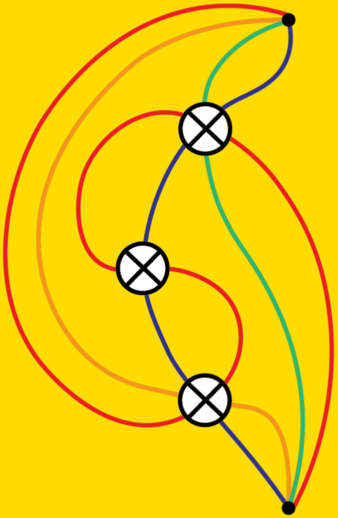
Cross-cap drawings and signed reversal distance

Niloufar Fuladi

Joint work with:
Arnaud de Mesmay
Alfredo Hubard

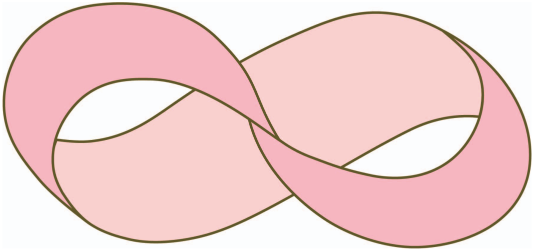


Journée-séminaire de combinatoire
Université Sorbonne Paris Nord
10 Dec 2024

Möbius band

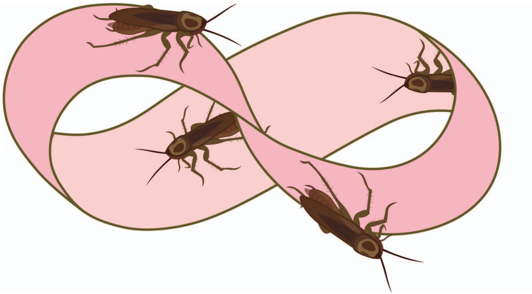
- The surface we obtain after gluing two ends of a strip of paper after a **half-twist** is a **Möbius band.**

Möbius band



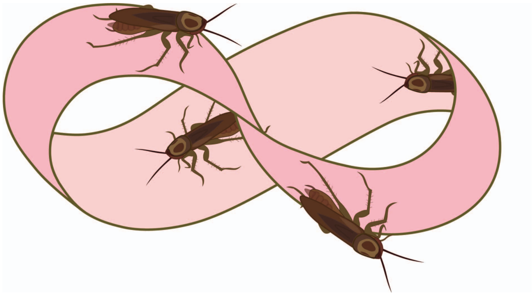
- The surface we obtain after gluing two ends of a strip of paper after a **half-twist** is a **Möbius band**.

Möbius band



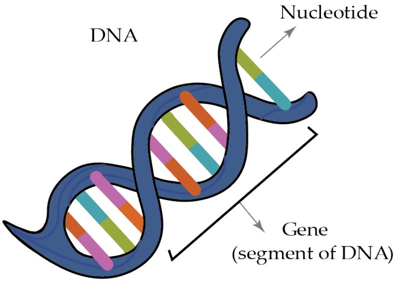
- The surface we obtain after gluing two ends of a strip of paper after a **half-twist** is a **Möbius band**.
- A Möbius band has only one side.
- Right and left on the band are not well defined.

Möbius band

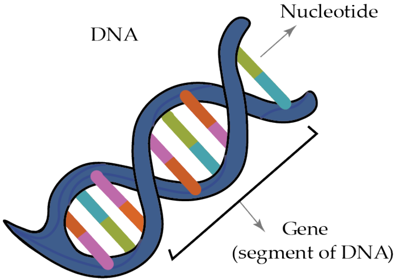


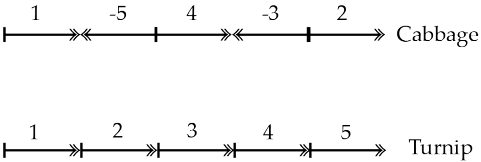
- The surface we obtain after gluing two ends of a strip of paper after a **half-twist** is a **Möbius band**.
- A Möbius band has only one side.
- Right and left on the band are not well defined.
- Simplest example of a **non-orientable surface**.

A "twist" in biology: Genome rearrangement



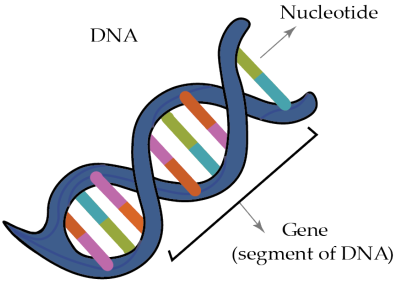
A "twist" in biology: Genome rearrangement

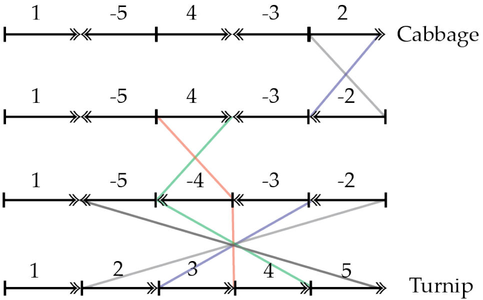




- The DNA of some species only differ by their gene sequences.

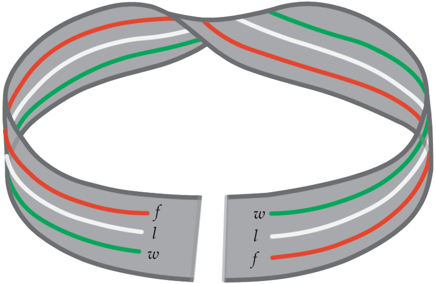
A "twist" in biology: Genome rearrangement

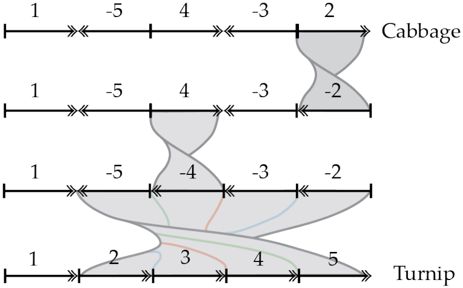




- The DNA of some species only differ by their gene sequences.
- The evolutionary distance between two species can be approximated by the number of **reversals** needed to transform one gene sequence into another.

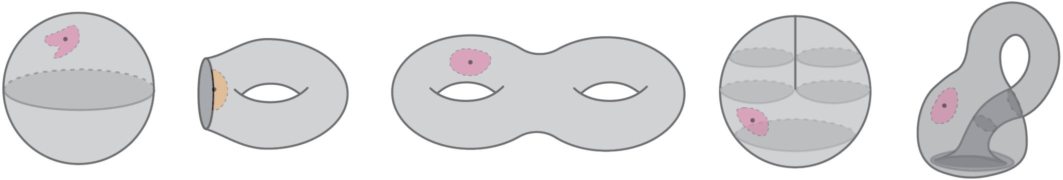
A "twist" in biology: Genome rearrangement





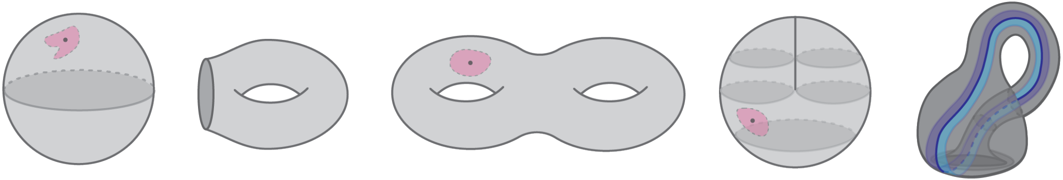
- The DNA of some species only differ by their gene sequences.
- The evolutionary distance between two species can be approximated by the number of **reversals** needed to transform one gene sequence into another.
- We use the similarity between reversals and Möbius band to solve two problems.

Surface



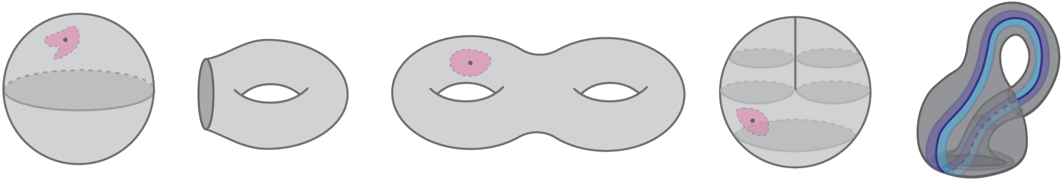
- A **surface** is a topological space that locally looks like the plane or the half plane.
- The points that locally look like the half plane comprise the **boundary** of the surface.

Surface



- A **surface** is a topological space that locally looks like the plane or the half plane.
- The points that locally look like the half plane comprise the **boundary** of the surface.
- A surface is **non-orientable** if and only if it contains a Möbius band, otherwise it is **orientable**.

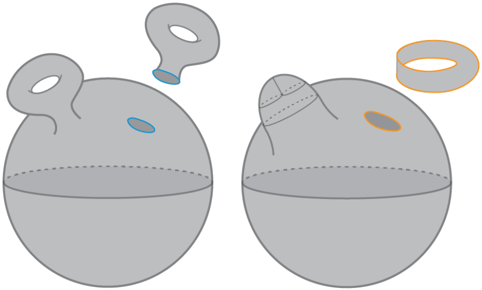
Surface



- A **surface** is a topological space that locally looks like the plane or the half plane.
- The points that locally look like the half plane comprise the **boundary** of the surface.
- A surface is **non-orientable** if and only if it contains a Möbius band, otherwise it is **orientable**.
- Two surfaces are **homeomorphic** if one can be transformed continuously to the other without cutting or gluing.

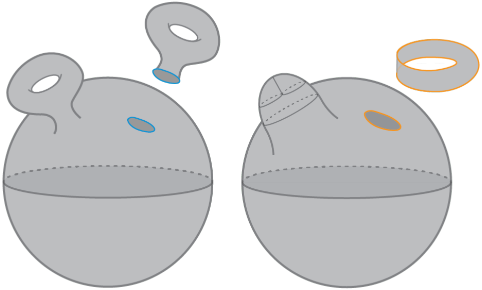
Surface

- We can obtain all surfaces by cutting disks from a sphere and attaching **handles** and **cross-caps**.
- A surface obtained by only attaching handles is **orientable**.



Surface

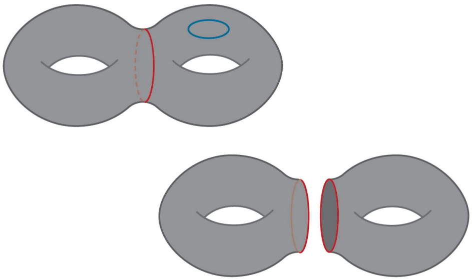
- We can obtain all surfaces by cutting disks from a sphere and attaching **handles** and **cross-caps**.
- A surface obtained by only attaching handles is **orientable**.
- All surfaces can be classified by:
  1. genus
  2. number of boundaries
  3. orientability



Types of curves on a surface

- Two curves are of the **same type** if there exists a homeomorphism of the surface that maps one to the other.

→ A curve is **separating** if it cuts the surface into two connected components.

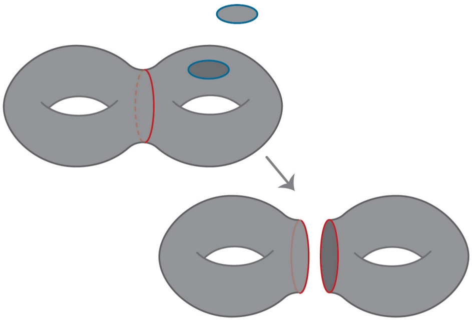


Types of curves on a surface

- Two curves are of the **same type** if there exists a homeomorphism of the surface that maps one to the other.

→ A curve is **separating** if it cuts the surface into two connected components.

→ A separating curve is **contractible** if it separates a disk from the surface.

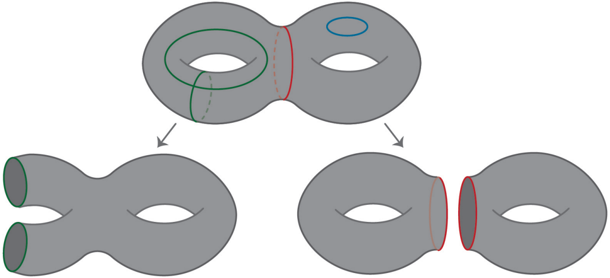


Types of curves on a surface

- Two curves are of the **same type** if there exists a homeomorphism of the surface that maps one to the other.

→ A curve is **separating** if it cuts the surface into two connected components.

→ A separating curve is **contractible** if it separates a disk from the surface.



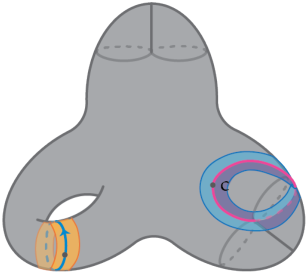
Types of curves on a surface

- Two curves are of the **same type** if there exists a homeomorphism of the surface that maps one to the other.

→ A curve is **separating** if it cuts the surface into two connected components.

→ A separating curve is **contractible** if it separates a disk from the surface.

→ A curve is **one-sided** (**two-sided**) if it has a neighborhood homeomorphic to the Möbius band (annulus).



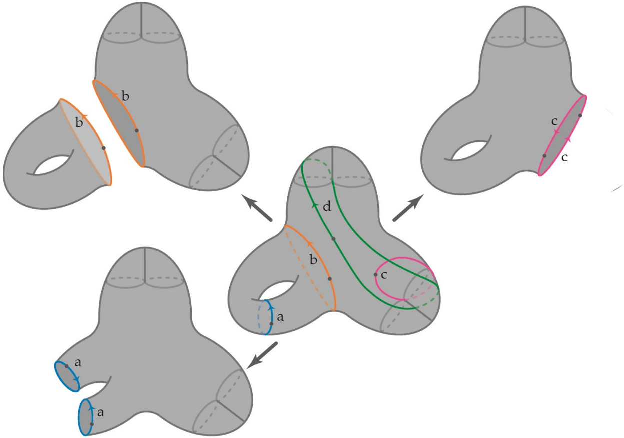
Types of curves on a surface

- Two curves are of the **same type** if there exists a homeomorphism of the surface that maps one to the other.

→ A curve is **separating** if it cuts the surface into two connected components.

→ A separating curve is **contractible** if it separates a disk from the surface.

→ A curve is **one-sided** (**two-sided**) if it has a neighborhood homeomorphic to the Möbius band (annulus).



Types of curves on a surface

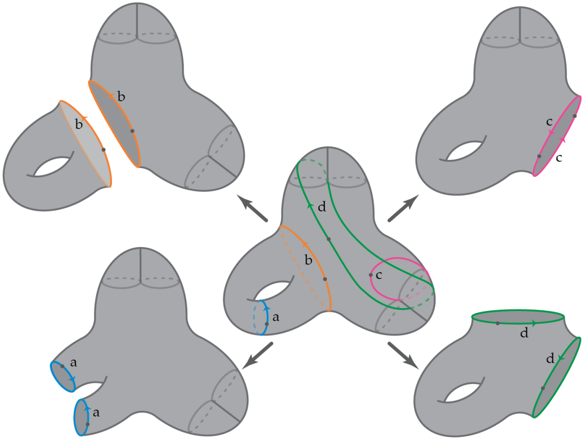
- Two curves are of the **same type** if there exists a homeomorphism of the surface that maps one to the other.

→ A curve is **separating** if it cuts the surface into two connected components.

→ A separating curve is **contractible** if it separates a disk from the surface.

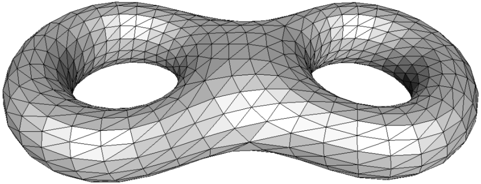
→ A curve is **one-sided** (**two-sided**) if it has a neighborhood homeomorphic to the Möbius band (annulus).

→ A curve on a non-orientable surface is **orienting** if it cuts the surface into an orientable one.



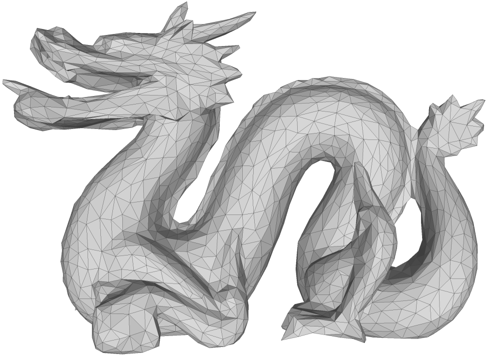
A discrete model

- Our surfaces are obtained by gluing polygonal disks.
- This can be seen as an **embedded graph** on the surface: An injective map $G \hookrightarrow S$ from a graph G to the surface S.



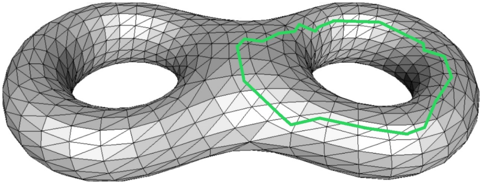
A discrete model

- Our surfaces are obtained by gluing polygonal disks.
- This can be seen as an **embedded graph** on the surface: An injective map $G \hookrightarrow S$ from a graph G to the surface S.



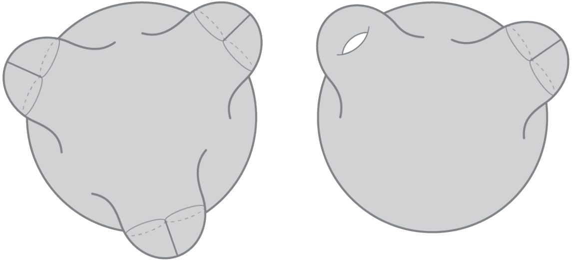
A discrete model

- Our surfaces are obtained by gluing polygonal disks.
- This can be seen as an **embedded graph** on the surface: An injective map $G \hookrightarrow S$ from a graph G to the surface S.



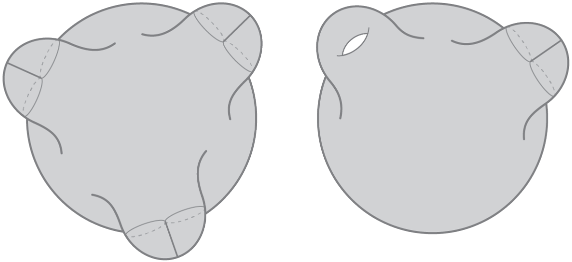
- A graph embedding induces a **discrete metric** on the surface.
- The length of a curve is the number of times it crosses the graph embedded.

Computing a homeomorphism



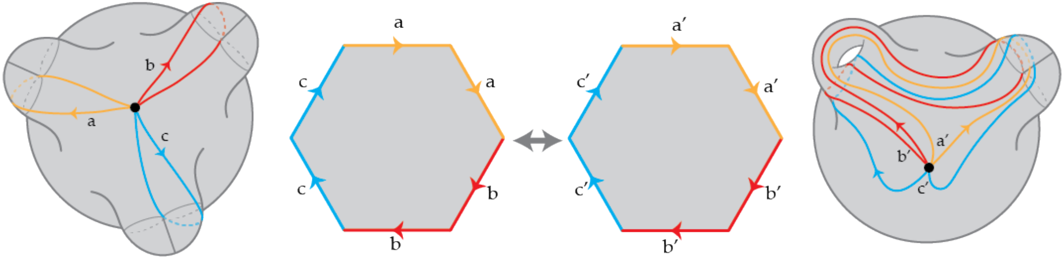
Computing a homeomorphism

- Visualizing the homeomorphism between these surfaces is not easy.



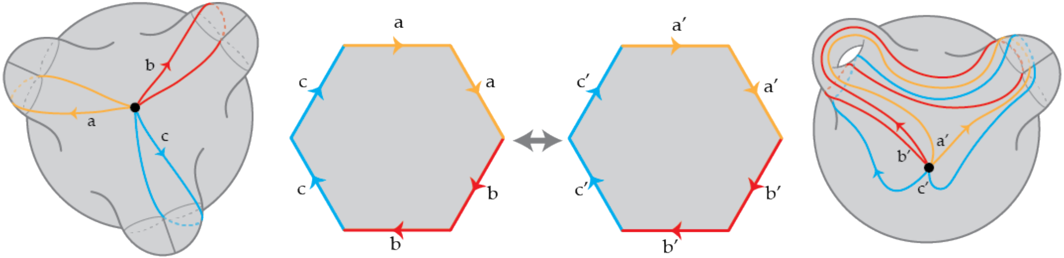
Computing a homeomorphism

- Visualizing the homeomorphism between these surfaces is not easy.
- A common approach is to cut both to a disk and put the disks in correspondence.



Computing a homeomorphism

- Visualizing the homeomorphism between these surfaces is not easy.
- A common approach is to cut both to a disk and put the disks in correspondence.



- Such a graph that cuts the surface into simpler pieces is called a **decomposition** of the surface.

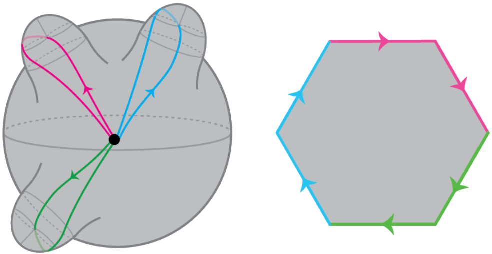
Question: How much can we control the **length** of a decomposition?

Overview

1. Two technical tools:
   - A model to represent non-orientable embeddings: Cross-cap drawing
   - An algorithm in genome rearrangement: Signed reversal distance
2. A short topological decomposition for non-orientable surfaces
3. Degenerate crossing number and Mohar's conjecture

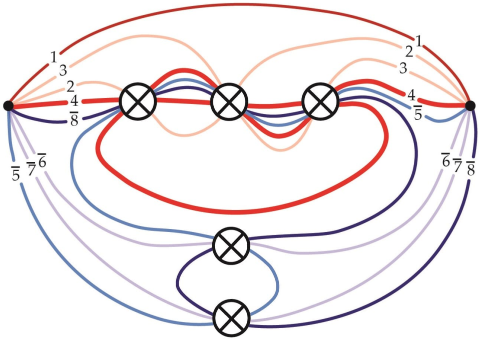
Overview

1. Two technical tools:
   - A model to represent non-orientable embeddings: Cross-cap drawing
   - An algorithm in genome rearrangement: Signed reversal distance
2. A short topological decomposition for non-orientable surfaces
3. Degenerate crossing number and Mohar's conjecture



Overview

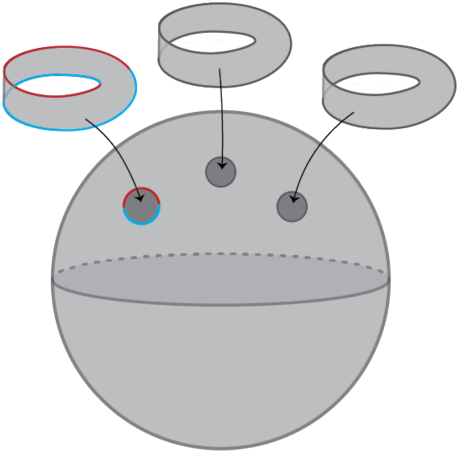
1. Two technical tools:
   - A model to represent non-orientable embeddings: Cross-cap drawing
   - An algorithm in genome rearrangement: Signed reversal distance
2. A short topological decomposition for non-orientable surfaces
3. Degenerate crossing number and Mohar's conjecture



Two technical tools

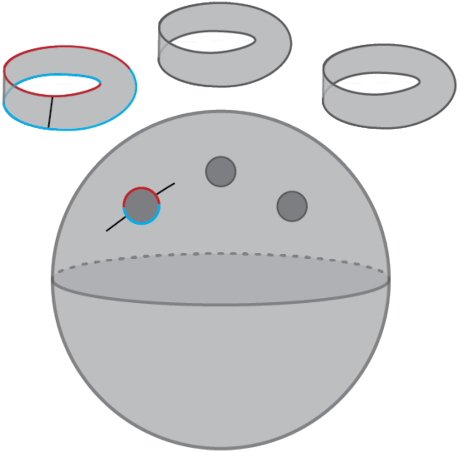
1) Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing.



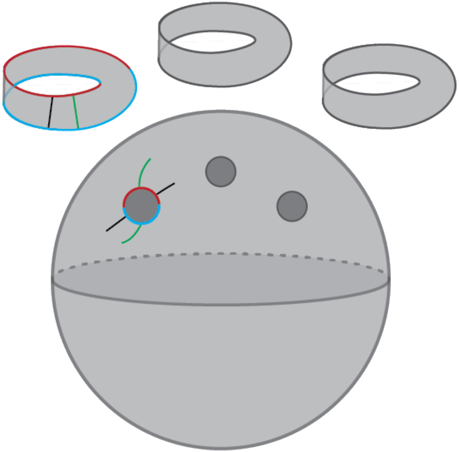
1) Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing.



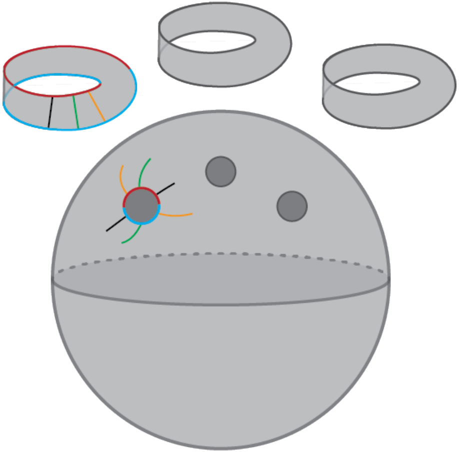
1) Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing.



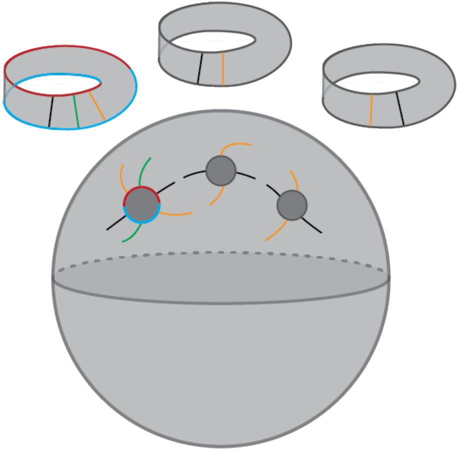
1) Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing.



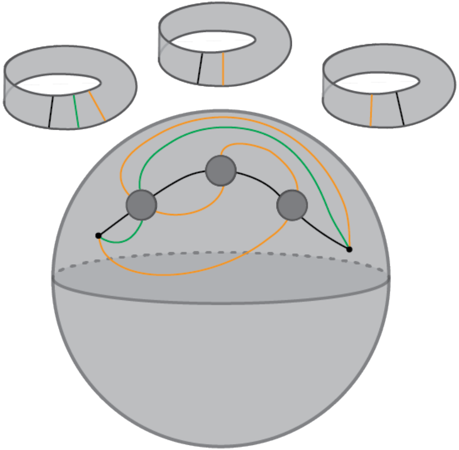
1) Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing.



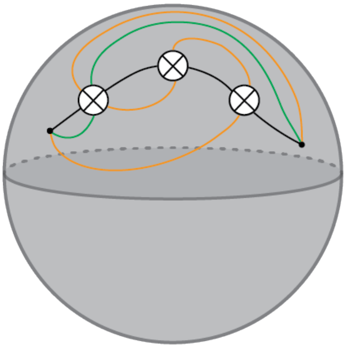
1) Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing.



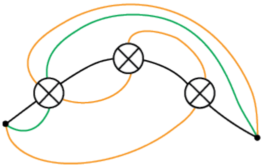
1) Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing.

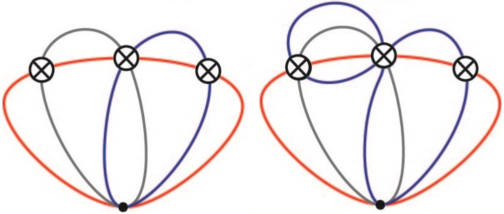


1) Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing.

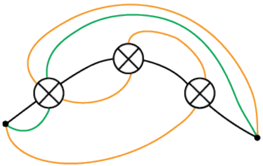


- A **cross-cap drawing** is a planar drawing with such transverse crossings at cross-caps.
- This **localization** of cross-caps is not "canonical"!



1) Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing.

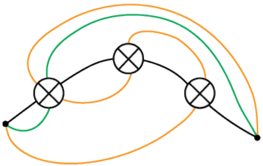


- A **cross-cap drawing** is a planar drawing with such transverse crossings at cross-caps.

Question: Can we control the number of times an edge enters a cross-cap?

1) Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing.



- A **cross-cap drawing** is a planar drawing with such transverse crossings at cross-caps.

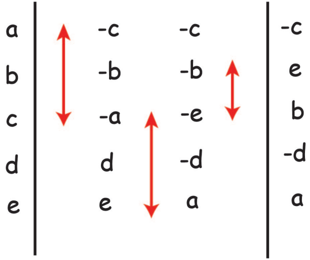
Question: Can we control the number of times an edge enters a cross-cap?

Theorem (Schaefer, Štefankovič '22)

A graph G embedded on a non-orientable surface admits a cross-cap drawing in which each edge enters each cross-cap ***at most twice****.*

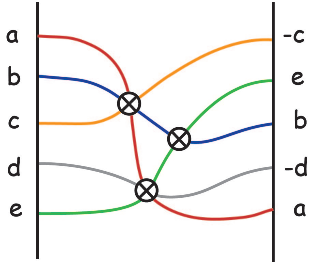
2) Genome rearrangement

- The **signed reversal distance** between two signed permutations is the minimum number of **reversals** to go from one to the other.
- It is computable in **polynomial time** [Hannenhali-Pevzner '99].
- This has strong similarities with crosscap drawings.



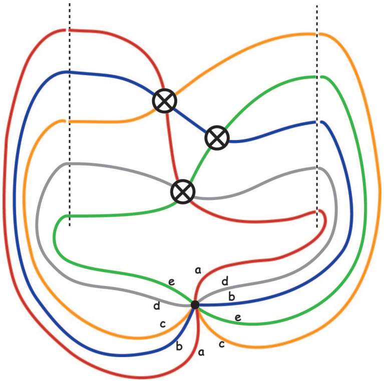
2) Genome rearrangement

- The **signed reversal distance** between two signed permutations is the minimum number of **reversals** to go from one to the other.
- It is computable in **polynomial time** [Hannenhali-Pevzner '99].
- This has strong similarities with crosscap drawings.



2) Genome rearrangement

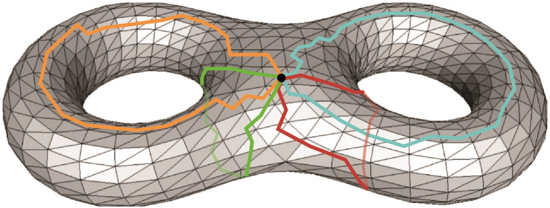
- The **signed reversal distance** between two signed permutations is the minimum number of **reversals** to go from one to the other.
- It is computable in **polynomial time** [Hannenhali-Pevzner ’99].
- This has strong similarities with crosscap drawings.

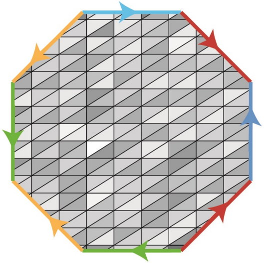


A short topological decomposition for non-orientable surfaces

Canonical decompositions

- **Orientable canonical decomposition**: a one-vertex graph with the fixed rotation system $a_1 b_1 a_1 b_1 a_2 b_2 a_2 b_2 \ldots$



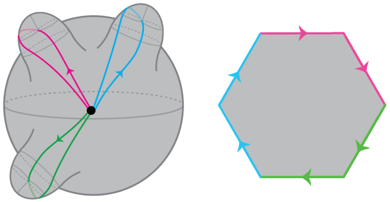


Theorem (Lazarus, Pocchiola, Vegter, Verroust '01)

*Given a graph cellularly embedded on an **orientable** surface of genus g, there exists an **orientable canonical decomposition**, so that each loop crosses each edge of the graph at most 4 times (total length $O(gn)$).*

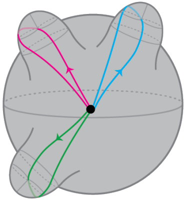
Canonical decompositions

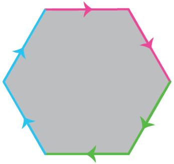
- Can I cut along the **non-orientable canonical decomposition**? the one vertex graph with rotation system $aabbcc\cdots$



Canonical decompositions

- Can I cut along the **non-orientable canonical decomposition**? the one vertex graph with rotation system *aabbcc* $\cdots$





Theorem (F., Hubard, de Mesmay '21)

*Given a graph cellularly embedded on a non-orientable surface, there exists a **non-orientable canonical decomposition** such that each loop in the system crosses each edge of the graph at most in 30 points (total length $O(gn)$).*

- Best previous bound is $O(g^2 n)$ (Lazarus '14).
- We use a new approach combining the Schaefer, Štefankovič algorithm and the Hannenhali-Pevzner algorithm.

Other cutting shapes

A more general question on finding short decompositions:

Negami's conjecture ('01)

Let G_1 and G_2 be two graphs cellularly embedded on a surface S of genus g. G_1 and G_2 can be embedded on S **simultaneously** such that each pair of their edges cross at most a constant number of times? (total of $O(n_1 n_2)$ crossings)

→ If true, any shape of decomposition can be computed with total length at most $O(n_1 n_2)$.

Other cutting shapes

A more general question on finding short decompositions:

Negami's conjecture ('01)

Let G_1 and G_2 be two graphs cellularly embedded on a surface S of genus g. G_1 and G_2 can be embedded on S **simultaneously** such that each pair of their edges cross at most a constant number of times? (total of $O(n_1 n_2)$ crossings)

$\rightarrow$ If true, any shape of decomposition can be computed with total length at most $O(n_1 n_2)$.

- The length of canonical decompositions matches the bound in Negami's conjecture.

Other cutting shapes

A more general question on finding short decompositions:

Negami's conjecture ('01)

Let G_1 and G_2 be two graphs cellularly embedded on a surface S of genus g. G_1 and G_2 can be embedded on S **simultaneously** such that each pair of their edges cross at most a constant number of times? (total of $O(n_1 n_2)$ crossings)

→ If true, any shape of decomposition can be computed with total length at most $O(n_1 n_2)$.

- The length of canonical decompositions matches the bound in Negami's conjecture.

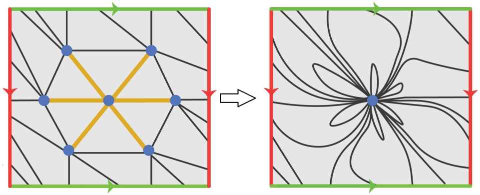
Best known bound:

Theorem (Negami '01)

Let G_1 and G_2 be two graphs cellularly embedded on a surface S of genus g. G_1 and G_2 can be embedded on S ***simultaneously*** *such that each pair of their edges cross at most $O(g)$ times (total length $O(gn_1 n_2)$).*

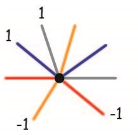
Reduction to the one-vertex case

- By contracting a **spanning tree**, our problem reduces to the case of one-vertex graphs.



- An embedding for a one-vertex graph, is entirely described by the cyclic ordering of the edges around the vertex, and, in the non-orientable case, the sidedness of the curves, an **embedding scheme**.

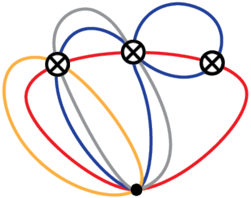
1 $\rightarrow$ Two-sided
-1 $\rightarrow$ One-sided



Short non-orientable canonical decomposition/ A different approach

Theorem (Schaefer-Štefankovič '15)

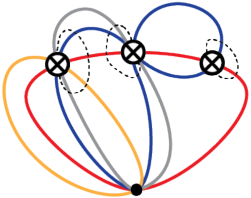
A graph G embeddable on a non-orientable surface admits a cross-cap drawing in which each edge enters each cross-cap ***at most twice****.*



Short non-orientable canonical decomposition/ A different approach

Theorem (Schaefer-Štefankovič '15)

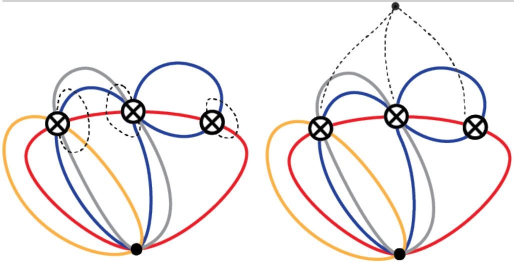
*A graph G embeddable on a non-orientable surface admits a cross-cap drawing in which each edge enters each cross-cap **at most twice**.*



Short non-orientable canonical decomposition/ A different approach

Theorem (Schaefer-Štefankovič '15)

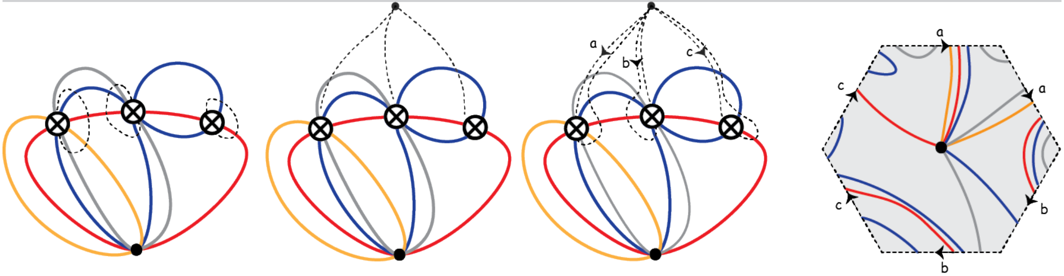
*A graph G embeddable on a non-orientable surface admits a cross-cap drawing in which each edge enters each cross-cap **at most twice**.*



Short non-orientable canonical decomposition/ A different approach

Theorem (Schaefer-Štefankovič '15)

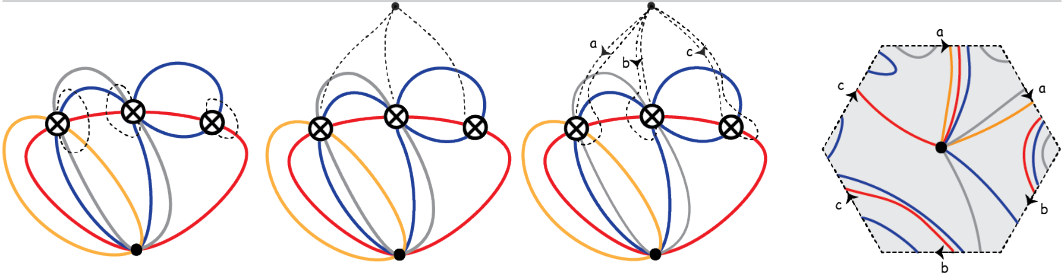
*A graph G embeddable on a non-orientable surface admits a cross-cap drawing in which each edge enters each cross-cap **at most twice**.*



Short non-orientable canonical decomposition/ A different approach

Theorem (Schaefer-Štefankovič '15)

A graph G embeddable on a non-orientable surface admits a cross-cap drawing in which each edge enters each cross-cap ***at most twice****.*

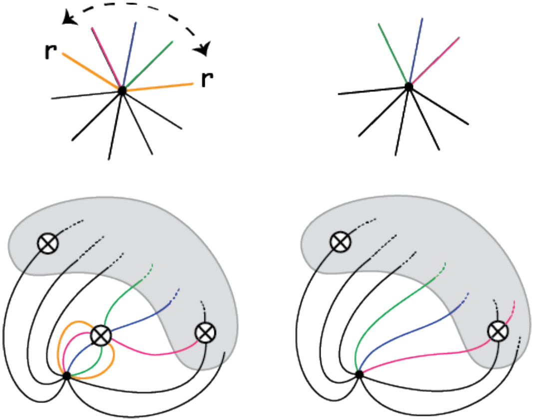


- If we can control the diameter of this cross-cap drawing, we can control the length of the canonical system of loops.

Sketch of the proof

- The proof is by induction on the number of edges.

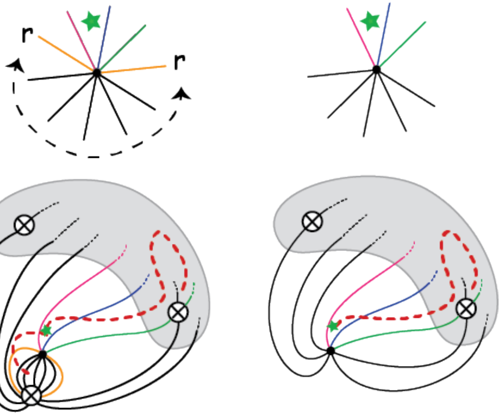
→ The case of <u>one-sided</u> <u>non-orienting</u> loops



Sketch of the proof

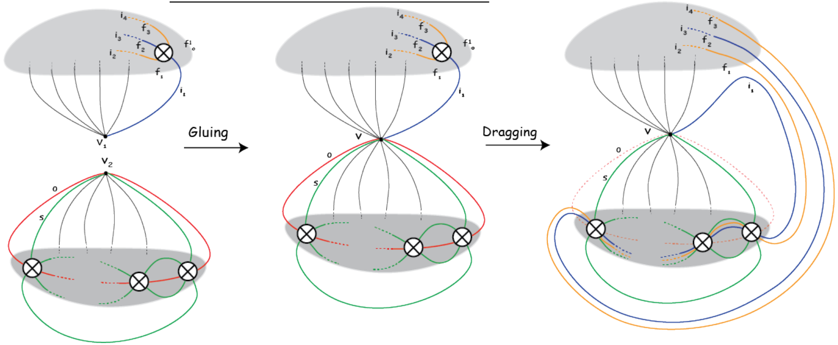
- The proof is by induction on the number of edges.
- We build a system of short paths on top of this algorithm.

→ The case of <u>one-sided non-orienting</u> loops



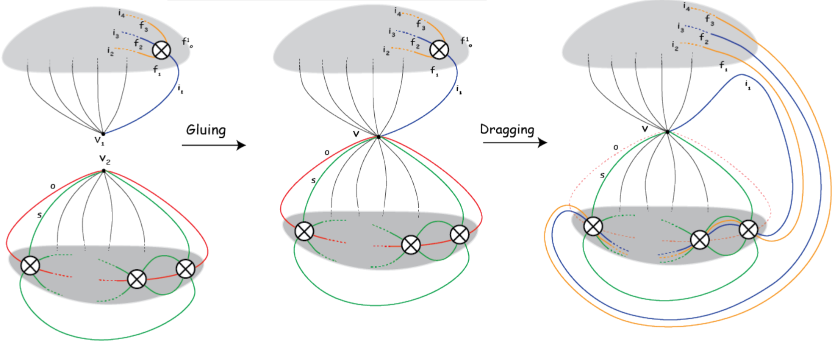
Sketch of the proof

- The proof is by induction on the number of edges.
- We build a system of short paths on top of this algorithm.
- We have to deal with non-contractible separating loops:



Sketch of the proof

- The proof is by induction on the number of edges.
- We build a system of short paths on top of this algorithm.
- We have to deal with non-contractible separating loops:



- To avoid cascading, we make sure to deal with all the separating loops at once, using ideas from the Hannenhali-Pevzner algorithm.

Other decompositions?

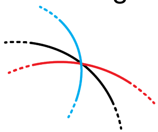
A similar approach lets us compute other short decompositions:

- An alternative computation of a short orientable canonical decomposition.
- Different short decompositions for non-orientable surfaces with rotation system: $a_1 a_1 \cdots a_k a_k b_1 c_1 \bar{b}_1 \bar{c}_1 \cdots b_m c_m \bar{b}_m \bar{c}_m$.

Degenerate crossing number

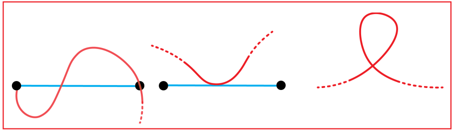
Crossing numbers for graphs

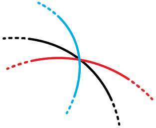
- Pach and Tóth: The **degenerate crossing number** of G, **dcr(G)**, is the minimum number of edge-crossings taken over all proper drawings of G in the plane in which multiple crossings at a point are counted as a single crossing.



Crossing numbers for graphs

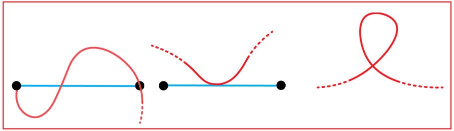
- Pach and Tóth: The **degenerate crossing number** of G, **dcr(G)**, is the minimum number of edge-crossings taken over all proper drawings of G in the plane in which multiple crossings at a point are counted as a single crossing.

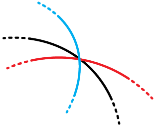




Crossing numbers for graphs

- Pach and Tóth: The **degenerate crossing number** of G, **dcr(G)**, is the minimum number of edge-crossings taken over all proper drawings of G in the plane in which multiple crossings at a point are counted as a single crossing.

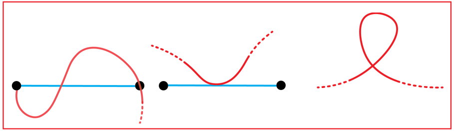


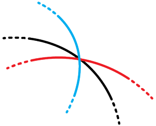


- Mohar: what if we allow self-crossings?
  $\rightarrow$ **genus crossing number gcr(G)**

Crossing numbers for graphs

- Pach and Tóth: The **degenerate crossing number** of G, **dcr(G)**, is the minimum number of edge-crossings taken over all proper drawings of G in the plane in which multiple crossings at a point are counted as a single crossing.



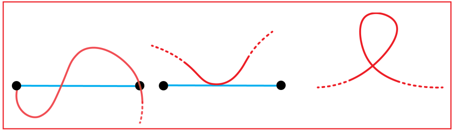


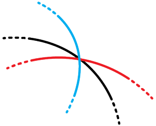
- Mohar: what if we allow self-crossings?
  $\rightarrow$ **genus crossing number gcr(G)**
- For any graph G:

$$\mathrm{gcr}(G) \leq \mathrm{dcr}(G)$$

Crossing numbers for graphs

- Pach and Tóth: The **degenerate crossing number** of G, **dcr(G)**, is the minimum number of edge-crossings taken over all proper drawings of G in the plane in which multiple crossings at a point are counted as a single crossing.





- Mohar: what if we allow self-crossings?
  $\rightarrow$ **genus crossing number gcr(G)**
- For any graph G:

$$\mathrm{gcr}(G) \leq \mathrm{dcr}(G)$$

Mohar's Conjecture 1 ('07)

For every graph G, gcr(G)=dcr(G).

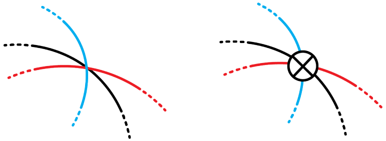
From crossing numbers to non-orientable genus

- The minimum cross-caps needed to draw a graph on a surface is called **non-orientable genus** $g(G)$ of the graph.

Theorem (Mohar '07)

For any graph G, $gcr(G) =$ *non-orientable genus of G.*

Cross-caps can be interpreted as multiple transverse crossings.



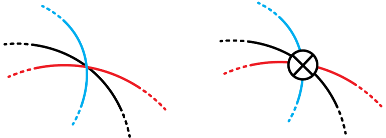
From crossing numbers to non-orientable genus

- The minimum cross-caps needed to draw a graph on a surface is called **non-orientable genus** $g(G)$ of the graph.

Theorem (Mohar '07)

For any graph G, $gcr(G) =$ non-orientable genus of G.

Cross-caps can be interpreted as multiple transverse crossings.



A cross-cap drawing is **perfect** if each edge enters each cross-cap **at most once**.

Mohar's Conjecture 1 ('07)

For every graph G, $dcr(G) = gcr(G) = g(G)$.
$\Downarrow$
Every graph G admits a **perfect** cross-cap drawing with $g(G)$ cross-caps.

The Conjectures

Mohar's Conjecture 1 ('07)

Every graph G admits a **perfect** cross-cap drawing with $g(G)$ cross-caps.

Mohar's (stronger) Conjecture 2 ('07)

Every loopless graph embedded on a non-orientable surface admits a **perfect** cross-cap drawing.

The Conjectures

Mohar's Conjecture 1 ('07)

Every graph G admits a **perfect** cross-cap drawing with $g(G)$ cross-caps.

Mohar's (stronger) Conjecture 2 ('07)

Every loopless graph embedded on a non-orientable surface admits a **perfect** cross-cap drawing.

Mohar's (even stronger) Conjecture 3 ('07)

Every graph embedded on a non-orientable surface in which loops are non-separating admits a **perfect** cross-cap drawing.

The Conjectures

Mohar's Conjecture 1 ('07)

Every graph G admits a **perfect** cross-cap drawing with $g(G)$ cross-caps.

Mohar's (stronger) Conjecture 2 ('07)

Every loopless graph embedded on a non-orientable surface admits a **perfect** cross-cap drawing.

Mohar's (even stronger) Conjecture 3 ('07)

Every graph embedded on a non-orientable surface in which loops are non-separating admits a **perfect** cross-cap drawing.

Conjecture 1

$\Uparrow$

Conjecture 2

$\Uparrow$

Conjecture 3

The Conjectures

Mohar's Conjecture 1 ('07)

Every graph G admits a **perfect** cross-cap drawing with $g(G)$ cross-caps.

Mohar's (stronger) Conjecture 2 ('07)

Every loopless graph embedded on a non-orientable surface admits a **perfect** cross-cap drawing.

Mohar's (even stronger) Conjecture 3 ('07)

Every graph embedded on a non-orientable surface in which loops are non-separating admits a **perfect** cross-cap drawing.

$\rightarrow$ Schaefer and Štefankovič disproved this.

The Conjectures

Mohar's Conjecture 1 ('07)

Every graph G admits a **perfect** cross-cap drawing with $g(G)$ cross-caps.

Mohar's (stronger) Conjecture 2 ('07)

Every loopless graph embedded on a non-orientable surface admits a **perfect** cross-cap drawing.

→ We provide a 2-vertex counter example.

Mohar's (even stronger) Conjecture 3 ('07)

Every graph embedded on a non-orientable surface in which loops are non-separating admits a **perfect** cross-cap drawing.

→ Schaefer and Štefankovič disproved this.

Theorem (F., Hubard, de Mesmay '23)

Apart from two exceptional families of graphs, all 2-vertex loopless graphs embedded on non-orientable surfaces satisfy Conjecture 2.

The counter example

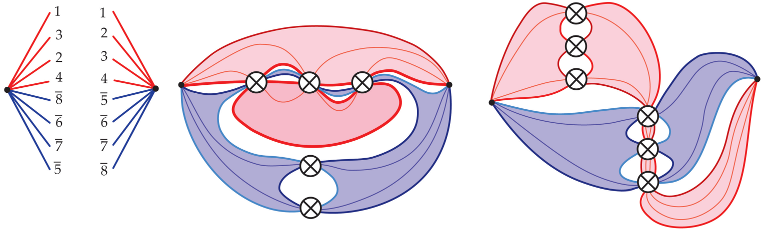
Mohar's (stronger) Conjecture 2 ('07)

Every loopless graph embedded on a non-orientable surface admits a **perfect** cross-cap drawing.

Conjecture 2 does not hold:

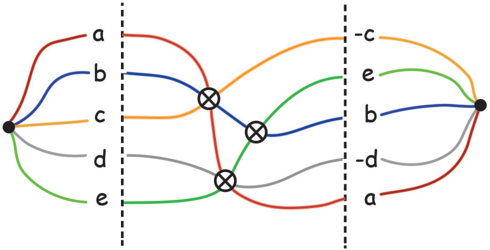
Theorem (F., Hubard, de Mesmay '23)

There exists a 2-vertex loopless graph embedded on a non-orientable surface that does not admit a ***perfect*** *cross-cap drawing.*



Signed reversals distance vs. Degenerate crossing

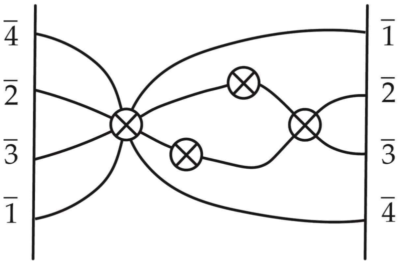
- Our main technical tool is the Hannenhali-Pevzner algorithm.
- The algorithm imposes an order on the cross-caps → each edge enters each cross-cap at most once.



Signed reversals distance vs. Degenerate crossing

- Our main technical tool is the Hannenhali-Pevzner algorithm.
- The Hannenhali-Pevzner algorithm focuses on handling the cases where the minimum number of signed reversals/crosscaps is different from the non-orientable genus.

→ There are sub-words that cost them extra cross-caps called **blocks**.

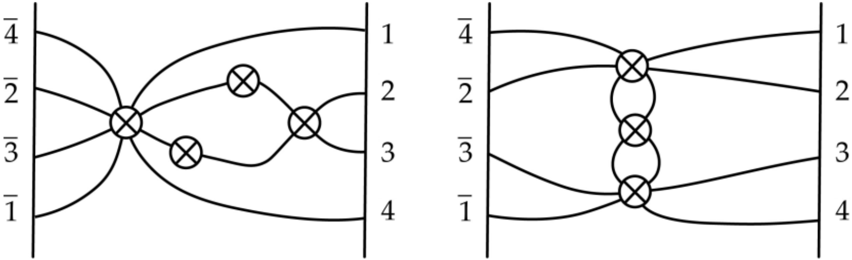


Signed reversals distance vs. Degenerate crossing

- Our main technical tool is the Hannenhali-Pevzner algorithm.
- The Hannenhali-Pevzner algorithm focuses on handling the cases where the minimum number of signed reversals/crosscaps is different from the non-orientable genus.

→ There are sub-words that cost them extra cross-caps called **blocks**.

- We prove that almost all of these cases can be handled in a topological setting.



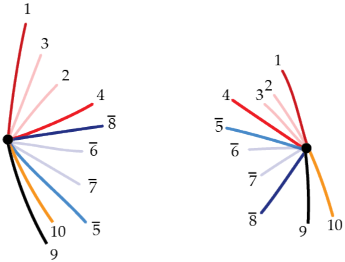
Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay '23)

*Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a **perfect** cross-cap drawing.*

Sketch of the proof:

→ **reduce** the scheme.



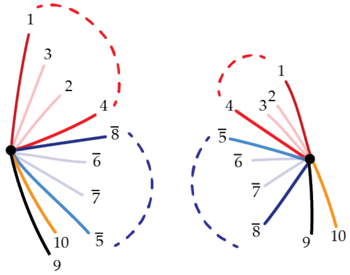
Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay '23)

*Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a **perfect** cross-cap drawing.*

Sketch of the proof:

→ **reduce** the scheme.



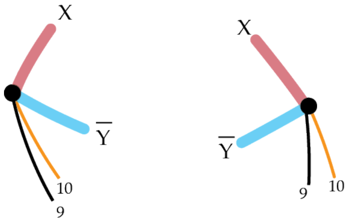
Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay '23)

Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a ***perfect*** *cross-cap drawing.*

Sketch of the proof:

→ **reduce** the scheme.



Cross-cap drawings of 2-vertex schemes

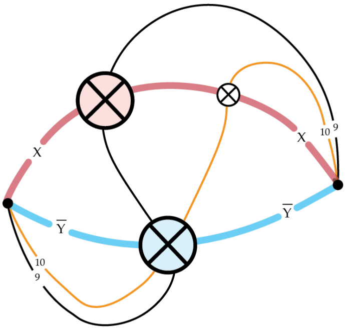
Theorem (F., Hubard, de Mesmay '23)

Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a ***perfect*** *cross-cap drawing.*

Sketch of the proof:

→ **reduce** the scheme.

→ apply Hannenhali-Pevzner algorithm.



Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay '23)

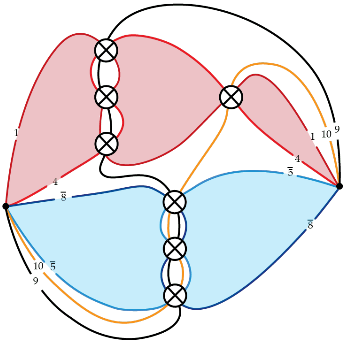
Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a ***perfect*** *cross-cap drawing.*

Sketch of the proof:

→ **reduce** the scheme.

→ apply Hannenhali-Pevzner algorithm.

→ **blow up** the cross-caps.



Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay ’23)

Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a ***perfect*** *cross-cap drawing.*

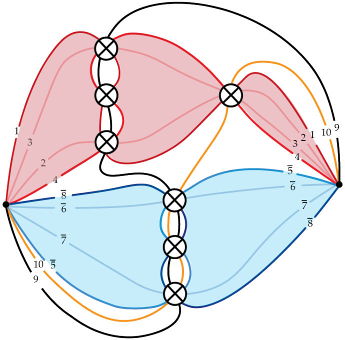
Sketch of the proof:

→ **reduce** the scheme.

→ apply Hannenhali-Pevzner algorithm.

→ **blow up** the cross-caps.

→ complete the drawing.



Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay '23)

*Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a **perfect** cross-cap drawing.*

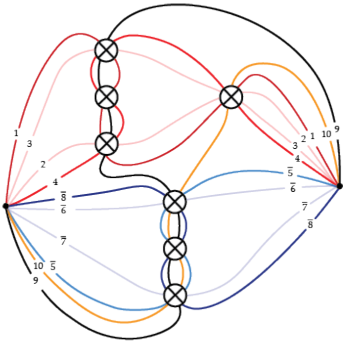
Sketch of the proof:

→ **reduce** the scheme.

→ apply Hannenhali-Pevzner algorithm.

→ **blow up** the cross-caps.

→ complete the drawing.

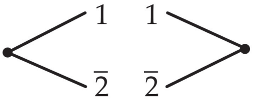


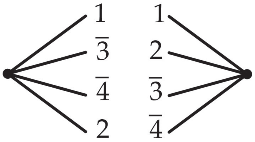
Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay '23)

*Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a **perfect** cross-cap drawing.*

- In particular under standard models of random maps, almost all 2-vertex loopless embedded graphs satisfy Conjecture 2.



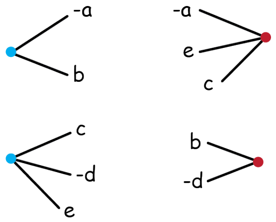


Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay '23)

*Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a **perfect** cross-cap drawing.*

- This algorithm can be extended for loopless bipartite embedding schemes.

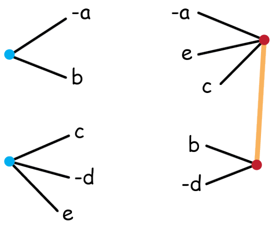


Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay '23)

Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a ***perfect*** *cross-cap drawing.*

- This algorithm can be extended for loopless bipartite embedding schemes.

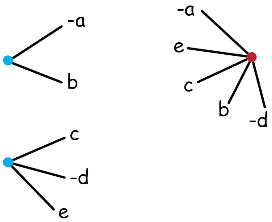


Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay '23)

Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a ***perfect*** *cross-cap drawing.*

- This algorithm can be extended for loopless bipartite embedding schemes.

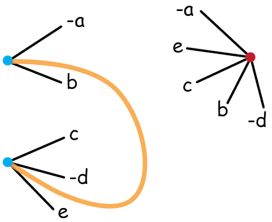


Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay '23)

Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a ***perfect*** *cross-cap drawing.*

- This algorithm can be extended for loopless bipartite embedding schemes.

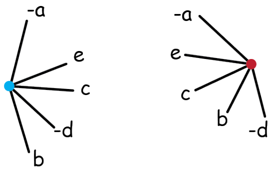


Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay '23)

Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a ***perfect*** *cross-cap drawing.*

- This algorithm can be extended for loopless bipartite embedding schemes.



Conclusion

Negami's conjecture ('01)

Let G_1 and G_2 be two graphs cellularly embedded on a surface S of genus g. G_1 and G_2 can be embedded on S **simultaneously** such that each pair of their edges cross at most a constant number of times? (total of $O(n_1 n_2)$ crossings)

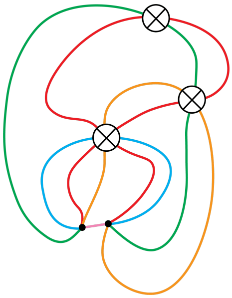
→ Any shape of decomposition can be computed shortly.

Conclusion

Negami's conjecture ('01)

Let G_1 and G_2 be two graphs cellularly embedded on a surface S of genus g. G_1 and G_2 can be embedded on S **simultaneously** such that each pair of their edges cross at most a constant number of times? (total of $O(n_1 n_2)$ crossings)

→ Any shape of decomposition can be computed shortly.



Mohar's Conjecture 1 ('07)

For every graph G, $\text{gcr}(G) = \text{dcr}(G)$.

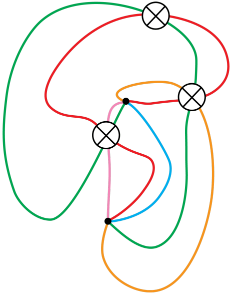
→ Allowing the graph to have more vertices, increases the possibility of having a **perfect** cross-cap drawing.

Conclusion

Negami's conjecture ('01)

Let G_1 and G_2 be two graphs cellularly embedded on a surface S of genus g. G_1 and G_2 can be embedded on S **simultaneously** such that each pair of their edges cross at most a constant number of times? (total of $O(n_1 n_2)$ crossings)

→ Any shape of decomposition can be computed shortly.



Mohar's Conjecture 1 ('07)

For every graph G, $\mathrm{gcr}(G) = \mathrm{dcr}(G)$.

→ Allowing the graph to have more vertices, increases the possibility of having a **perfect** cross-cap drawing.

Conclusion

Negami's conjecture ('01)

Let G_1 and G_2 be two graphs cellularly embedded on a surface S of genus g. G_1 and G_2 can be embedded on S **simultaneously** such that each pair of their edges cross at most a constant number of times? (total of $O(n_1 n_2)$ crossings)

→ Any shape of decomposition can be computed shortly.

Mohar's Conjecture 1 ('07)

For every graph G, $\mathrm{gcr}(G) = \mathrm{dcr}(G)$.

→ Allowing the graph to have more vertices, increases the possibility of having a **perfect** cross-cap drawing.

Thank You!