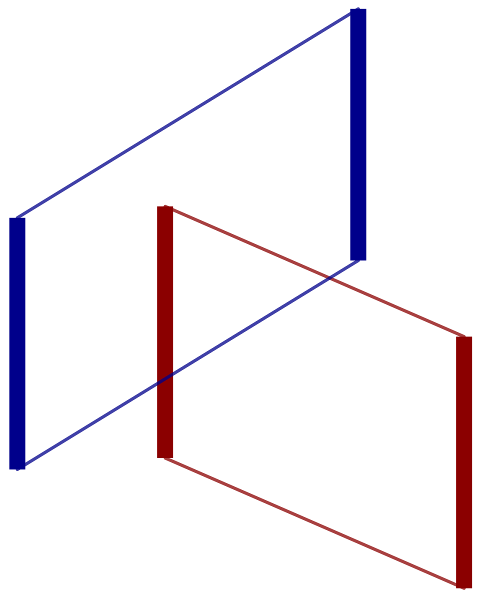


single intersection

not pseudo-disks

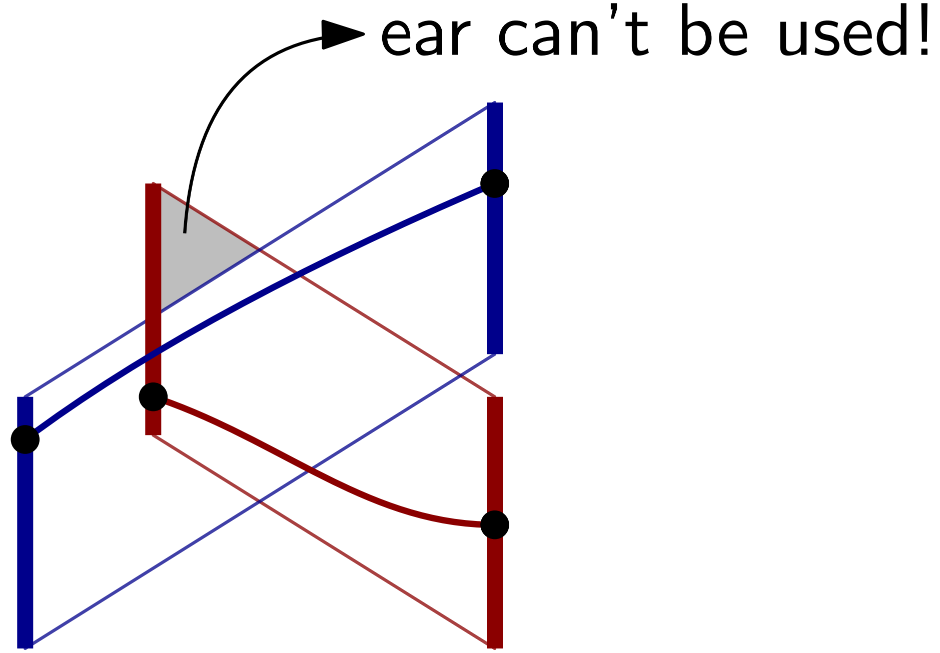
Are our tubes pseudo-disks?





single intersection

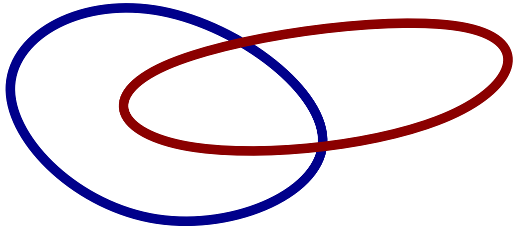
pseudo-disks!

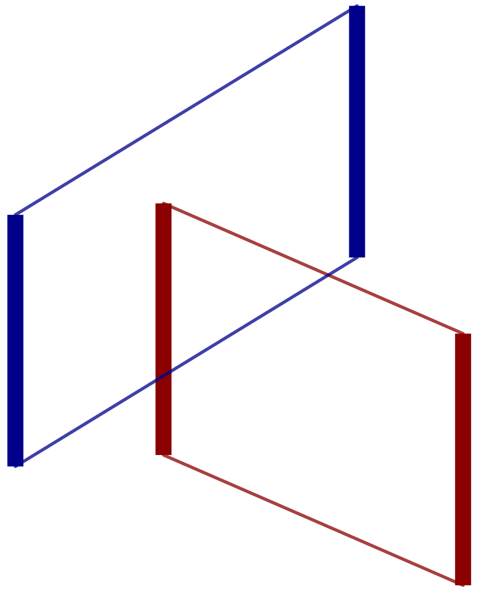


single intersection

not pseudo-disks

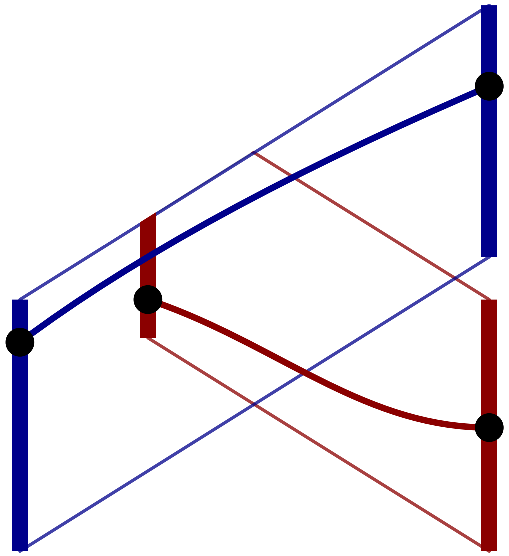
Are our tubes pseudo-disks?





single intersection

pseudo-disks!

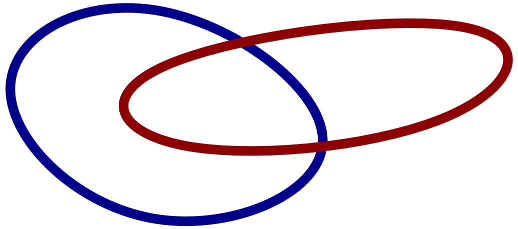


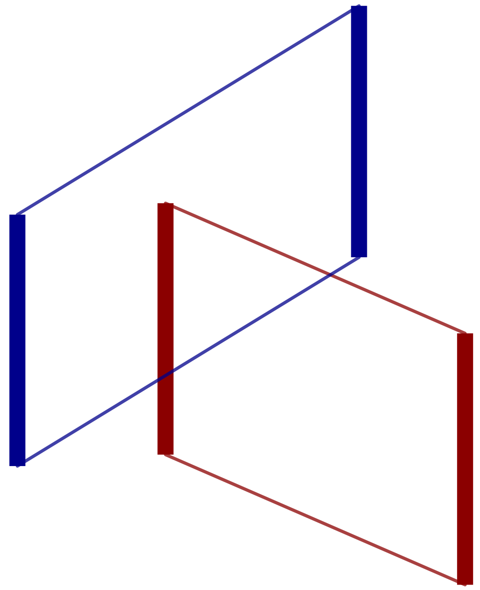
single intersection

~~not pseudo-disks~~

pseudo-disks!

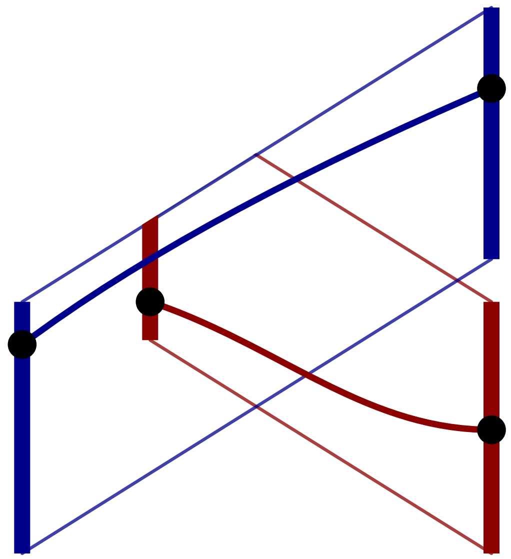
Are our tubes pseudo-disks?





single intersection

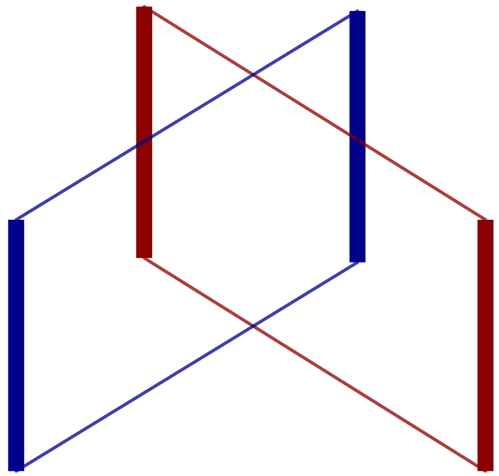
pseudo-disks!



single intersection

not pseudo-disks

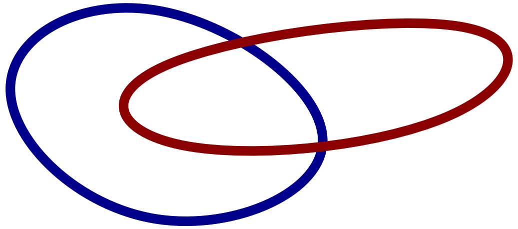
pseudo-disks!

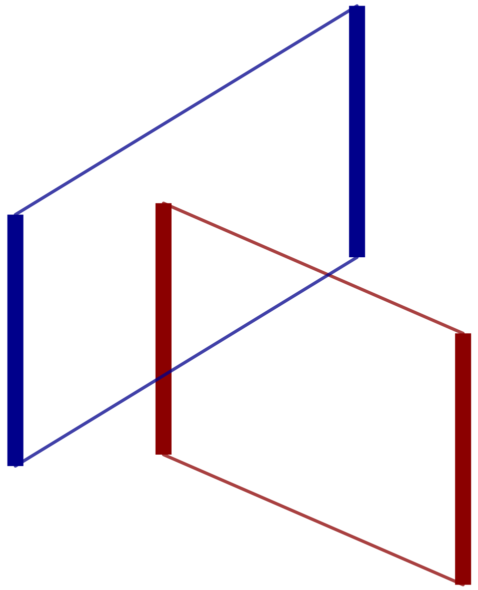


double intersection

not pseudo-disks

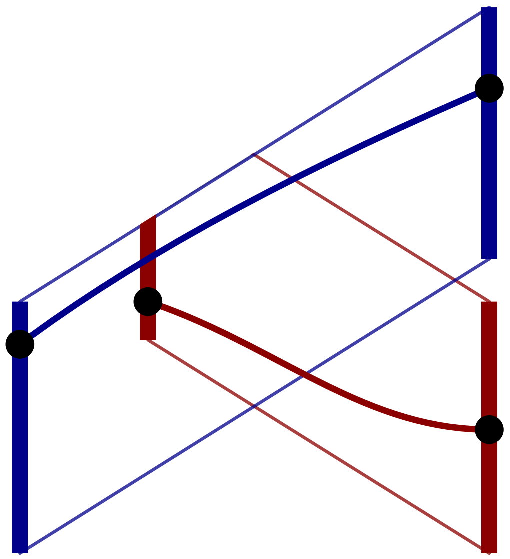
Are our tubes pseudo-disks?





single intersection

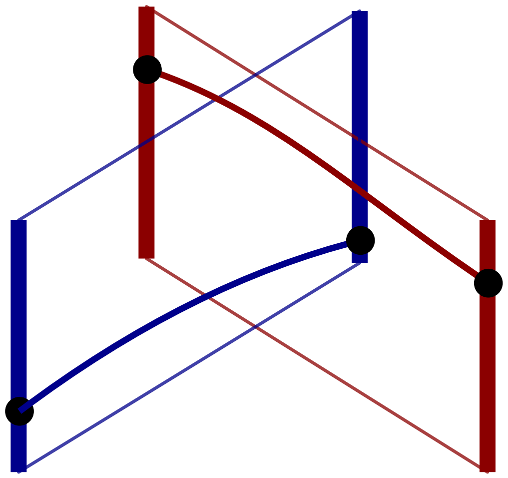
pseudo-disks!



single intersection

~~not pseudo-disks~~

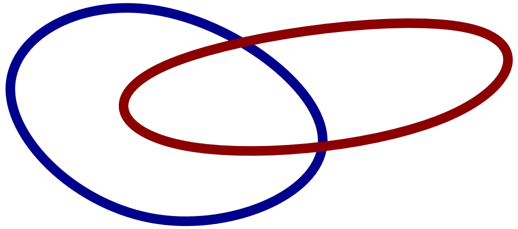
pseudo-disks!

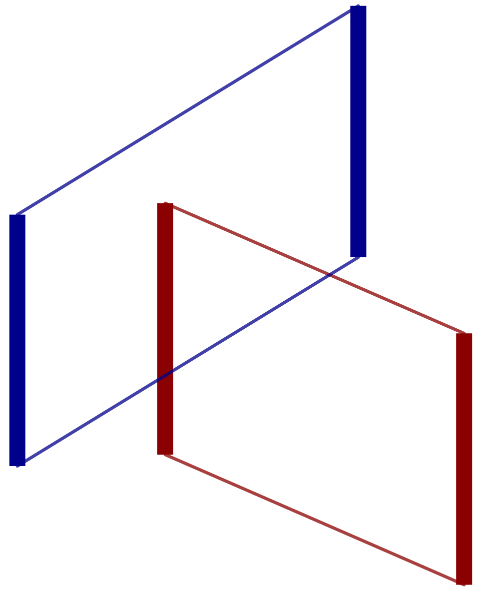


double intersection

not pseudo-disks

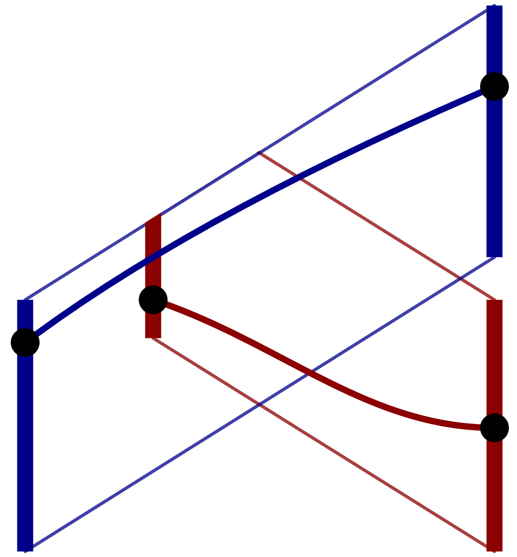
Are our tubes pseudo-disks?





single intersection

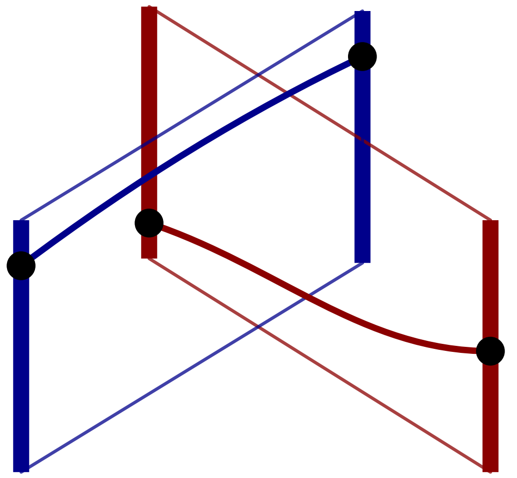
pseudo-disks!



single intersection

~~not pseudo-disks~~

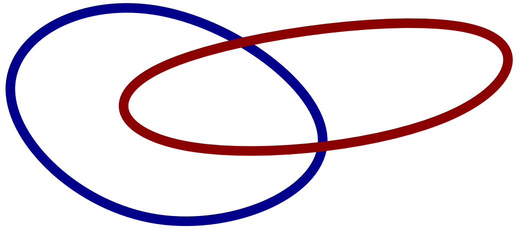
pseudo-disks!

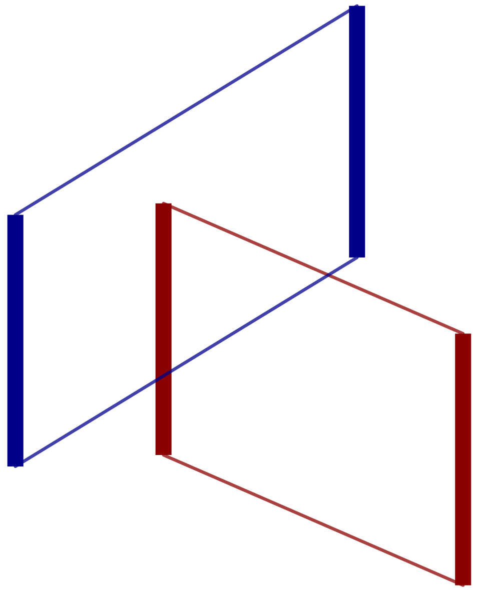


double intersection

not pseudo-disks

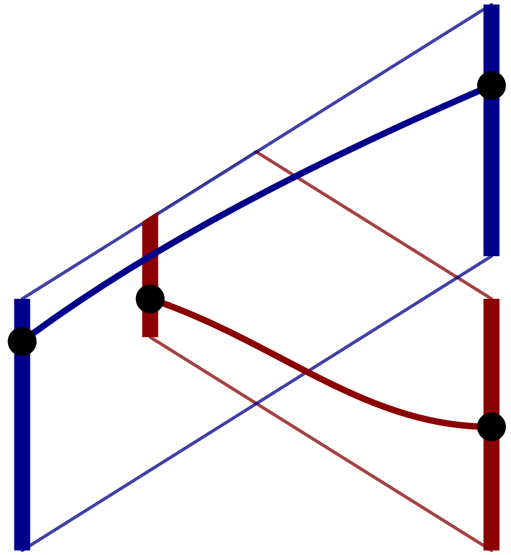
Are our tubes pseudo-disks?





single intersection

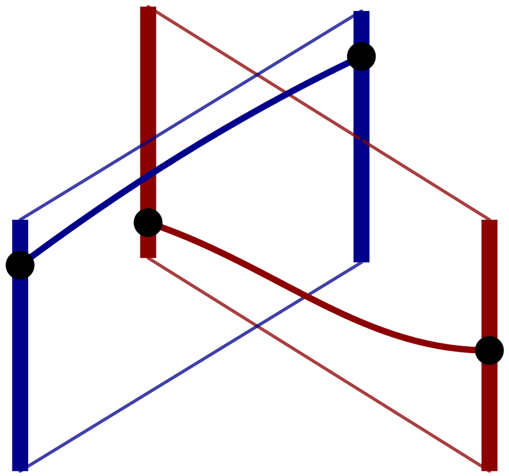
pseudo-disks!



single intersection

~~not pseudo-disks~~

pseudo-disks!

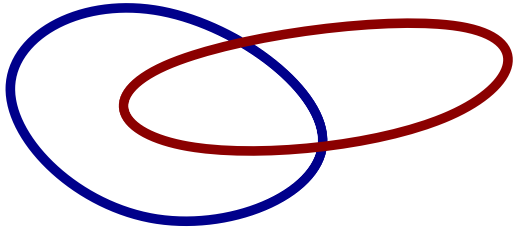


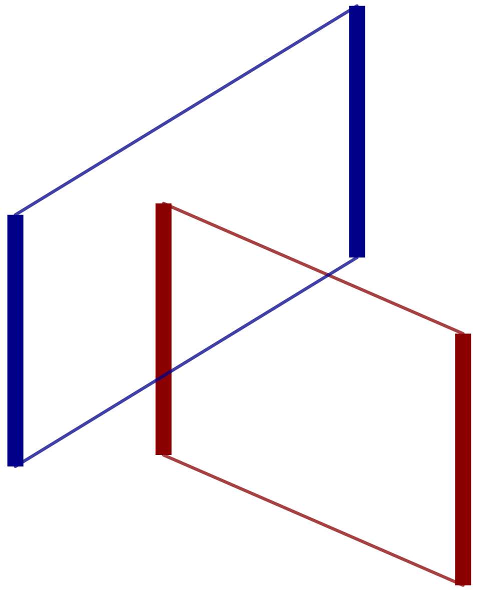
double intersection

not pseudo-disks

can't cut off ears

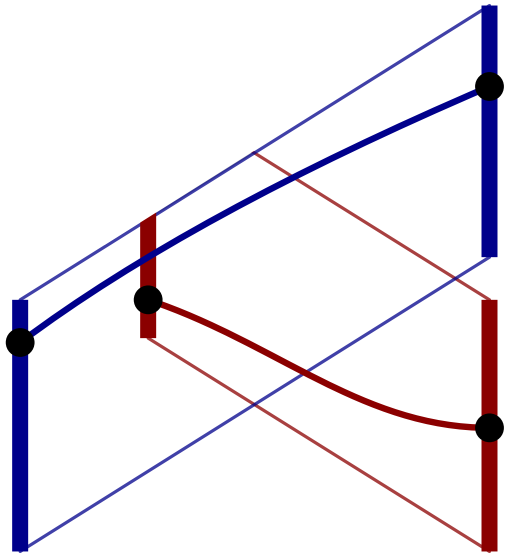
Are our tubes pseudo-disks?





single intersection

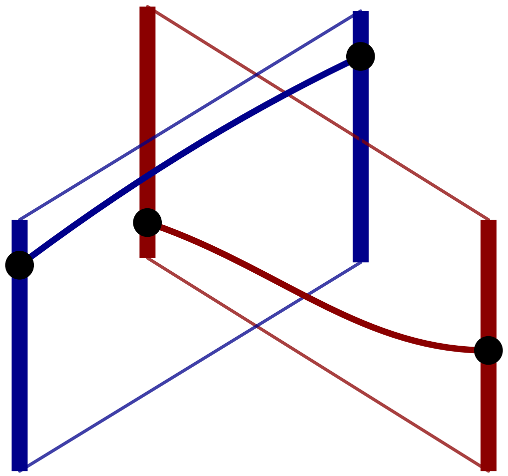
pseudo-disks!



single intersection

~~not pseudo-disks~~

pseudo-disks!



double intersection

We assume there are no double intersections

If no double intersections

Algorithm: while there is pair of tubes not pseudo-disks, cut off ear.

If no double intersections

Algorithm: while there is pair of tubes not pseudo-disks, cut off ear.

Result: set of pruned tubes that are pseudo-disks

If no double intersections

Algorithm: while there is pair of tubes not pseudo-disks, cut off ear.

Result: set of pruned tubes that are pseudo-disks

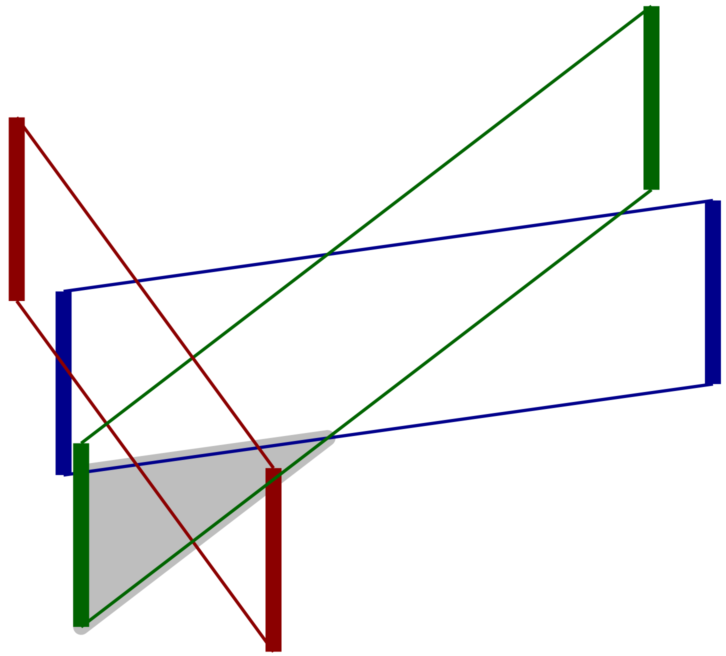
... then there is always a solution?

If no double intersections

Algorithm: while there is pair of tubes not pseudo-disks, cut off ear.

Result: set of pruned tubes that are pseudo-disks

... then there is always a solution?

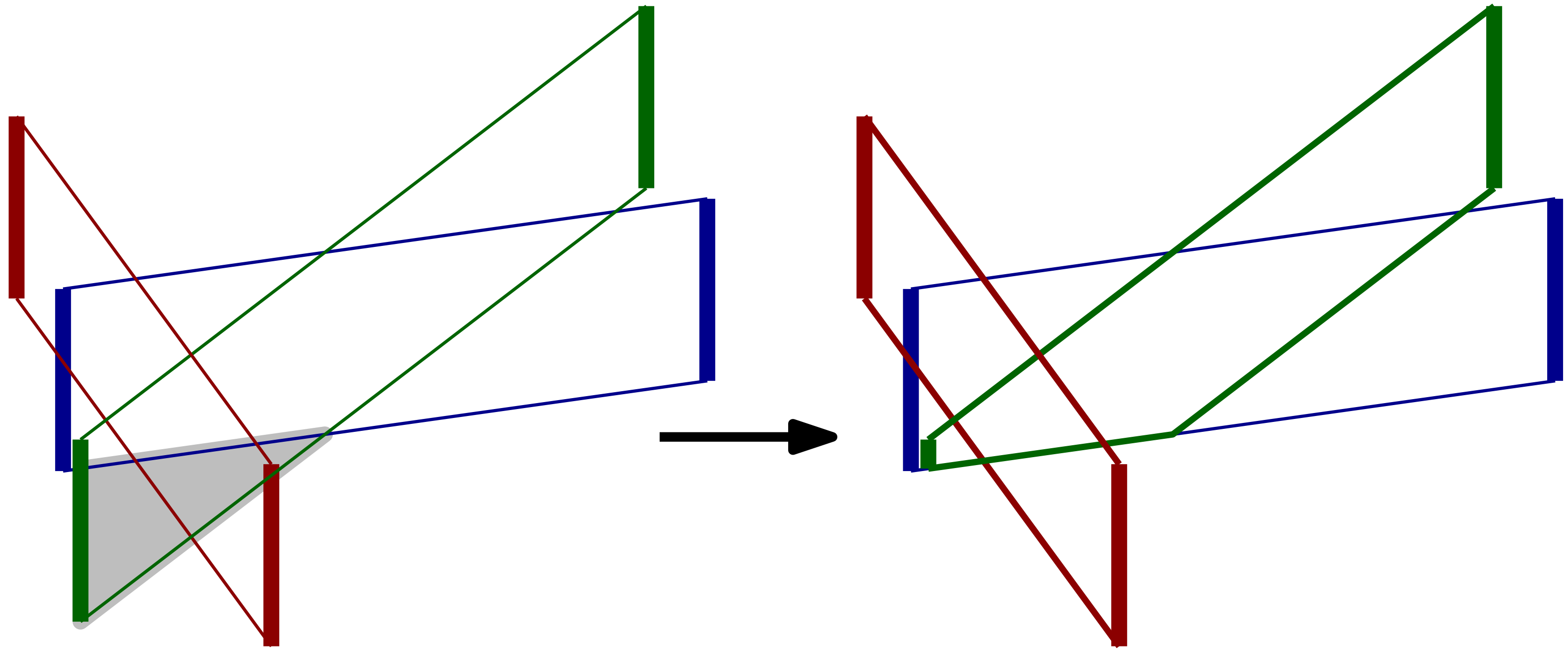


If no double intersections

Algorithm: while there is pair of tubes not pseudo-disks, cut off ear.

Result: set of pruned tubes that are pseudo-disks

... then there is always a solution?

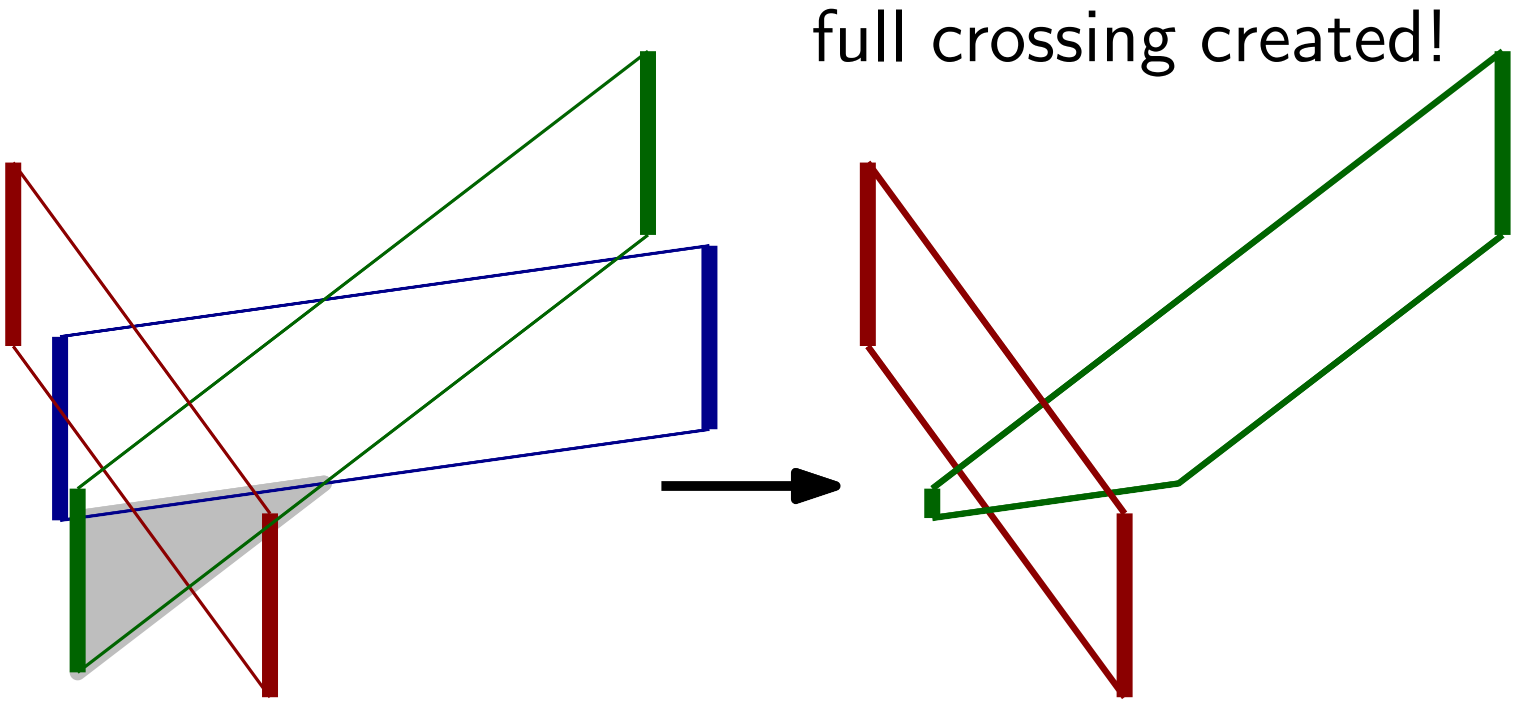


If no double intersections

Algorithm: while there is pair of tubes not pseudo-disks, cut off ear.

Result: set of pruned tubes that are pseudo-disks

... then there is always a solution?



If no double intersections

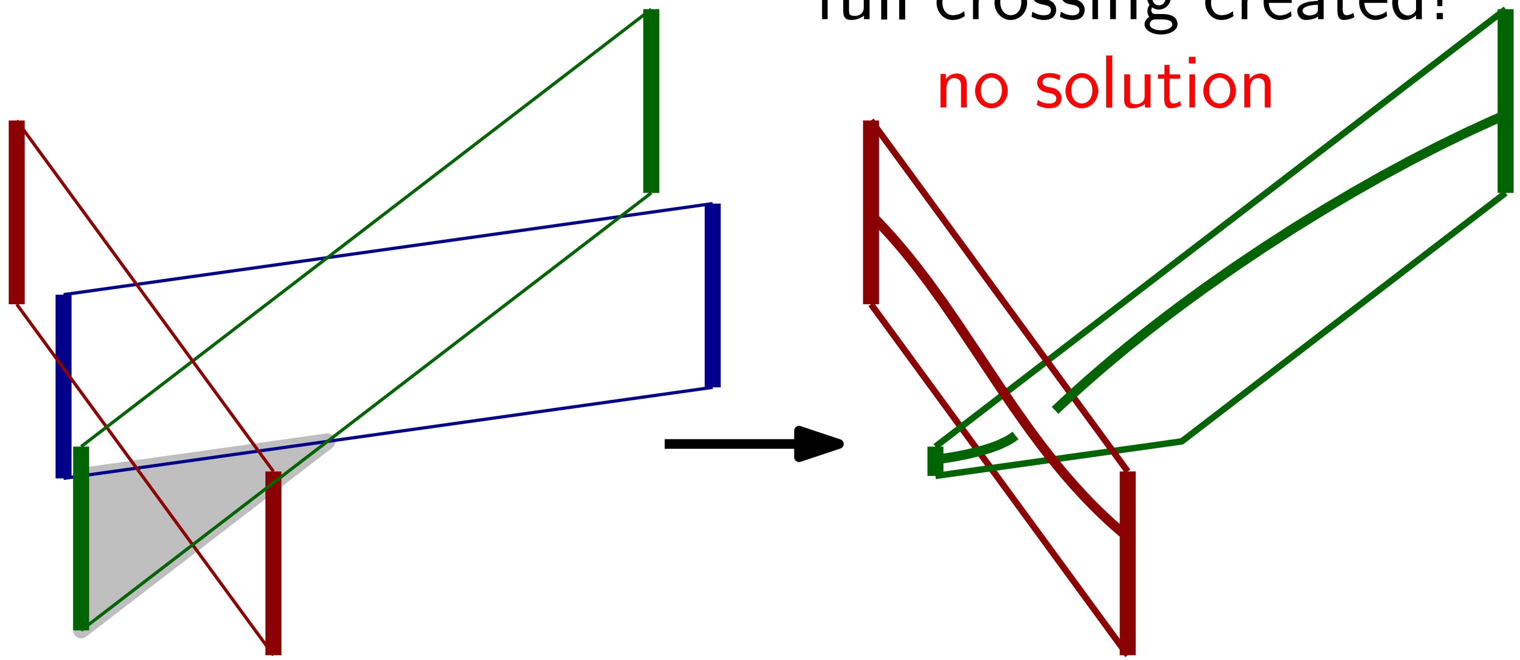
Algorithm: while there is pair of tubes not pseudo-disks, cut off ear.

Result: set of pruned tubes that are pseudo-disks

... then there is always a solution?

full crossing created!

no solution

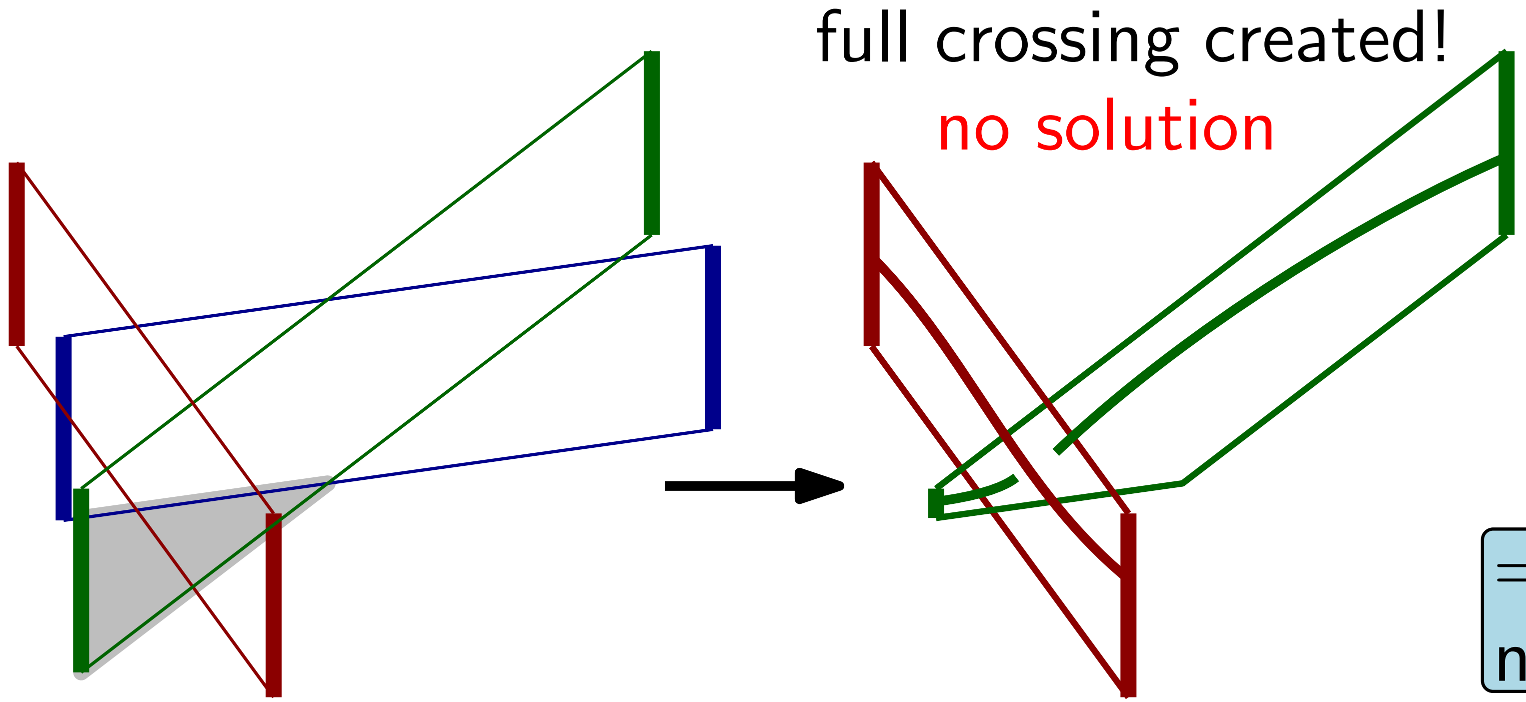


If no double intersections

Algorithm: while there is pair of tubes not pseudo-disks, cut off ear.

Result: set of pruned tubes that are pseudo-disks

... then there is always a solution?



⇒ original instance had no solution either

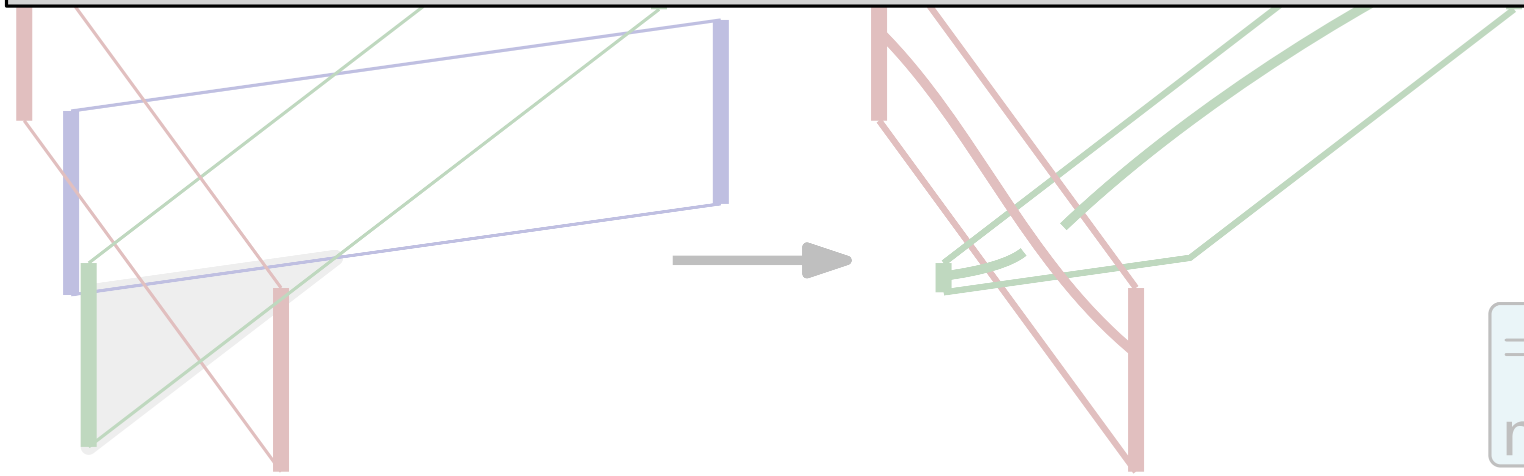
If no double intersections

Algorithm: while there is pair of tubes not pseudo-disks, cut off ear.

Result: set of pruned tubes that are pseudo-disks

... then there is always a solution?

If there are no double intersections, one can determine if all the tubes can be connected using arbitrary paths in polynomial time

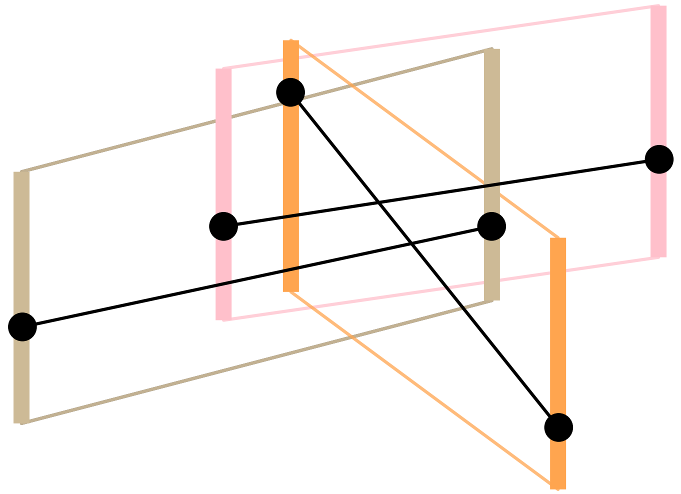


⇒ original instance had no solution either

Summary

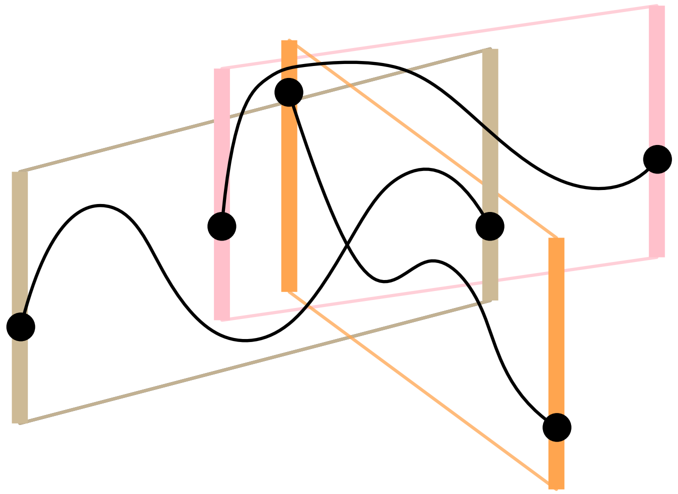
Summary

straight-line paths



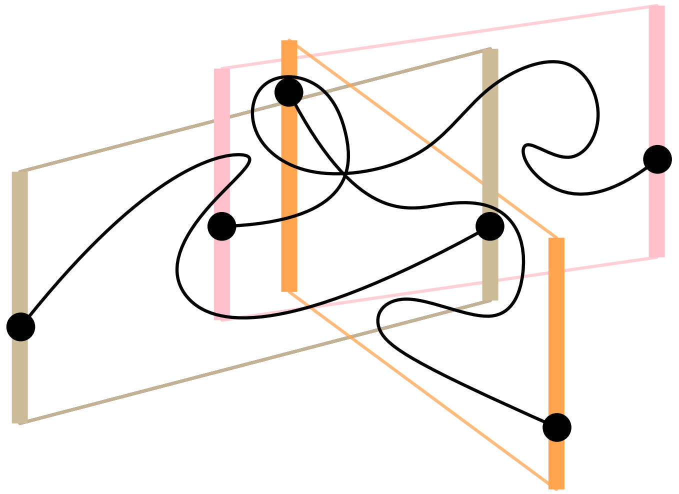
NP-complete

(x-)monotone paths



Polynomial

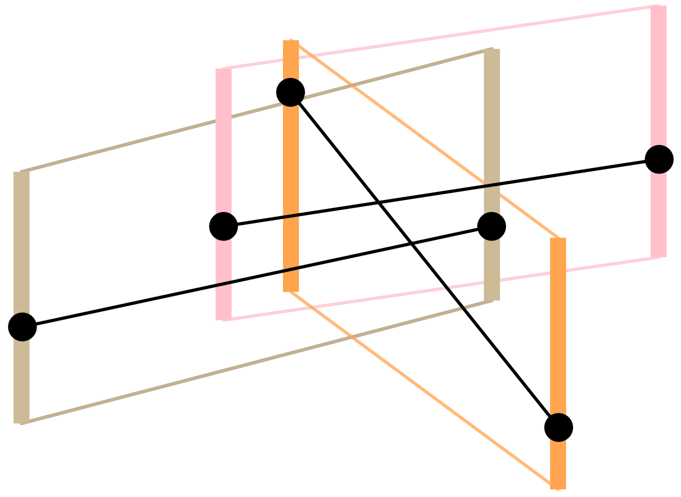
arbitrary paths



Polynomial if no double intersections

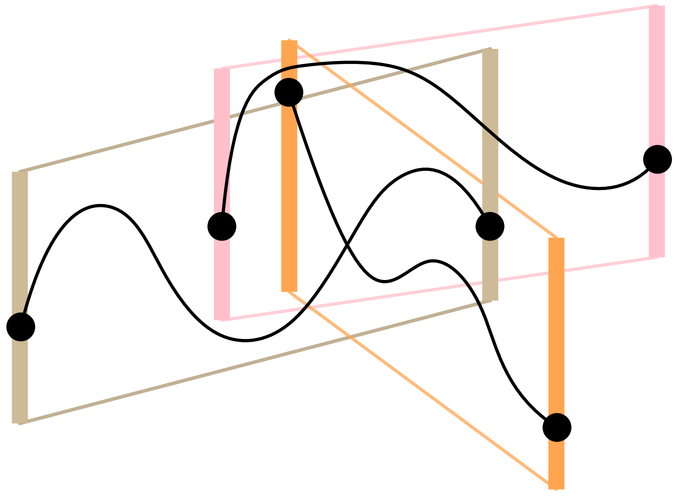
Summary

straight-line paths



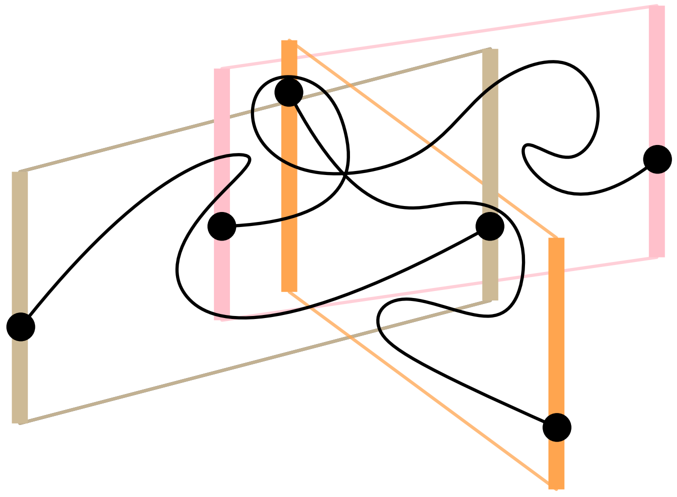
NP-complete

(x-)monotone paths



Polynomial

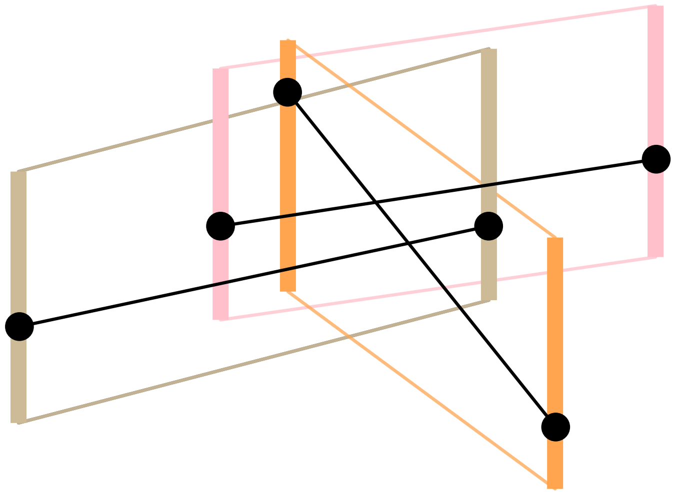
arbitrary paths



Polynomial if no double intersections

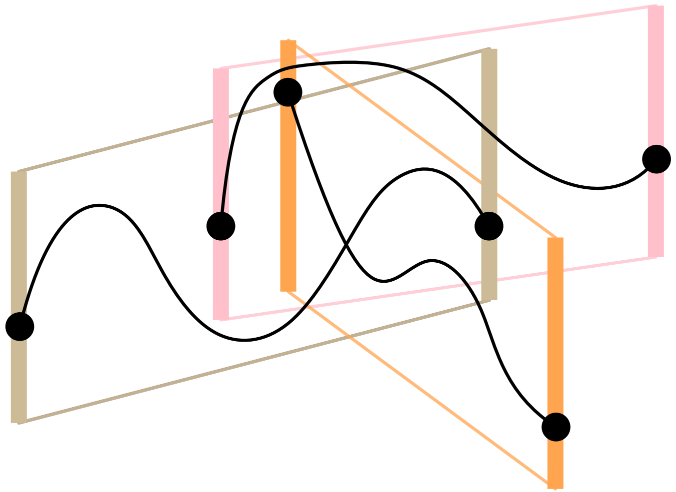
Summary

straight-line paths



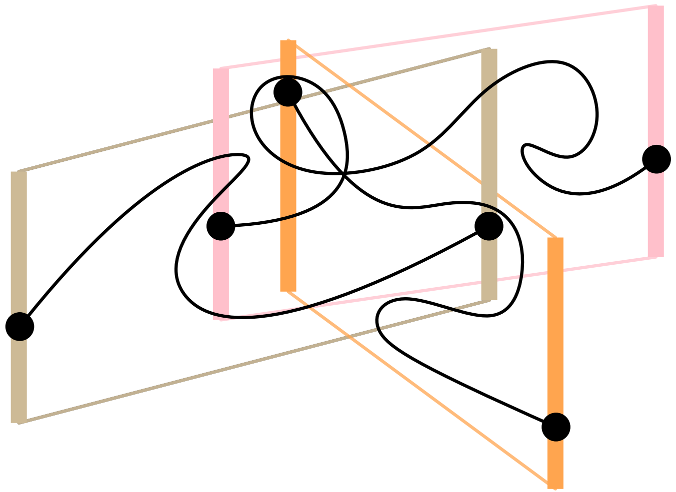
NP-complete

(x-)monotone paths



Polynomial

arbitrary paths

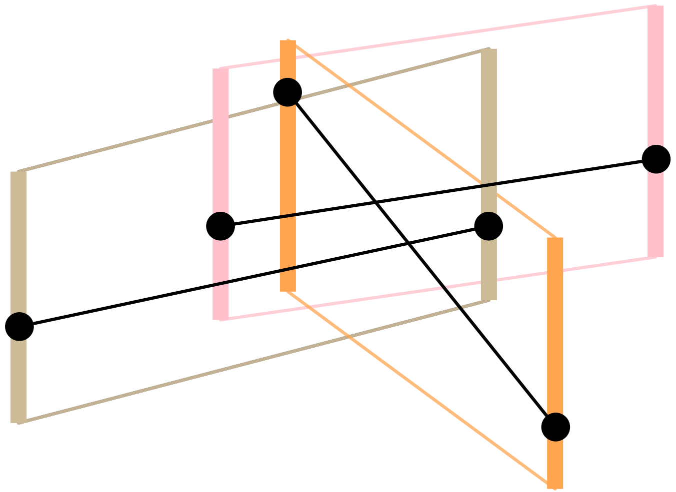


Polynomial if no double intersections

Is it also polynomial if there are double intersections?

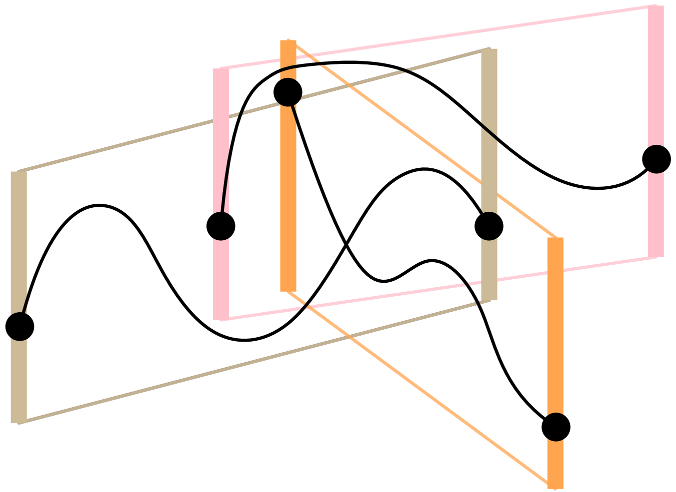
Summary

straight-line paths



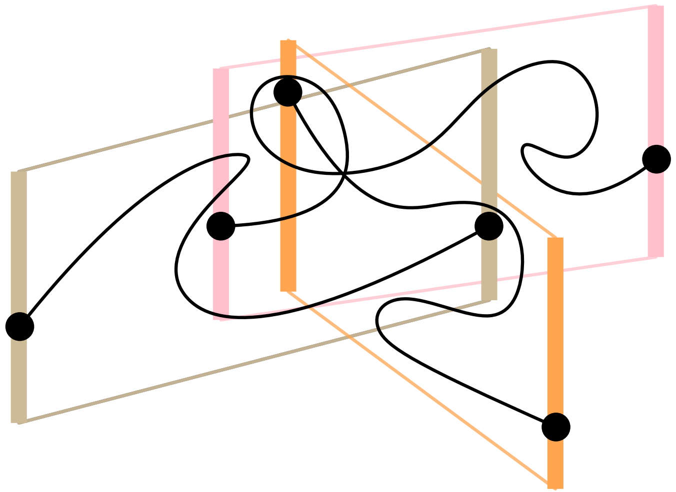
NP-complete

(x-)monotone paths



Polynomial

arbitrary paths



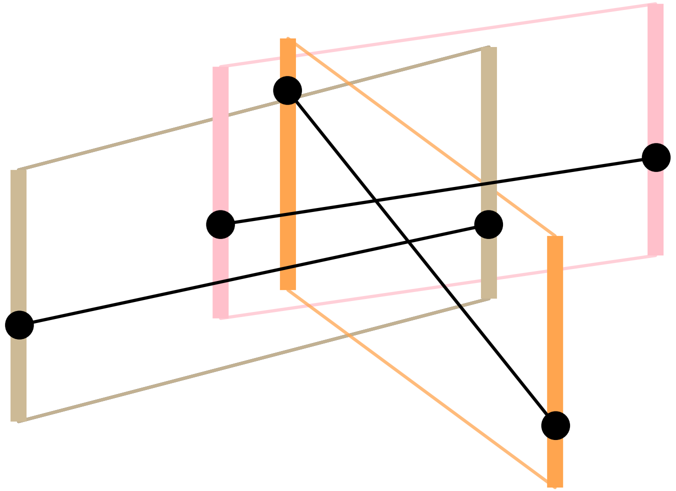
Polynomial if no double intersections

Is it also polynomial if there are double intersections?

We conjecture the answer is **Yes**

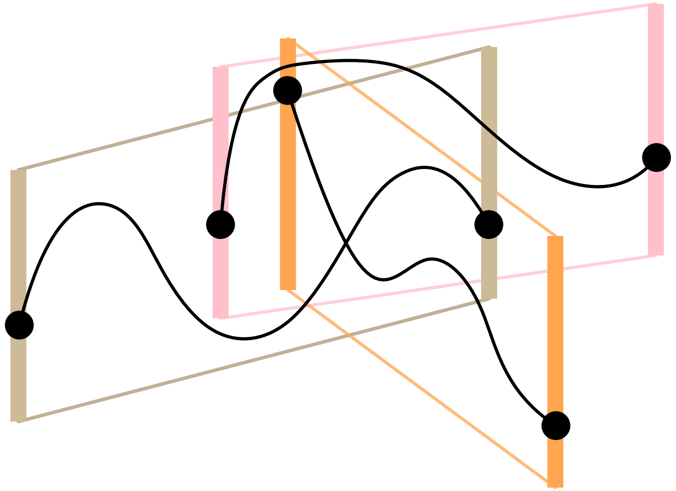
Our results

straight line paths



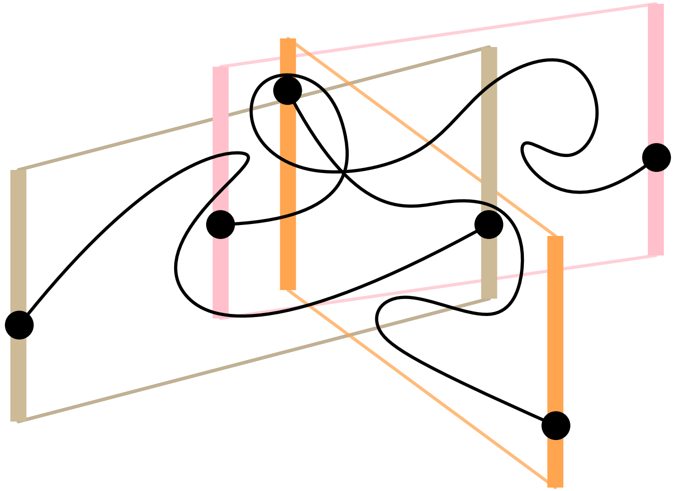
NP-complete

(x-)monotone paths



Polynomial

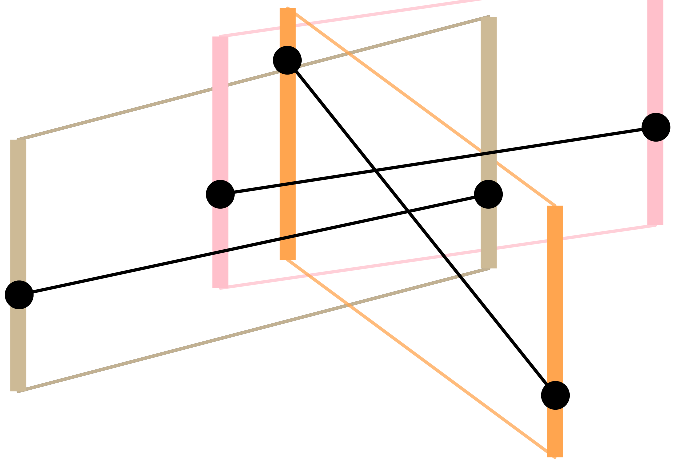
arbitrary paths



Polynomial under certain assumptions

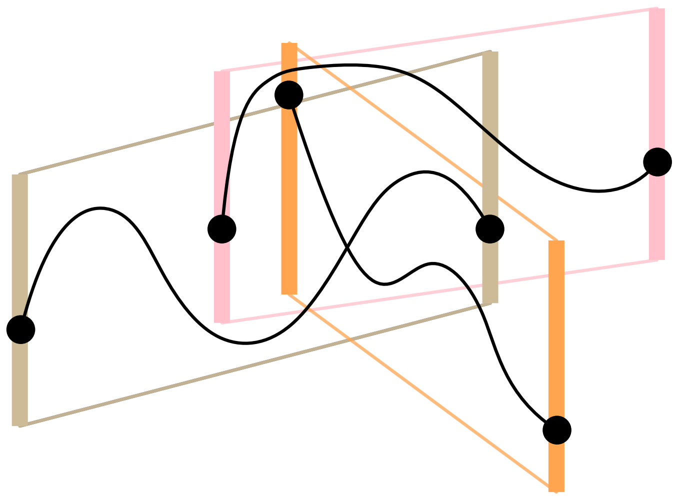
Our results

straight line paths



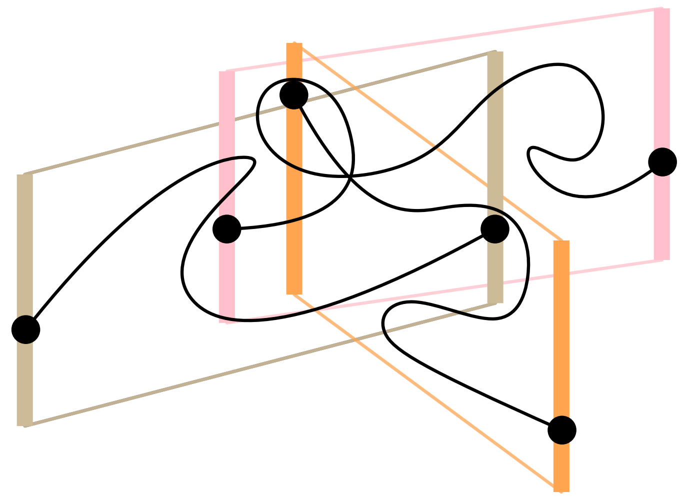
NP-complete

(x-)monotone paths



Polynomial

arbitrary paths



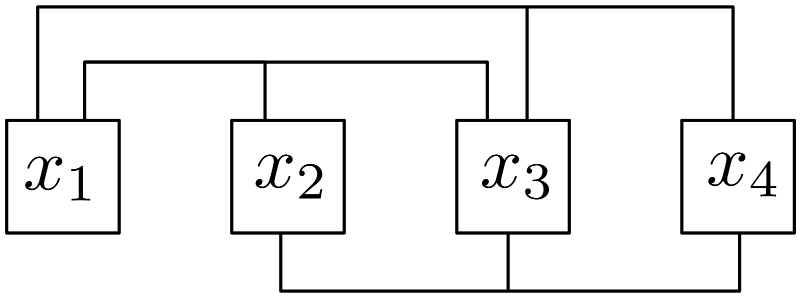
Polynomial under certain assumptions

Straight line paths

Given n tubes defined by unit vertical segments, deciding if the tubes can be connected with straight line segments is NP-Complete

Straight line paths

Given n tubes defined by unit vertical segments, deciding if the tubes can be connected with straight line segments is NP-Complete

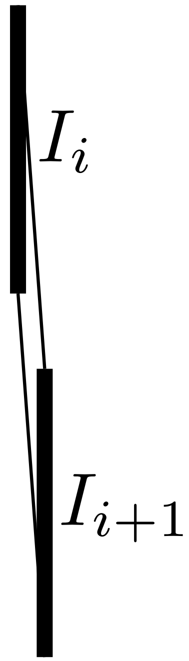


Reduction from Rectilinear Planar 3-SAT

Straight line paths

Given n tubes defined by unit vertical segments, deciding if the tubes can be connected with straight line segments is NP-Complete

Two useful gadgets:

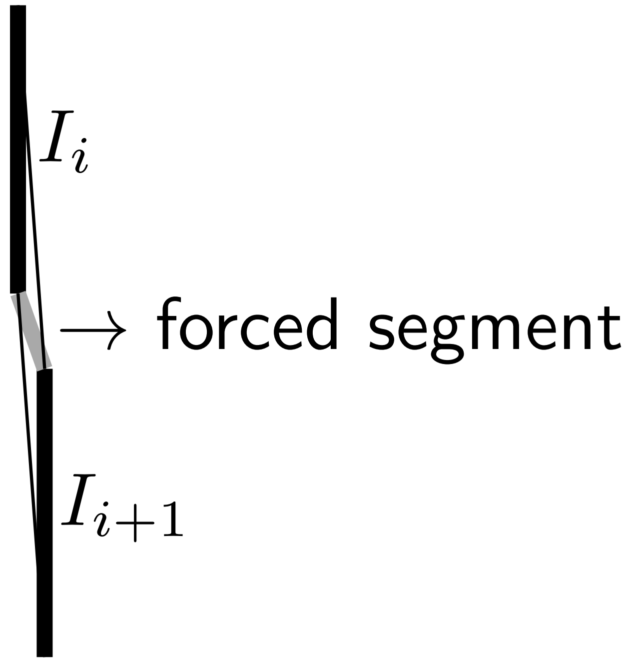


Blocker

Straight line paths

Given n tubes defined by unit vertical segments, deciding if the tubes can be connected with straight line segments is NP-Complete

Two useful gadgets:

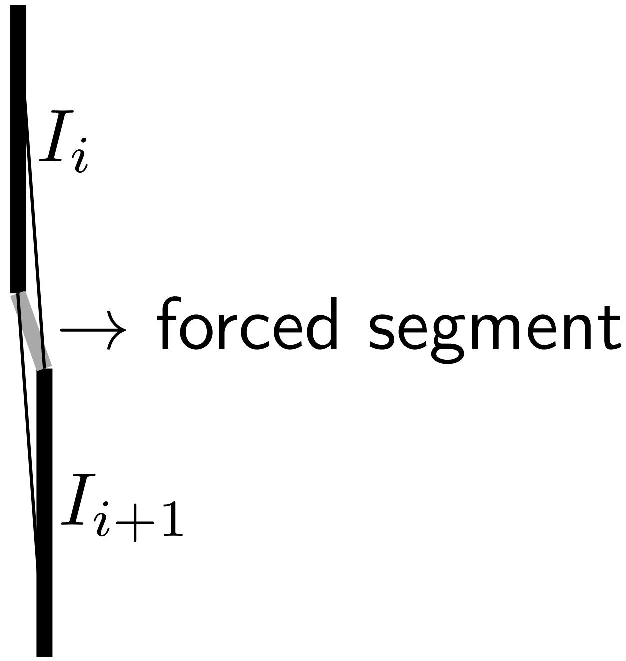


Blocker

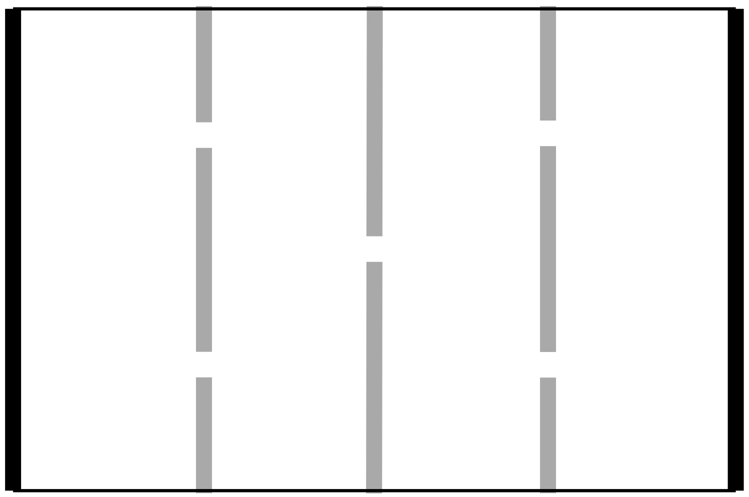
Straight line paths

Given n tubes defined by unit vertical segments, deciding if the tubes can be connected with straight line segments is NP-Complete

Two useful gadgets:



Blocker

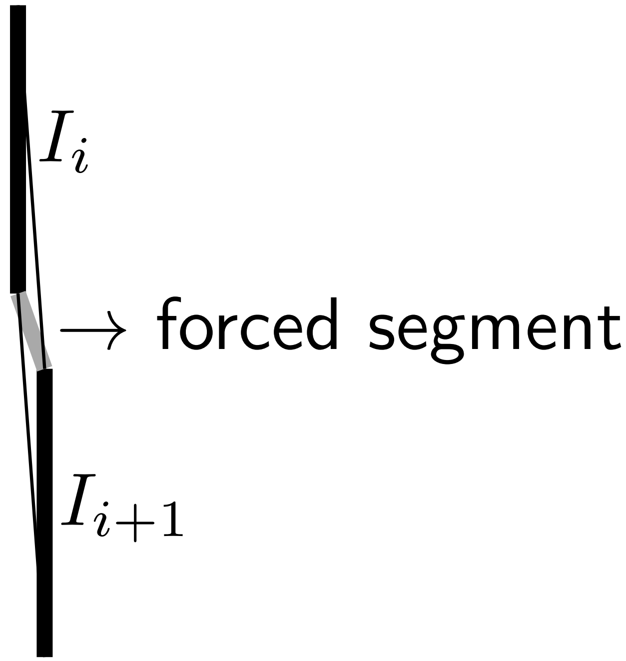


Variable

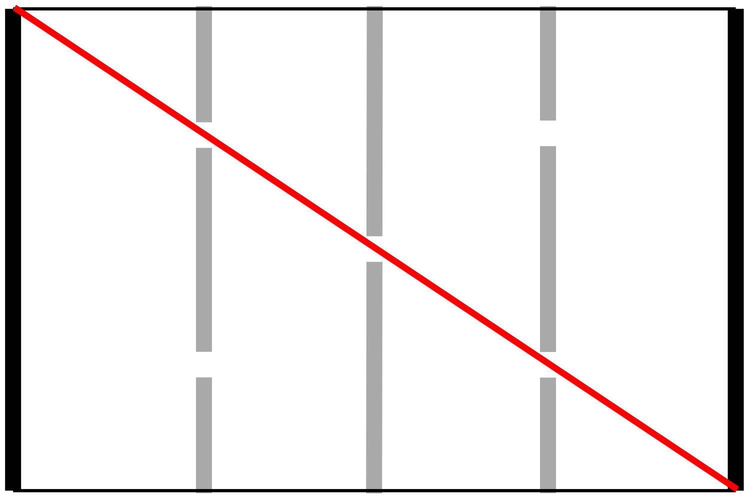
Straight line paths

Given n tubes defined by unit vertical segments, deciding if the tubes can be connected with straight line segments is NP-Complete

Two useful gadgets:



Blocker

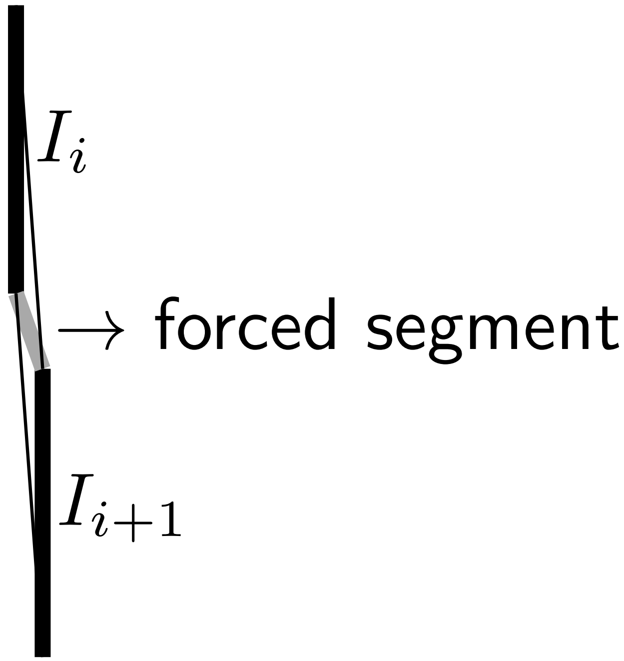


Variable

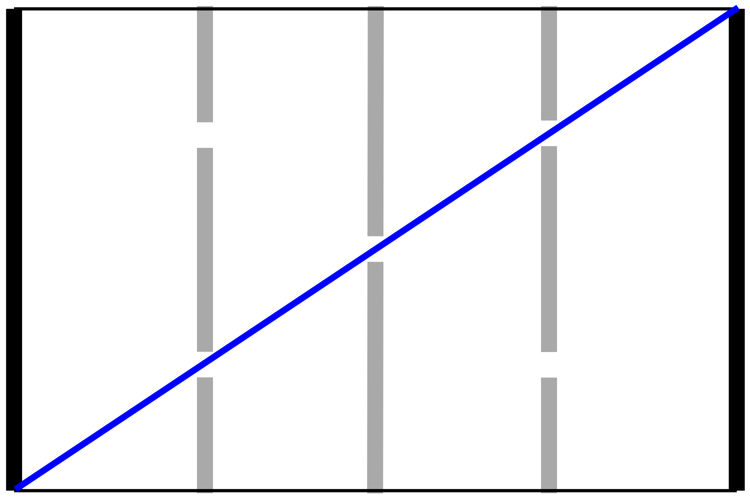
Straight line paths

Given n tubes defined by unit vertical segments, deciding if the tubes can be connected with straight line segments is NP-Complete

Two useful gadgets:



Blocker

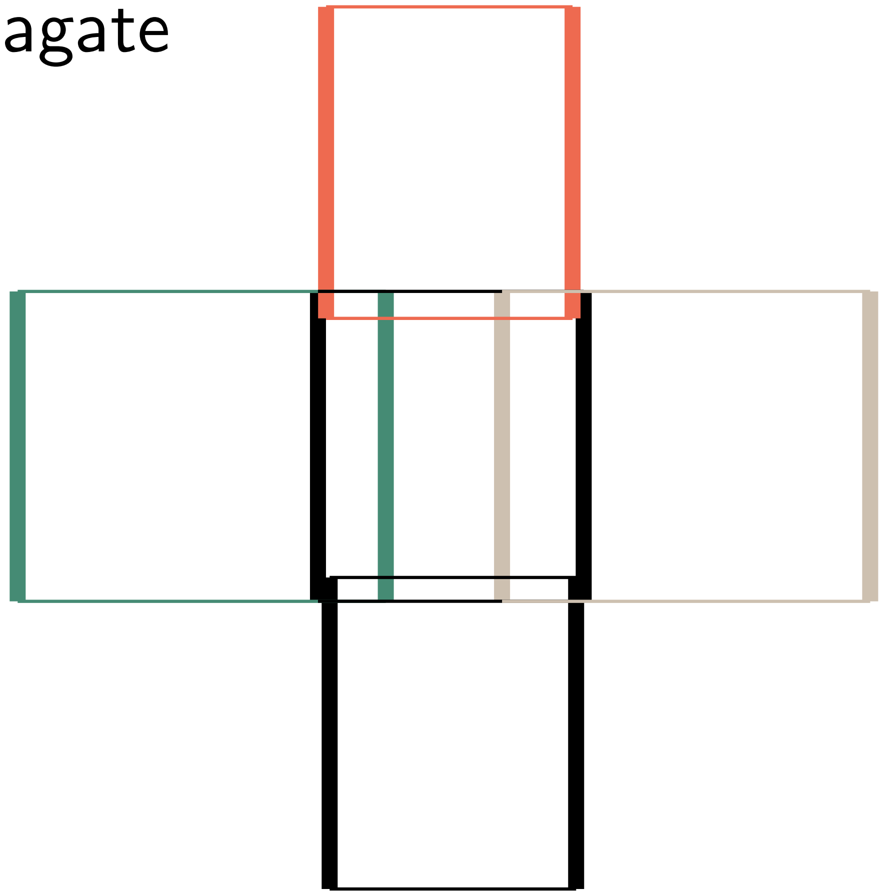


Variable

Straight line paths

Given n tubes defined by unit vertical segments, deciding if the tubes can be connected with straight line segments is NP-Complete

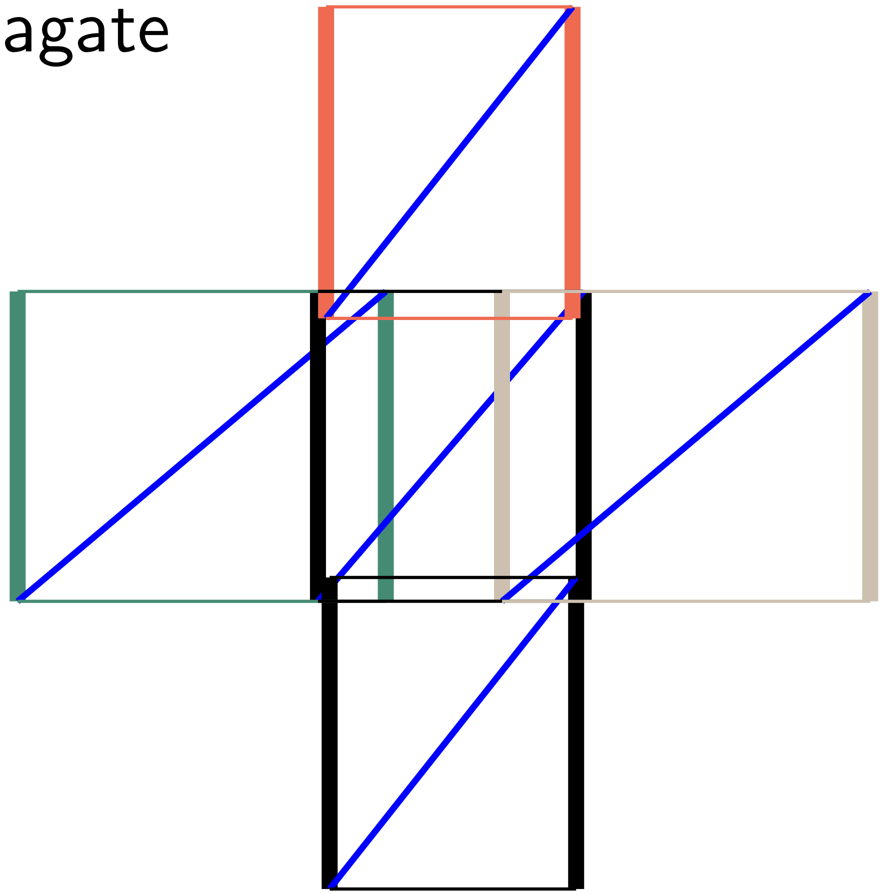
We can now propagate truth values...



Straight line paths

Given n tubes defined by unit vertical segments, deciding if the tubes can be connected with straight line segments is NP-Complete

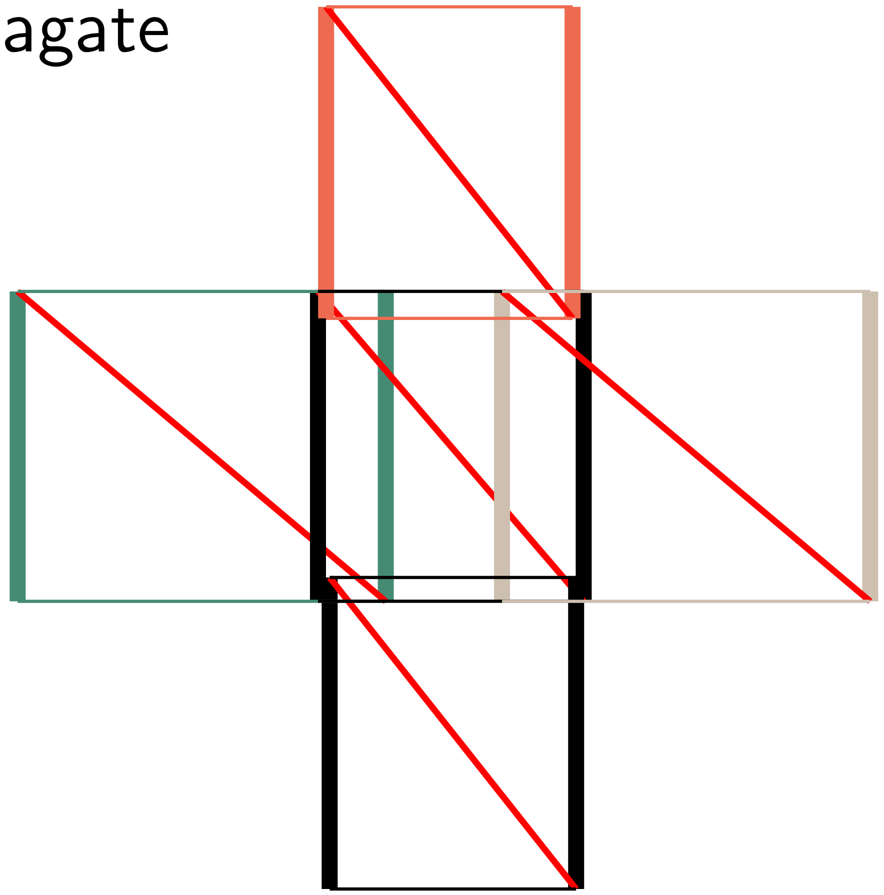
We can now propagate truth values...



Straight line paths

Given n tubes defined by unit vertical segments, deciding if the tubes can be connected with straight line segments is NP-Complete

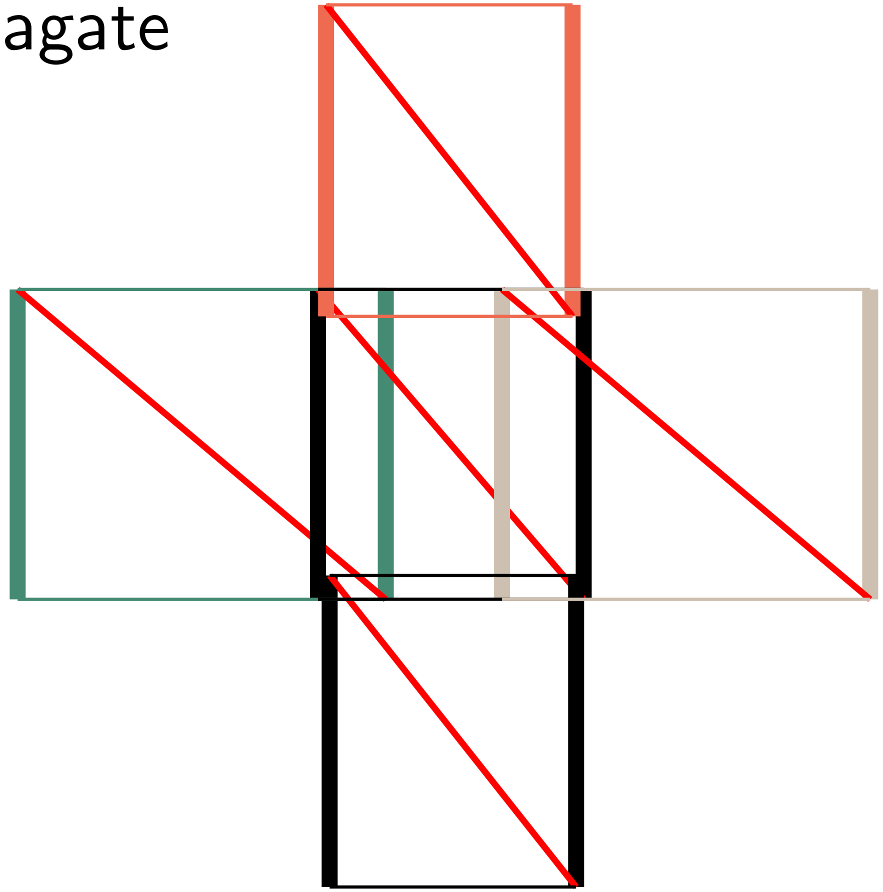
We can now propagate truth values...



Straight line paths

Given n tubes defined by unit vertical segments, deciding if the tubes can be connected with straight line segments is NP-Complete

We can now propagate truth values...



...as well as representing negations and clauses, so that the reduction works