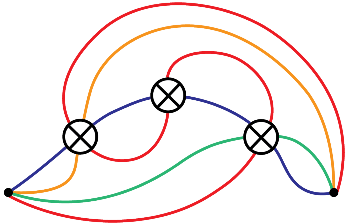
Degenerate crossing number and signed reversal distance

Niloufar FULADI Alfredo HUBARD
Arnaud de MESMAY

Université Gustave Eiffel, Paris

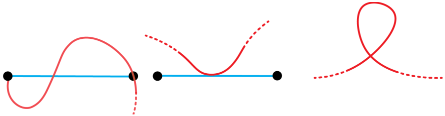


International Symposium on Graph Drawing
Palermo, September 2023

Outline

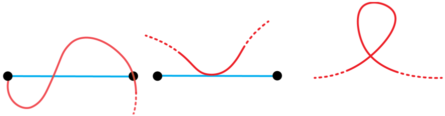
Crossing numbers for graphs

- The **crossing number** of G, **cr(G)**, is the minimum number of edge-crossings taken over all proper drawings of G in the plane.



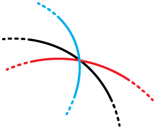
Crossing numbers for graphs

- The **crossing number** of G, **cr**(G), is the minimum number of edge-crossings taken over all proper drawings of G in the plane.



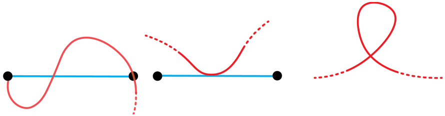
- Pach and Tóth: what if a multiple crossings at a point is counted as a single crossing?

  $\rightarrow$ **degenerate crossing number dcr**(G)**.**



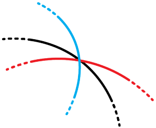
Crossing numbers for graphs

- The **crossing number** of G, **cr**(G), is the minimum number of edge-crossings taken over all proper drawings of G in the plane.



- Pach and Tóth: what if a multiple crossings at a point is counted as a single crossing?

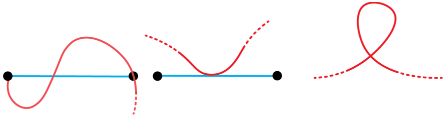
  $\rightarrow$ **degenerate crossing number dcr**(G)**.**



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

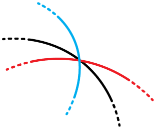
Crossing numbers for graphs

- The **crossing number** of G, **cr**(G), is the minimum number of edge-crossings taken over all proper drawings of G in the plane.



- Pach and Tóth: what if a multiple crossings at a point is counted as a single crossing?

  $\rightarrow$ **degenerate crossing number dcr**(G)**.**

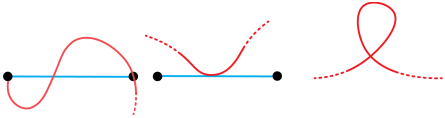


- Mohar: what if we allow self-crossings?
  $\rightarrow$ **genus crossing number gcr**(G).

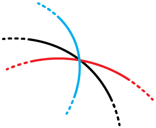
- For any graph G: $$\mathrm{gcr}(G) \leq \mathrm{dcr}(G) \leq \mathrm{cr}(G)$$

Crossing numbers for graphs

- The **crossing number** of G, **cr(G)**, is the minimum number of edge-crossings taken over all proper drawings of G in the plane.



- Pach and Tóth: what if a multiple crossings at a point is counted as a single crossing?
  $\rightarrow$ **degenerate crossing number dcr(G).**



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

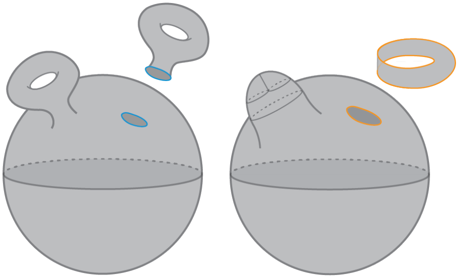
$$\mathrm{gcr}(G) \leq \mathrm{dcr}(G) \leq \mathrm{cr}(G)$$

Mohar's Conjecture 1 ('07)

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

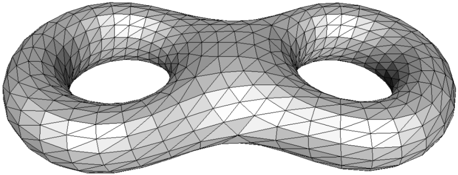
Graphs embedded on surfaces

- A surface is a topological space that locally looks like the plane.
- In this talk, we deal with connected compact surfaces.
- They are classified by their orientability and their genus.



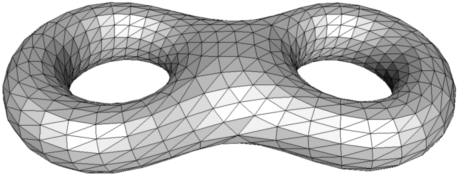
Graphs embedded on surfaces

- A surface is a topological space that locally looks like the plane.
- In this talk, we deal with connected compact surfaces.
- They are classified by their orientability and their genus.
- An embedding of G on a surface S is an injective map $G \hookrightarrow S$.



Graphs embedded on surfaces

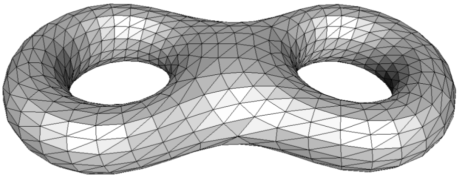
- A surface is a topological space that locally looks like the plane.
- In this talk, we deal with connected compact surfaces.
- They are classified by their orientability and their genus.
- An embedding of G on a surface S is an injective map $G \hookrightarrow S$.



- The **non-orientable genus** $g(G)$ of a graph G is the minimum number of cross-caps that it needs to be embedded on a surface.

Graphs embedded on surfaces

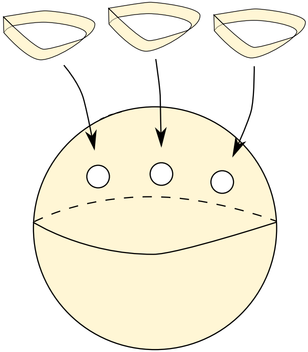
- A surface is a topological space that locally looks like the plane.
- In this talk, we deal with connected compact surfaces.
- They are classified by their orientability and their genus.
- An embedding of G on a surface S is an injective map $G \hookrightarrow S$.



- The **non-orientable genus** $g(G)$ of a graph G is the minimum number of cross-caps that it needs to be embedded on a surface.
- Graph embeddings are hard to visualize on a surface.

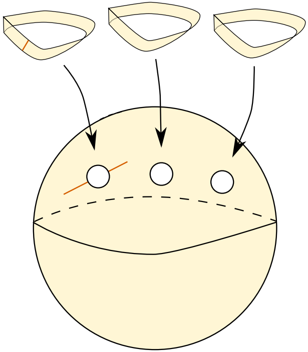
Cross-cap drawings and non-orientable embeddings

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



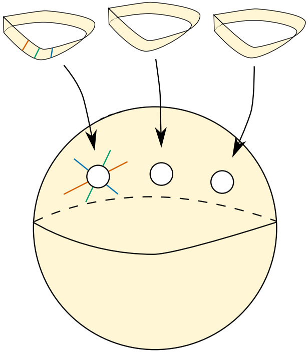
Cross-cap drawings and non-orientable embeddings

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



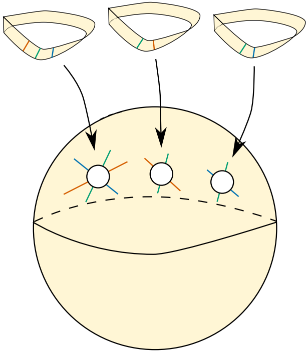
Cross-cap drawings and non-orientable embeddings

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



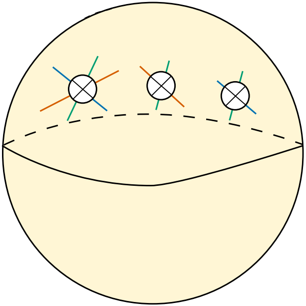
Cross-cap drawings and non-orientable embeddings

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



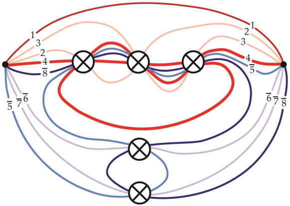
Cross-cap drawings and non-orientable embeddings

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



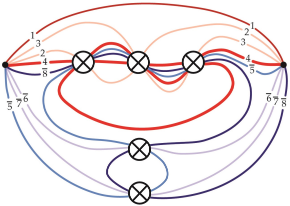
Cross-cap drawings and non-orientable embeddings

- 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.

Cross-cap drawings and non-orientable embeddings



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

Question

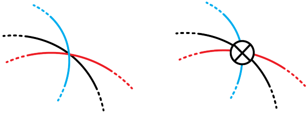
Can we control the number of times an edge enters a cross-cap?

From crossing numbers to non-orientable genus

These cross-caps can be interpreted as multiple transverse crossings.

Theorem (Mohar '07)

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

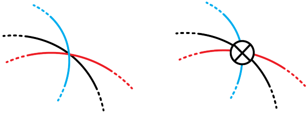


From crossing numbers to non-orientable genus

These cross-caps can be interpreted as multiple transverse crossings.

Theorem (Mohar '07)

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



A **perfect cross-cap drawing** for a graph is one in which 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.

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.

→ Schaefer and Štefankovič disprove 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.

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 disprove this.

Theorem (Schaefer-Štefankovič)

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

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.

- → We provide a 2-vertex counter example.
- → Schaefer and Štefankovič disprove this.

Theorem (Schaefer-Štefankovič)

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

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.

- → We provide a 2-vertex counter example.
- → Schaefer and Štefankovič disprove this.

Theorem (Schaefer-Štefankovič)

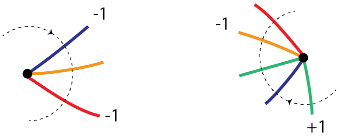
A graph G embeddable on N_g admits a cross-cap drawing in which each edge enters each cross-cap at most ***twice****.*

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

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

Embedding schemes

- An embedding for a graph, is entirely described by **an embedding scheme**:
    - the cyclic ordering of the edges around the vertex
    - (in the non-orientable case) a signature $+1$ or -1 associated to each edge

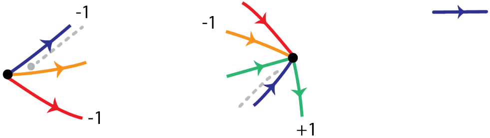


Embedding schemes

- An embedding for a graph, is entirely described by **an embedding scheme**:
  - the cyclic ordering of the edges around the vertex
  - (in the non-orientable case) a signature $+1$ or -1 associated to each edge
- Given an embedding scheme, we can compute the faces of the embedding:

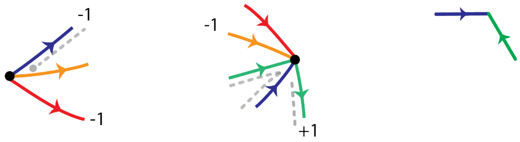
Embedding schemes

- An embedding for a graph, is entirely described by **an embedding scheme**:
  - the cyclic ordering of the edges around the vertex
  - (in the non-orientable case) a signature $+1$ or -1 associated to each edge
- Given an embedding scheme, we can compute the faces of the embedding:



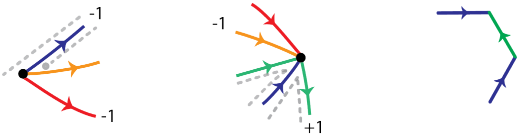
Embedding schemes

- An embedding for a graph, is entirely described by **an embedding scheme**:
  - the cyclic ordering of the edges around the vertex
  - (in the non-orientable case) a signature $+1$ or -1 associated to each edge
- Given an embedding scheme, we can compute the faces of the embedding:



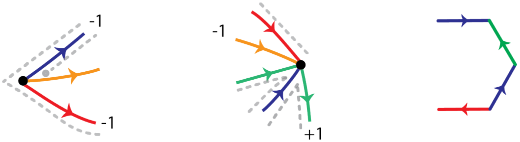
Embedding schemes

- An embedding for a graph, is entirely described by **an embedding scheme**:
  - the cyclic ordering of the edges around the vertex
  - (in the non-orientable case) a signature $+1$ or -1 associated to each edge
- Given an embedding scheme, we can compute the faces of the embedding:



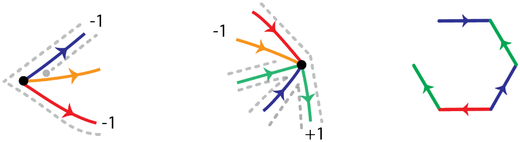
Embedding schemes

- An embedding for a graph, is entirely described by **an embedding scheme**:
  - the cyclic ordering of the edges around the vertex
  - (in the non-orientable case) a signature $+1$ or -1 associated to each edge
- Given an embedding scheme, we can compute the faces of the embedding:



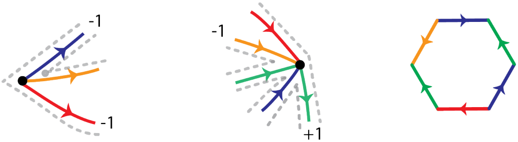
Embedding schemes

- An embedding for a graph, is entirely described by **an embedding scheme**:
    - the cyclic ordering of the edges around the vertex
    - (in the non-orientable case) a signature $+1$ or -1 associated to each edge
- Given an embedding scheme, we can compute the faces of the embedding:



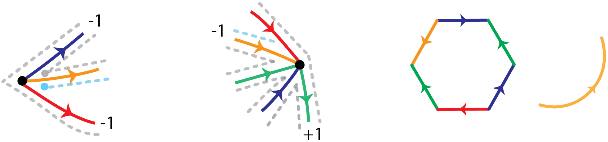
Embedding schemes

- An embedding for a graph, is entirely described by **an embedding scheme**:
  - the cyclic ordering of the edges around the vertex
  - (in the non-orientable case) a signature $+1$ or -1 associated to each edge
- Given an embedding scheme, we can compute the faces of the embedding:



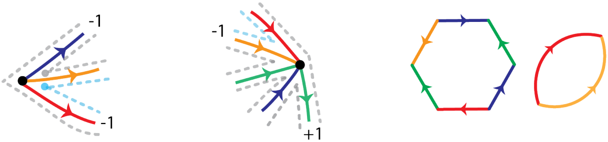
Embedding schemes

- An embedding for a graph, is entirely described by **an embedding scheme**:
  - the cyclic ordering of the edges around the vertex
  - (in the non-orientable case) a signature $+1$ or -1 associated to each edge
- Given an embedding scheme, we can compute the faces of the embedding:



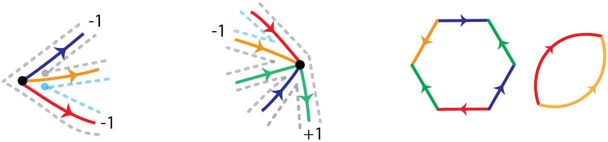
Embedding schemes

- An embedding for a graph, is entirely described by **an embedding scheme**:
  - the cyclic ordering of the edges around the vertex
  - (in the non-orientable case) a signature $+1$ or -1 associated to each edge
- Given an embedding scheme, we can compute the faces of the embedding:



Embedding schemes

- An embedding for a graph, is entirely described by **an embedding scheme**:
  - the cyclic ordering of the edges around the vertex
  - (in the non-orientable case) a signature $+1$ or -1 associated to each edge
- Given an embedding scheme, we can compute the faces of the embedding:

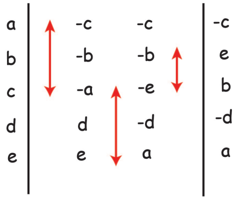


- A cross-cap drawing of an embedding scheme respects the signatures: each edge with signature $+1$ (resp. -1) enters even (resp. odd) number of cross-caps.

An unexpected connection

Our main technical tool for our results comes from computational biology.

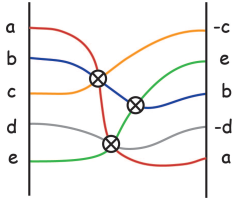
- The signed reversal distance between two signed words is the minimum number of reversals to go from one to the other one.
- Very important in computational genomics, computable in polynomial time [Hannenhalli-Pevzner '99].
- Strong similarities with crosscap drawings, which we leverage in all of our results.



An unexpected connection

Our main technical tool for our results comes from computational biology.

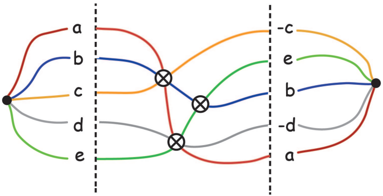
- The signed reversal distance between two signed words is the minimum number of reversals to go from one to the other one.
- Very important in computational genomics, computable in polynomial time [Hannenhalli-Pevzner '99].
- Strong similarities with crosscap drawings, which we leverage in all of our results.



An unexpected connection

Our main technical tool for our results comes from computational biology.

- The signed reversal distance between two signed words is the minimum number of reversals to go from one to the other one.
- Very important in computational genomics, computable in polynomial time [Hannenhalli-Pevzner '99].
- Strong similarities with crosscap drawings, which we leverage in all of our results.



From signed reversals to cross-cap drawings

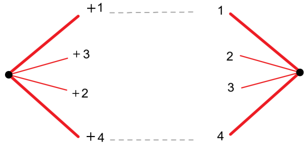
- In our words, the Hannenhalli-Pevzner algorithm focuses on counting and handling the cases where the minimum number of signed reversals/crosscaps is different from the non-orientable genus.

From signed reversals to cross-cap drawings

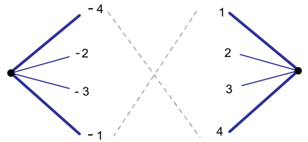
- In our words, the Hannenhalli-Pevzner algorithm focuses on counting and handling the cases where the minimum number of signed reversals/crosscaps is different from the non-orientable genus.

→ dealing with these sub-words costs them extra cross-caps:

- **Positive block**:
  - The **frames** 1 and 4 appear with 14 and 41 order around vertices.
  - all $+1$ signatures.

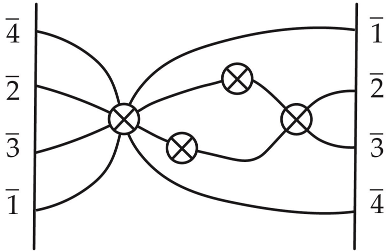


- **Negative block**:
  - The **frames** 1 and 4 appear with 14 around both vertices.
  - all -1 signatures.



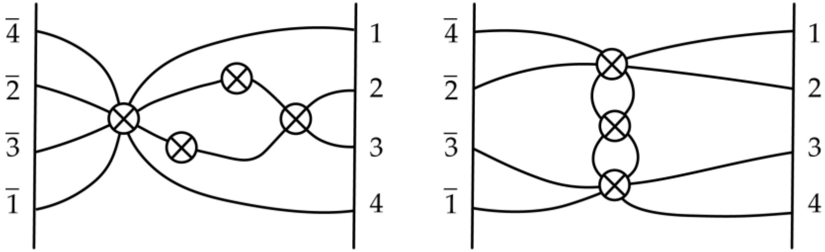
From signed reversals to cross-cap drawings

- In our words, the Hannenhalli-Pevzner algorithm focuses on counting and handling the cases where the minimum number of signed reversals/crosscaps is different from the non-orientable genus.
- We prove that almost all of these cases can be handled in a topological setting.



From signed reversals to cross-cap drawings

- In our words, the Hannenhalli-Pevzner algorithm focuses on counting and handling the cases where the minimum number of signed reversals/crosscaps is different from the non-orientable genus.
- We prove that almost all of these cases can be handled in a topological setting.



The counter example

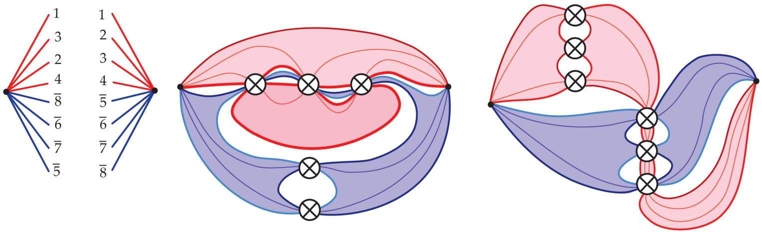
Mohar's (stronger) Conjecture 2 ('07)

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.*



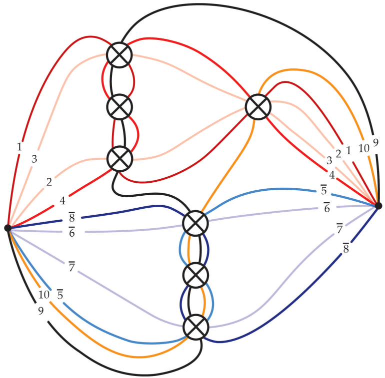
Cross-cap drawings of 2-vertex schemes

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

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

In particular:

- Under standard models of random maps, almost all 2-vertex loopless embedded graphs satisfy Conjecture 2.
- The behavior under adding edges is counter-intuitive.



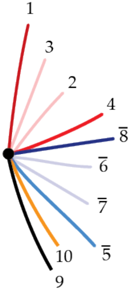
Cross-cap drawings of 2-vertex schemes

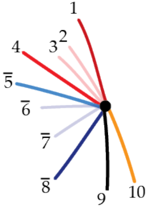
Theorem (F., Hubard, de Mesmay '23)

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

Sketch of the proof:

→ **reduce** the scheme.





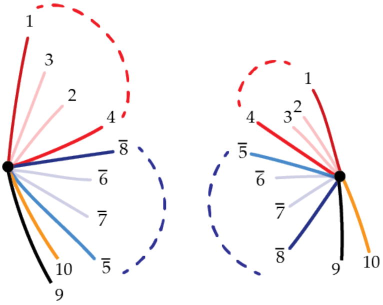
Cross-cap drawings of 2-vertex schemes

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

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

Sketch of the proof:

→ **reduce** the scheme.



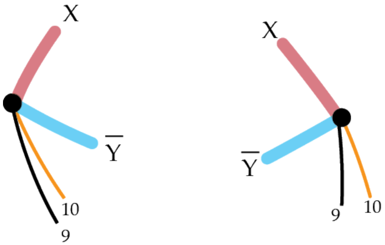
Cross-cap drawings of 2-vertex schemes

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

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

Sketch of the proof:

→ **reduce** the scheme.



Cross-cap drawings of 2-vertex schemes

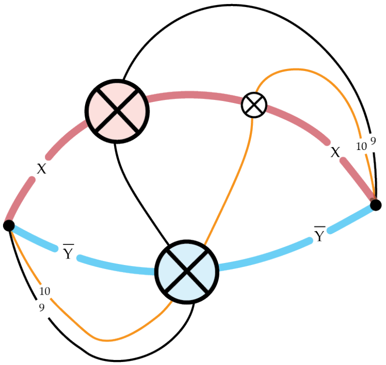
Theorem (F., Hubard, de Mesmay '23)

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

Sketch of the proof:

→ **reduce** the scheme.

→ apply Hannenhalli-Pevzner algorithm.



Cross-cap drawings of 2-vertex schemes

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

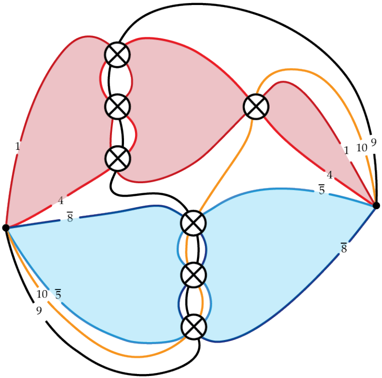
Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces satisfy Conjecture 2.

Sketch of the proof:

→ **reduce** the scheme.

→ apply Hannenhalli-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 satisfy Conjecture 2.

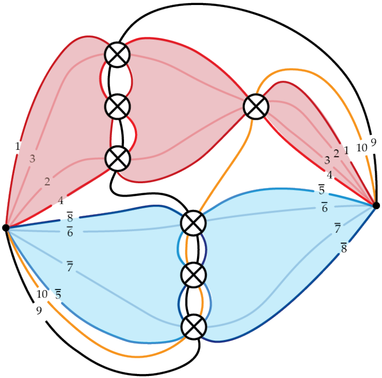
Sketch of the proof:

→ **reduce** the scheme.

→ apply Hannenhalli-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 satisfy Conjecture 2.

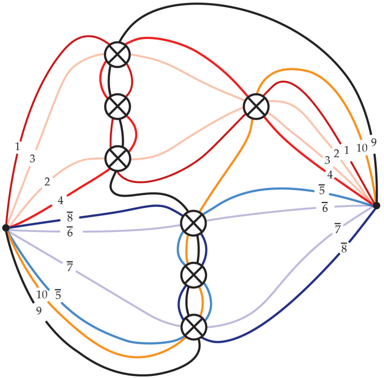
Sketch of the proof:

→ **reduce** the scheme.

→ apply Hannenhalli-Pevzner algorithm.

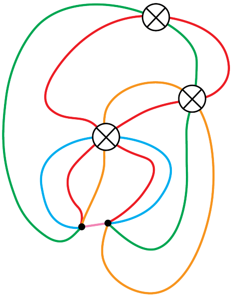
→ **blow up** the cross-caps.

→ complete the drawing.



Conclusion

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



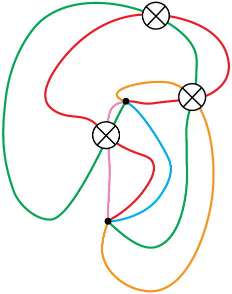
→ Although Mohar's conjectures 2 and 3 are wrong, there is a great chance that conjecture 1 is correct.

Mohar's Conjecture 1 ('07)

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

Conclusion

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



→ Although Mohar's conjectures 2 and 3 are wrong, there is a great chance that conjecture 1 is correct.

Mohar's Conjecture 1 ('07)

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

Thank You!