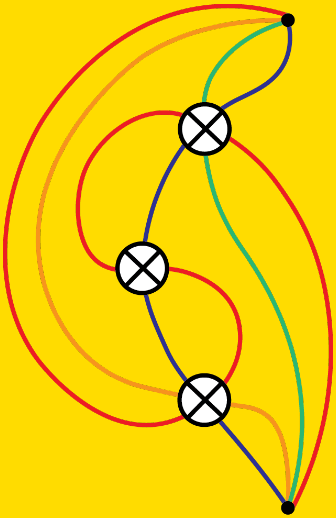
Cross-cap drawings and signed reversal distance

Niloufar Fuladi

Joint work with:
Arnaud de Mesmay
Alfredo Hubard

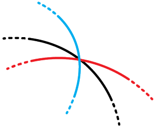


SMaRT workshop, Luxembourg
18 March 2024

Table of contents

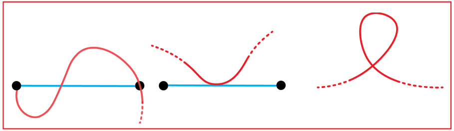
Crossing numbers for graphs

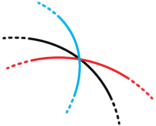
- Pach and Tóth: The **degenerate crossing number** of G, **dcr(G)**, is the minimum number of edge-crossings taken over all proper drawings of G in the plane in which multiple crossings at a point are counted as a single crossing.



Crossing numbers for graphs

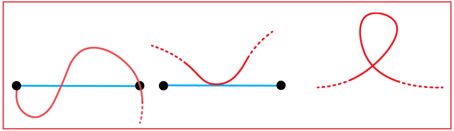
- Pach and Tóth: The **degenerate crossing number** of G, **dcr(G)**, is the minimum number of edge-crossings taken over all proper drawings of G in the plane in which multiple crossings at a point are counted as a single crossing.

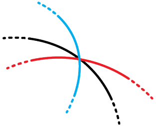




Crossing numbers for graphs

- 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.

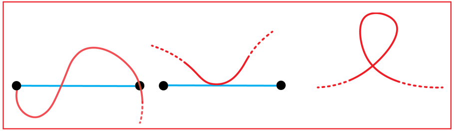


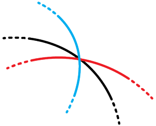


- Mohar: what if we allow self-crossings?
  $\rightarrow$ **genus crossing number gcr(G)**

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.



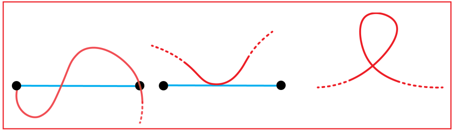


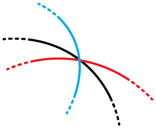
- Mohar: what if we allow self-crossings?
  $\rightarrow$ **genus crossing number gcr(G)**
- For any graph G:

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

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:

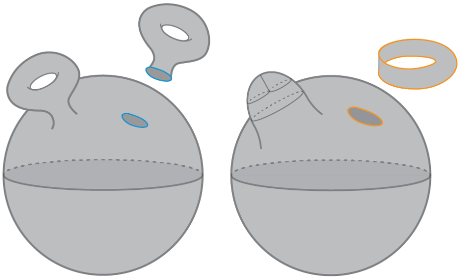
$$\mathrm{gcr}(G) \leq \mathrm{dcr}(G)$$

Mohar's Conjecture 1 ('07)

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

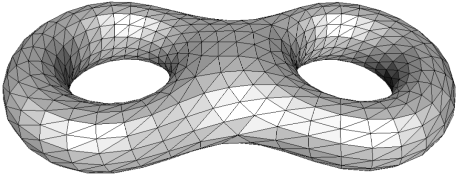
Graphs embedded on surfaces

- In this talk, we deal with connected compact surfaces.
- They are classified by their **orientability** and their genus.



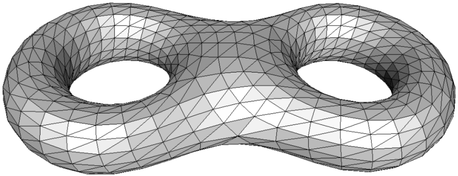
Graphs embedded on surfaces

- 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

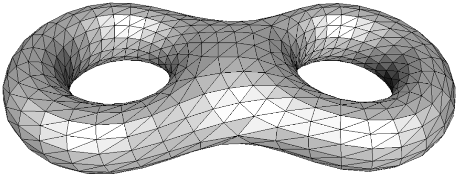
- In this talk, we deal with connected compact surfaces.
- They are classified by their **orientability** and their genus.
- An **embedding** of G on a surface S is an injective map $G \hookrightarrow S$.



- 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

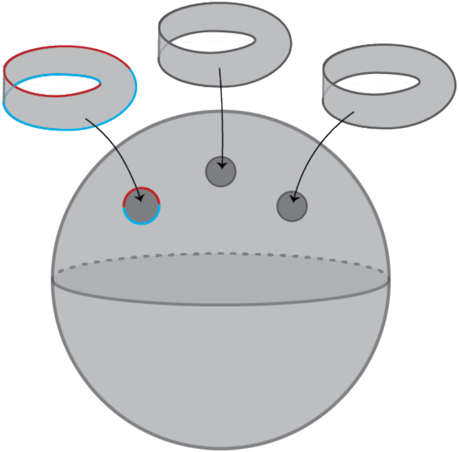
- In this talk, we deal with connected compact surfaces.
- They are classified by their **orientability** and their genus.
- An **embedding** of G on a surface S is an injective map $G \hookrightarrow S$.



- 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.

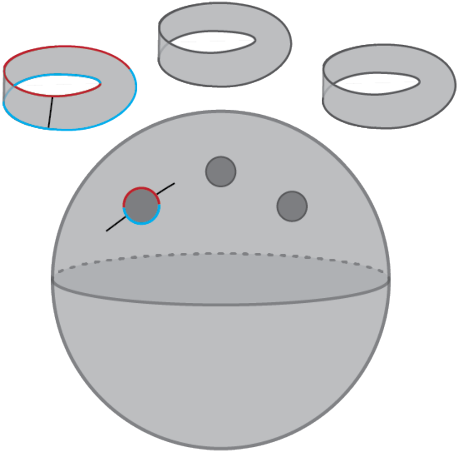
A geometric model: Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing: A **cross-cap drawing**.



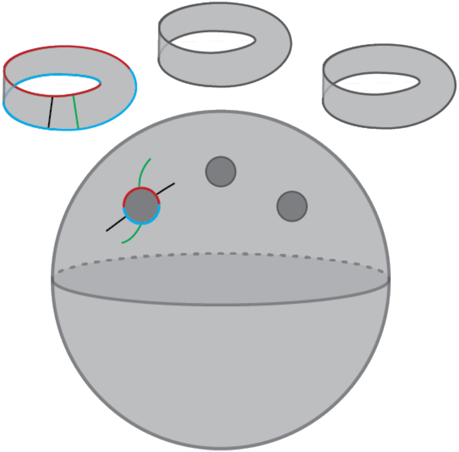
A geometric model: Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing: A **cross-cap drawing**.



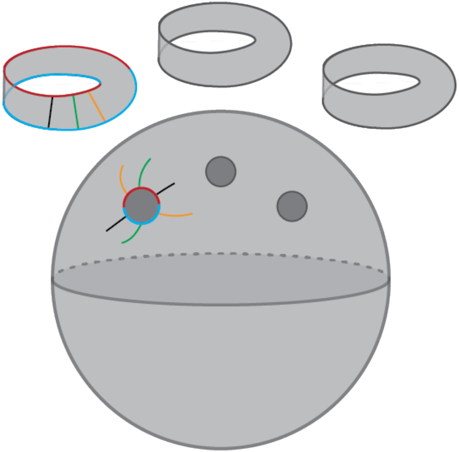
A geometric model: Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing: A **cross-cap drawing**.



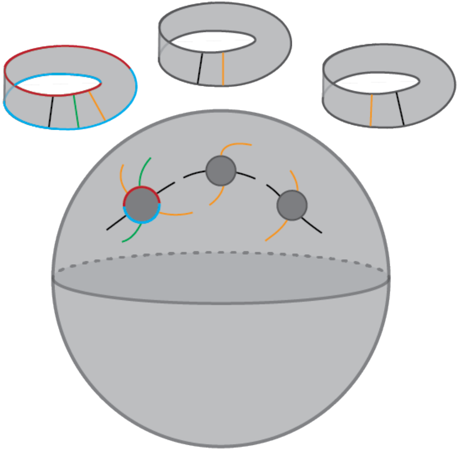
A geometric model: Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing: A **cross-cap drawing**.



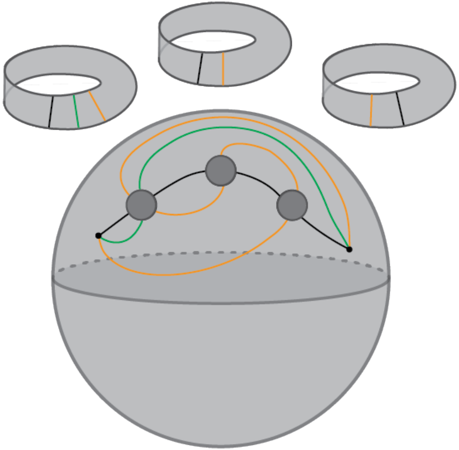
A geometric model: Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing: A **cross-cap drawing**.



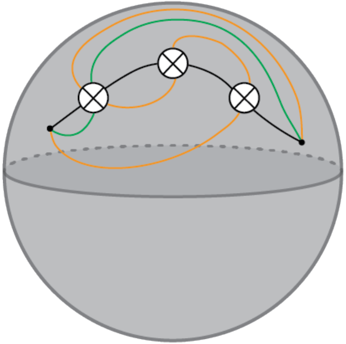
A geometric model: Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing: A **cross-cap drawing**.



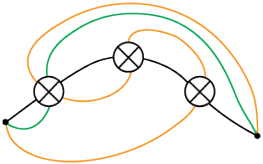
A geometric model: Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing: A **cross-cap drawing**.

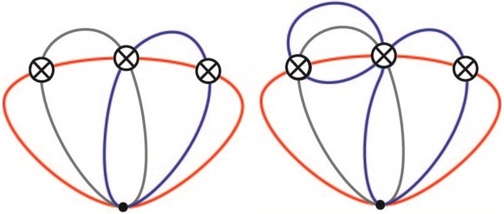


A geometric model: Cross-cap drawings

- One can represent a non-orientable embedding by a planar drawing: A **cross-cap drawing**.



- This **localization** of cross-caps is not "canonical"!



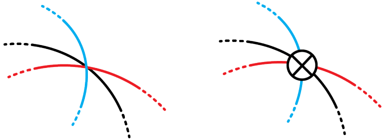
Question: How much can we control the complexity of the drawing?

From crossing numbers to non-orientable genus

Theorem (Mohar '07)

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

Cross-caps can be interpreted as multiple transverse crossings.

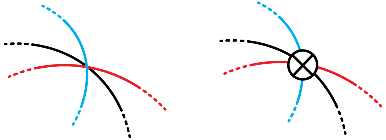


From crossing numbers to non-orientable genus

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.

→ Schaefer and Štefankovič disproved this.

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****.*

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.

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.

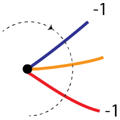
→ Schaefer and Štefankovič disproved this.

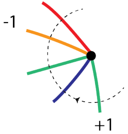
Theorem (F., Hubard, de Mesmay '23)

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

Embedding schemes

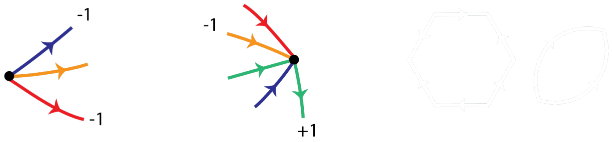
- An embedding for a graph, is entirely described by an **embedding scheme**:
  - the cyclic ordering of the edges around the vertex
  - (in the non-orientable case) a signature $+1$ or -1 associated to each edge





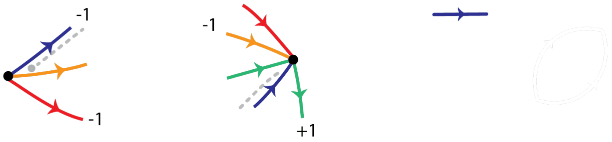
Embedding schemes

- 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:



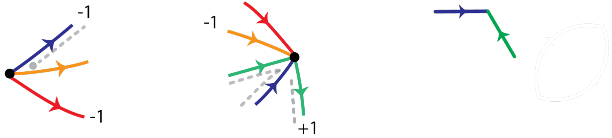
Embedding schemes

- 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:



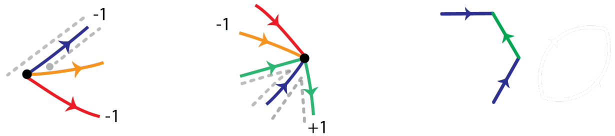
Embedding schemes

- 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:



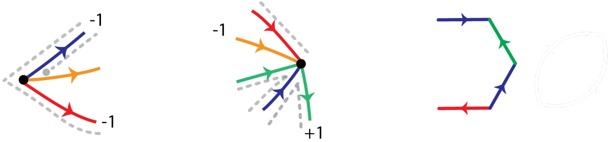
Embedding schemes

- 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:



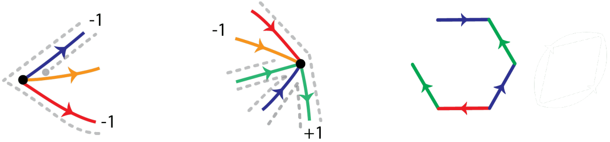
Embedding schemes

- 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:



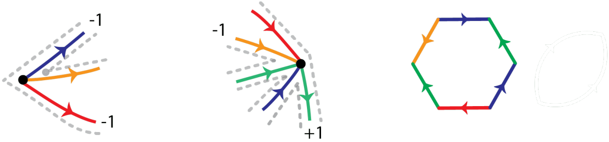
Embedding schemes

- 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:



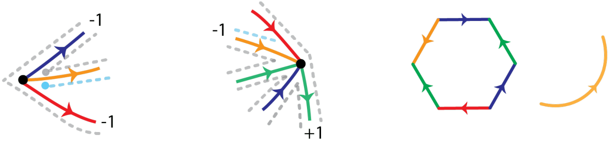
Embedding schemes

- 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:



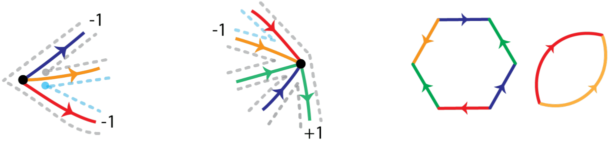
Embedding schemes

- 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:



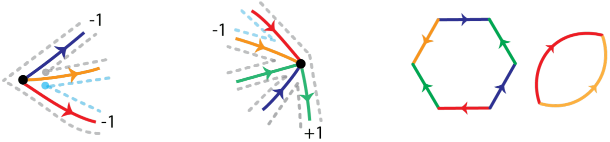
Embedding schemes

- 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

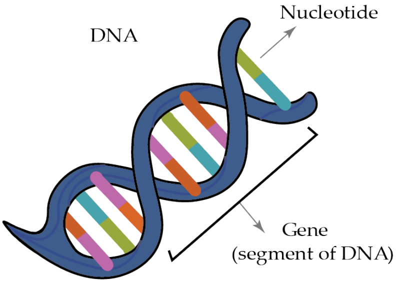


- I consider cross-cap drawings for an embedding scheme that "respect" the signatures of the edges.

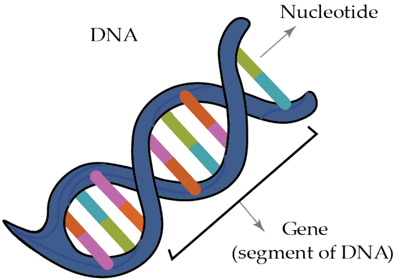
→ If an edge has signature **+1** (resp. **-1**) , it enters an **even** (resp. **odd**) number of cross-caps.

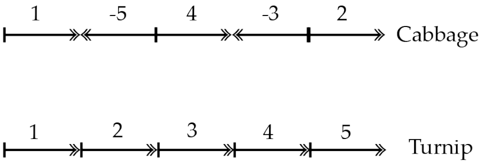
Main tool: Signed reversal distance algorithm

A "twist" in biology: Genome rearrangement



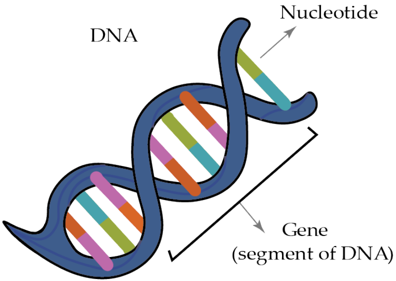
A "twist" in biology: Genome rearrangement

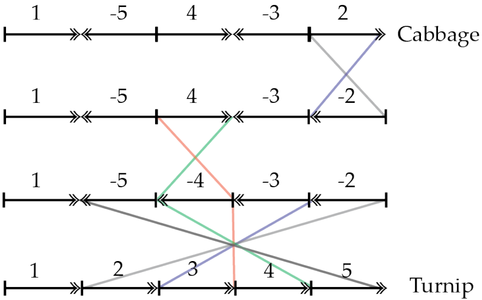




- The DNA of some species only differ by their gene sequences.

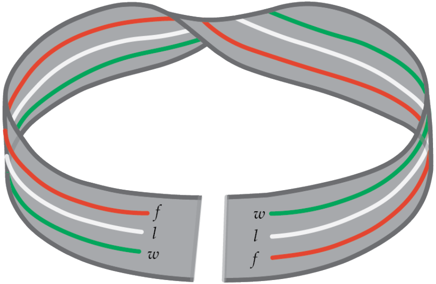
A "twist" in biology: Genome rearrangement

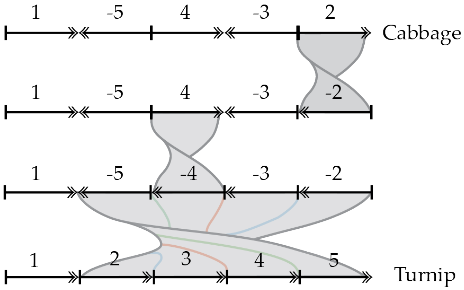




- The DNA of some species only differ by their gene sequences.
- The evolutionary distance between two species can be approximated by the number of **reversals** needed to transform one gene sequence into another.

A "twist" in biology: Genome rearrangement

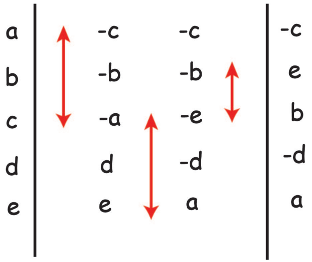




- The DNA of some species only differ by their gene sequences.
- The evolutionary distance between two species can be approximated by the number of **reversals** needed to transform one gene sequence into another.

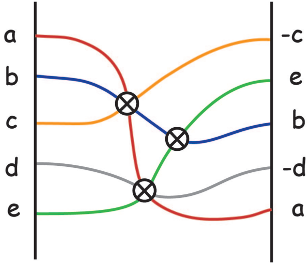
Signed reversal distance

- 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.



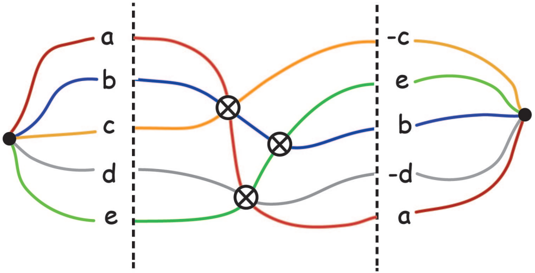
Signed reversal distance

- 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.



Signed reversal distance

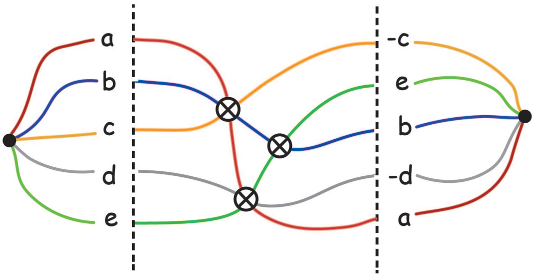
- The **signed reversal distance** between two signed permutations is the minimum number of **reversals** to go from one to the other.
- It is computable in **polynomial time** [Hannenhali-Pevzner '99].
- This has strong similarities with crosscap drawings.



Properties of Hannenhali-Pevzner algorithm

- The algorithm imposes an order on the cross-caps

→ if we orient edges from left to right, if an edge enters cross-cap 1 and then 2, then no edge can enter 2 and then 1.

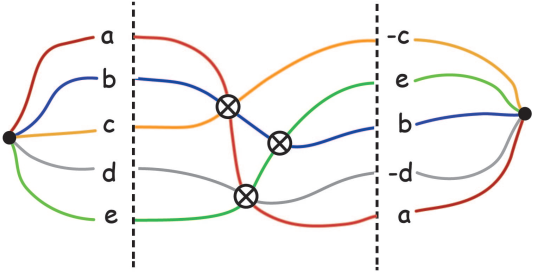


Properties of Hannenhali-Pevzner algorithm

- The algorithm imposes an order on the cross-caps

→ if we orient edges from left to right, if an edge enters cross-cap 1 and then 2, then no edge can enter 2 and then 1.

→ each edge enters each cross-cap at most once.

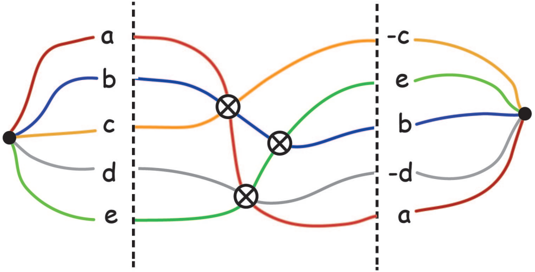


Properties of Hannenhali-Pevzner algorithm

- The algorithm imposes an order on the cross-caps

→ if we orient edges from left to right, if an edge enters cross-cap 1 and then 2, then no edge can enter 2 and then 1.

→ each edge enters each cross-cap at most once.

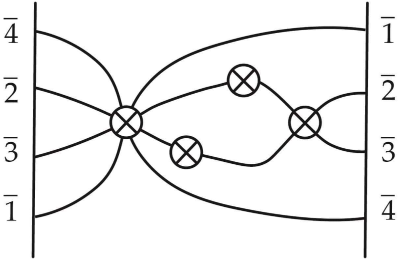


- Their algorithm uses more cross-caps/ reversals than the non-orientable genus/ *gcr* of the graph.

Blocks in Signed reversal distance

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

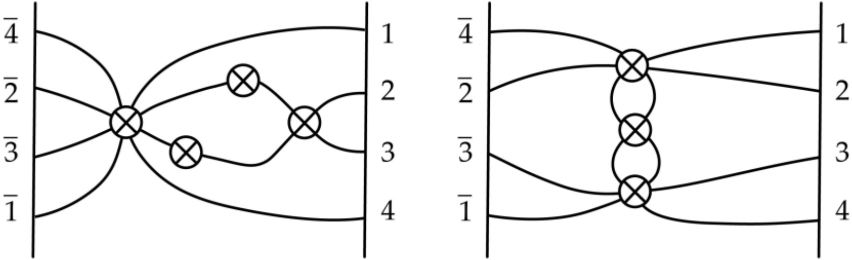
→ When there are no blocks, their algorithm exactly uses $g(G)$ reversals.



Blocks in Signed reversal distance

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

→ When there are no blocks, their algorithm exactly uses $g(G)$ reversals.



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

Sketch of proofs

The counter example

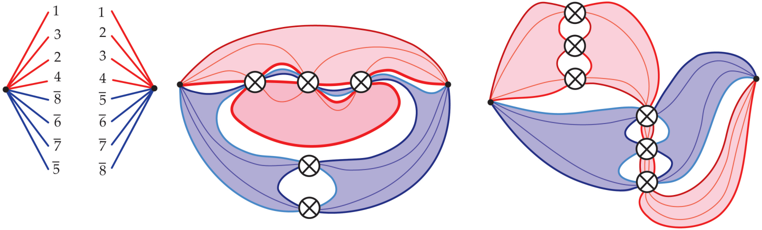
Mohar's (stronger) Conjecture 2 ('07)

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.*



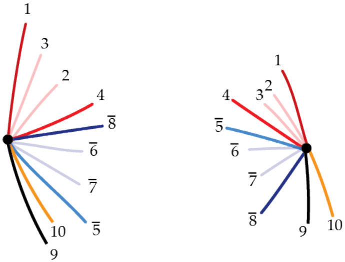
Cross-cap drawings of 2-vertex schemes

Theorem (F., Hubard, de Mesmay ’23)

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

Sketch of the proof:

→ **reduce** the scheme.



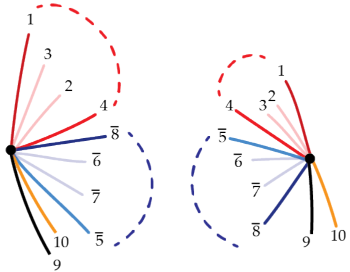
Cross-cap drawings of 2-vertex schemes

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.



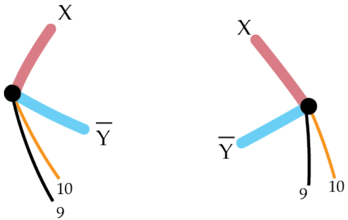
Cross-cap drawings of 2-vertex schemes

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

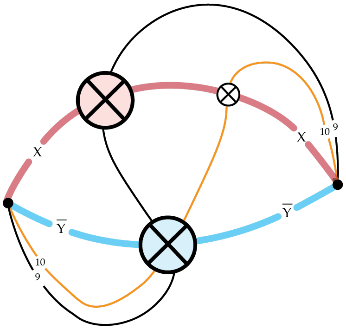
Theorem (F., Hubard, de Mesmay '23)

*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)

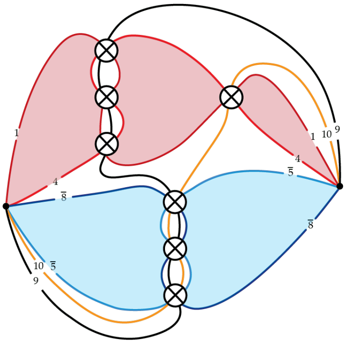
Apart from two exceptional families of graphs, all the 2-vertex loopless graphs embedded on non-orientable surfaces admit a ***perfect*** *cross-cap drawing.*

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.*

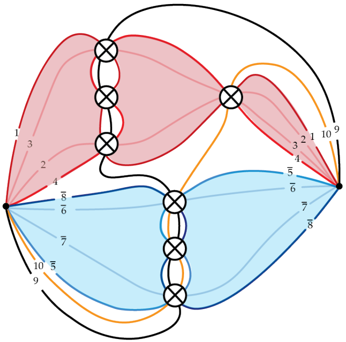
Sketch of the proof:

→ **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.*

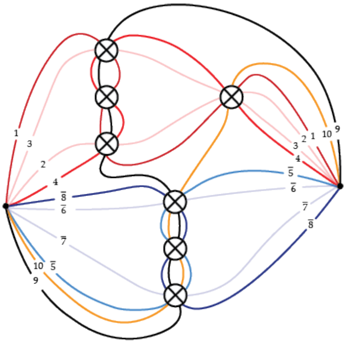
Sketch of the proof:

→ **reduce** the scheme.

→ apply Hannenhali-Pevzner algorithm.

→ **blow up** the cross-caps.

→ complete the drawing.

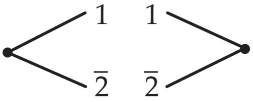


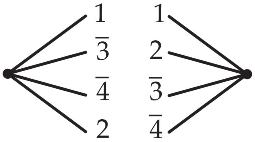
Cross-cap drawings of 2-vertex schemes

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.*

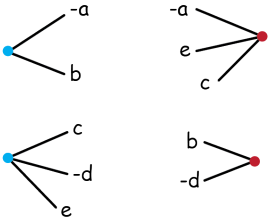
- In particular under standard models of random maps, almost all 2-vertex loopless embedded graphs satisfy Conjecture 2.





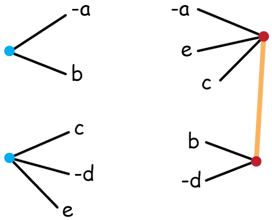
Conclusion

- Our algorithm for perfect cross-cap drawings can be extended to the case of loopless bipartite embedding schemes.



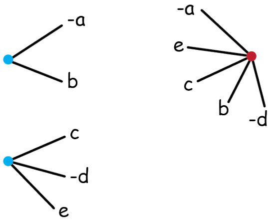
Conclusion

- Our algorithm for perfect cross-cap drawings can be extended to the case of loopless bipartite embedding schemes.



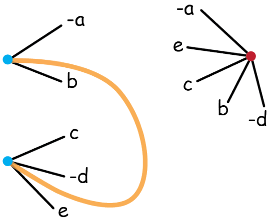
Conclusion

- Our algorithm for perfect cross-cap drawings can be extended to the case of loopless bipartite embedding schemes.



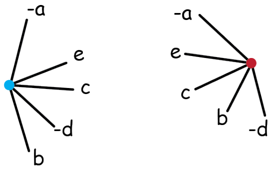
Conclusion

- Our algorithm for perfect cross-cap drawings can be extended to the case of loopless bipartite embedding schemes.



Conclusion

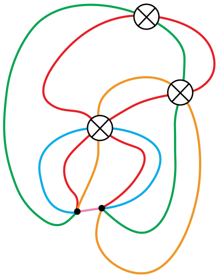
- Our algorithm for perfect cross-cap drawings can be extended to the case of loopless bipartite embedding schemes.



Mohar's Conjecture 1 ('07)

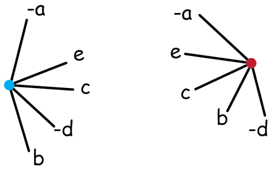
For every graph G, $\mathrm{gcr}(G) = \mathrm{dcr}(G)$.

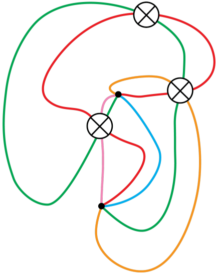
→ Allowing the graph to have more vertices, increases the possibility of having a **perfect** cross-cap drawing.



Conclusion

- Our algorithm for perfect cross-cap drawings can be extended to the case of loopless bipartite embedding schemes.





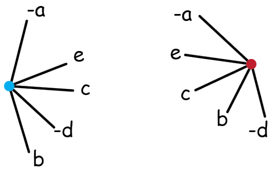
Mohar's Conjecture 1 ('07)

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

- Our algorithm for perfect cross-cap drawings can be extended to the case of loopless bipartite embedding schemes.



Thank You!

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.