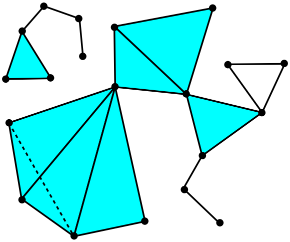
Embedding Graphs into 2-Dimensional Simplicial Complexes

Éric Colin de Verdière

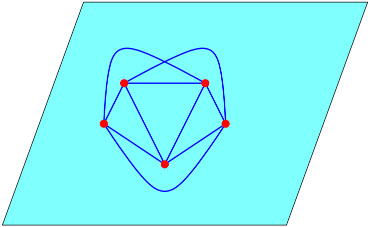
CNRS, LIGM, Marne-la-Vallée, France

Joint work with Thomas Magnard



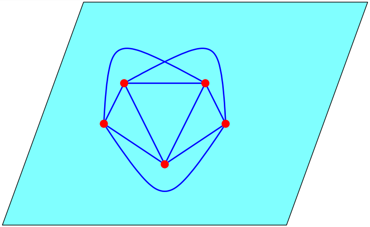
Embedding graphs in the plane

Input: A graph G
Question: Does G have a topological embedding in the plane?



Embedding graphs in the plane

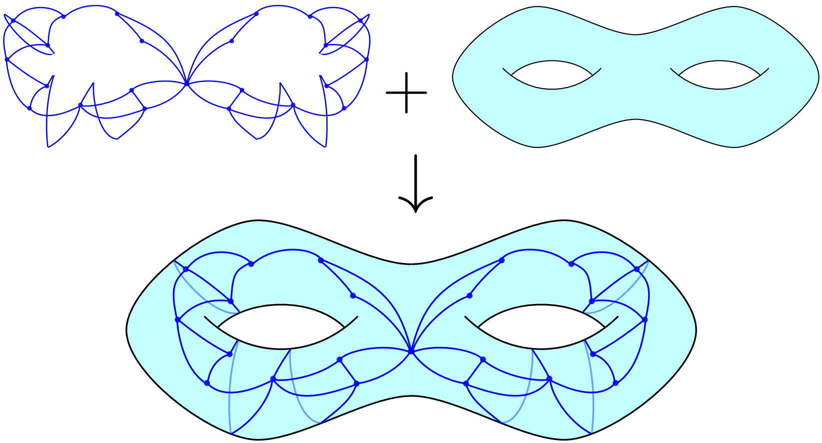
Input: A graph G
Question: Does G have a topological embedding in the plane?



This is planarity testing, solvable in linear time [Hopcroft, Tarjan, 1974].

Embedding graphs on surfaces

Input: A graph G and an integer g
Question: Does G have a topological embedding in the orientable (or non-orientable) surface of genus g?



Embedding graphs on surfaces

Input: A graph G and an integer g
Question: Does G have a topological embedding in the orientable (or non-orientable) surface of genus g?

NP-complete [Thomassen, 1989]

Existing algorithms

- [Mohar, 1999]: $f(g) \cdot n$
- [Kawarabayashi, Mohar, Reed, 2008]: $2^{\text{poly}(g)} \cdot n$
- Graph minor theory: $f(g) \cdot n^3$ [Robertson and Seymour, 1995]+[Adler et al., 2008].

Motivation

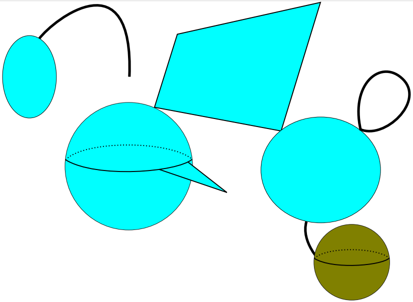
Many problems can be solved faster for graphs embeddable on a fixed surface than for general graphs (shortest paths, (multi)flows and (multi)cuts, disjoint paths, (sub)graph isomorphism, TSP, Steiner trees, etc.)

Embedding graphs in topological spaces

Input:
- A graph G,
- a topological space T.

Question: Does G have a topological embedding on T?

- model topological spaces as simplicial complexes;
- actually as 2-dimensional simplicial complexes, or 2-complexes: graphs on which we attach a triangle to some of its cycles of length 3.

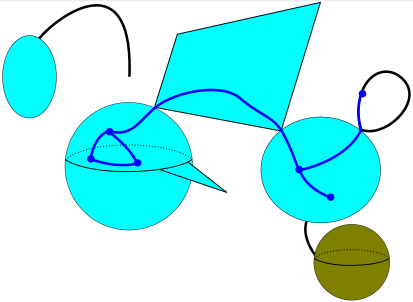


Embedding graphs in topological spaces

Input:
- A graph G,
- a topological space T.

Question: Does G have a topological embedding on T?

- model topological spaces as simplicial complexes;
- actually as 2-dimensional simplicial complexes, or 2-complexes: graphs on which we attach a triangle to some of its cycles of length 3.

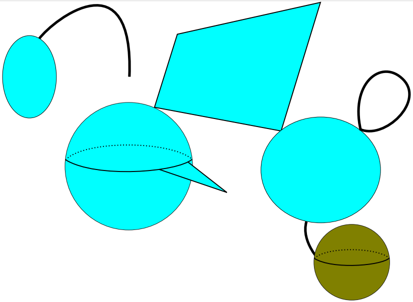


Embedding graphs in topological spaces

Input:
- A graph G,
- a topological space T.

Question: Does G have a topological embedding on T?

- model topological spaces as simplicial complexes;
- actually as 2-dimensional simplicial complexes, or 2-complexes: graphs on which we attach a triangle to some of its cycles of length 3.

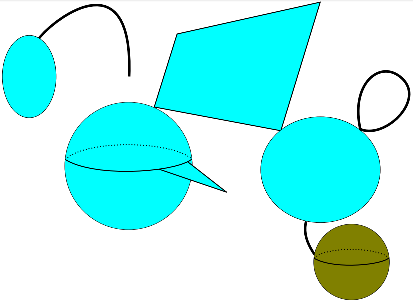


Embedding graphs in topological spaces

Input:
- A graph G,
- a topological space T.

Question: Does G have a topological embedding on T?

- model topological spaces as simplicial complexes;
- actually as 2-dimensional simplicial complexes, or 2-complexes: graphs on which we attach a triangle to some of its cycles of length 3.

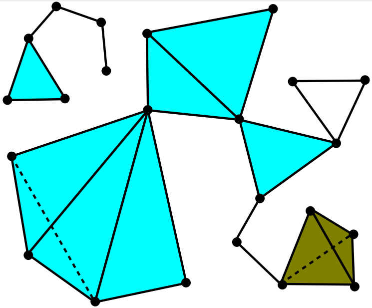


Embedding graphs in topological spaces

Input:
- A graph G,
- a ~~topological space T~~ simplicial complex C.

Question: Does G have a topological embedding on C?

- model topological spaces as simplicial complexes;
- actually as 2-dimensional simplicial complexes, or 2-complexes: graphs on which we attach a triangle to some of its cycles of length 3.

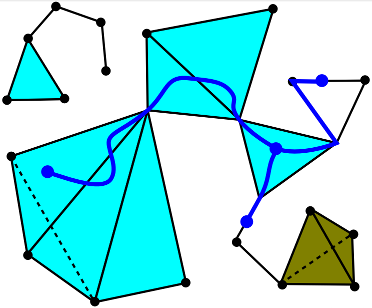


Embedding graphs in topological spaces

Input:
- A graph G,
- a ~~topological space T~~ simplicial complex C.

Question: Does G have a topological embedding on C?

- model topological spaces as simplicial complexes;
- actually as 2-dimensional simplicial complexes, or 2-complexes: graphs on which we attach a triangle to some of its cycles of length 3.

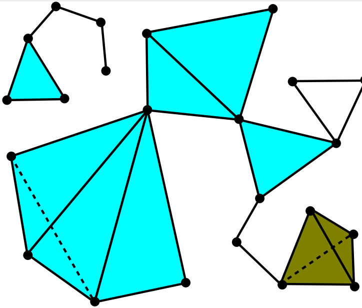


Embedding graphs in topological spaces

Input:
- A graph G,
- a ~~topological space $\mathcal{T}$~~ 2-dim simplicial complex C.

Question: Does G have a topological embedding on C?

- model topological spaces as simplicial complexes;
- actually as 2-dimensional simplicial complexes, or 2-complexes: graphs on which we attach a triangle to some of its cycles of length 3.

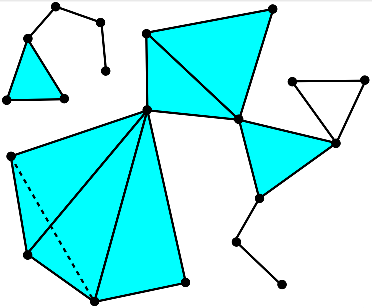


Embedding graphs in topological spaces

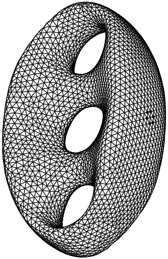
Input:
- A graph G,
- a ~~topological space $\mathcal{T}$~~ 2-dim simplicial complex C.

Question: Does G have a topological embedding on C?

- model topological spaces as simplicial complexes;
- actually as 2-dimensional simplicial complexes, or 2-complexes: graphs on which we attach a triangle to some of its cycles of length 3.



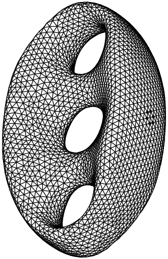
Our result



NP-hardness

- Any surface of genus g is (homeomorphic to) a 2-complex with $O(g)$ simplices;
- deciding embeddability of a graph on a surface is NP-hard [Thomassen, 1989];
- thus, our problem is NP-hard.

Our result



NP-hardness

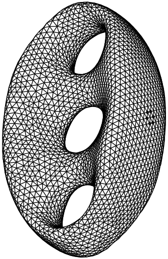
- Any surface of genus g is (homeomorphic to) a 2-complex with $O(g)$ simplices;
- deciding embeddability of a graph on a surface is NP-hard [Thomassen, 1989];
- thus, our problem is NP-hard.

Algorithm [CdV, Magnard, 2021]

Given:
- a graph G with n vertices and edges
- a 2-complex C with c simplices

one can decide whether G has an embedding into C in time $f(c) \cdot n^2$.

Our result



NP-hardness

- Any surface of genus g is (homeomorphic to) a 2-complex with $O(g)$ simplices;
- deciding embeddability of a graph on a surface is NP-hard [Thomassen, 1989];
- thus, our problem is NP-hard.

Algorithm [CdV, Magnard, 2021]

Given:
- a graph G with n vertices and edges
- a 2-complex C with c simplices

one can decide whether G has an embedding into C in time $f(c) \cdot n^2$.

Remarks

- independent from previous works to embed graphs on surfaces;
- previous result in $f(c) \cdot n^{O(c)}$ [CdV, Magnard, Mohar, 2018].

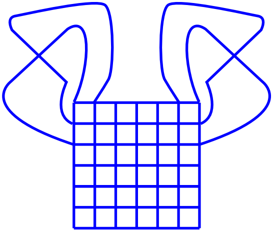
Special case: The crossing number problem

Problem

Input: a graph G and an integer k.
Problem: decide whether G can be drawn in the plane with at most k crossings.

Results

- NP-hard [Garey, 1983],
- linear-time for fixed k [Kawarabayashi and Reed, 2007],
- our result directly implies a quadratic-time algorithm for a more general problem.



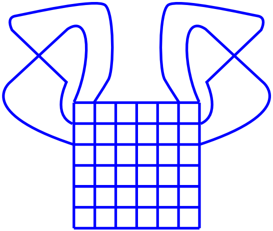
Special case: The crossing number problem

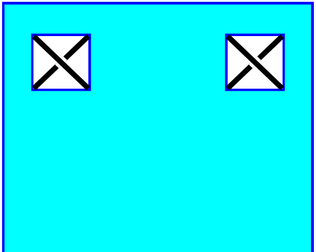
Problem

Input: a graph G and an integer k.
Problem: decide whether G can be drawn in the plane with at most k crossings.

Results

- NP-hard [Garey, 1983],
- linear-time for fixed k [Kawarabayashi and Reed, 2007],
- our result directly implies a quadratic-time algorithm for a more general problem.





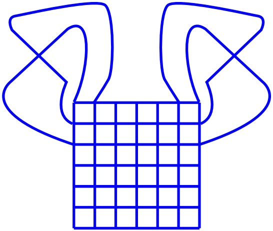
Special case: The crossing number problem

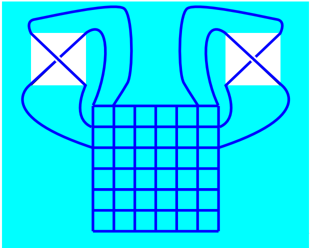
Problem

Input: a graph G and an integer k.
Problem: decide whether G can be drawn in the plane with at most k crossings.

Results

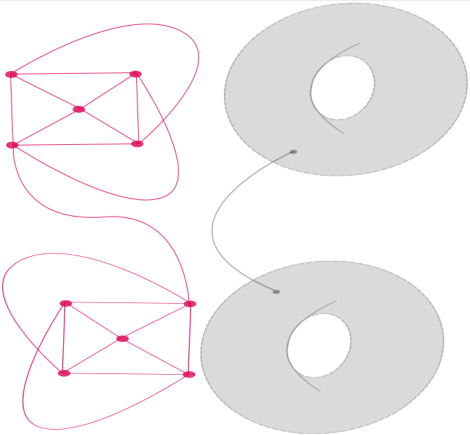
- NP-hard [Garey, 1983],
- linear-time for fixed k [Kawarabayashi and Reed, 2007],
- our result directly implies a quadratic-time algorithm for a more general problem.





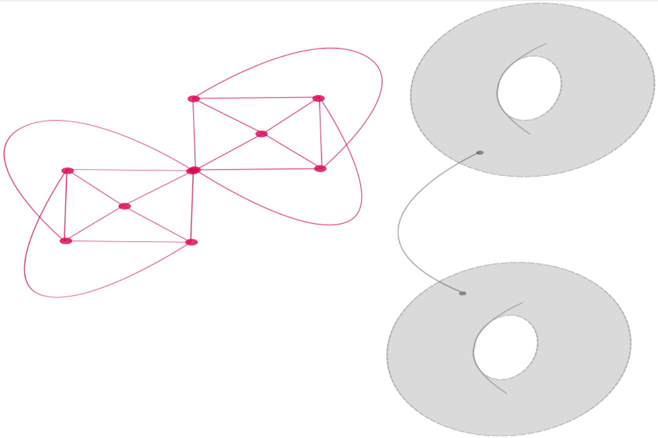
Why 2-complexes are/look harder to handle than surfaces

The graphs embeddable on a given 2-complex is not minor-closed.



Why 2-complexes are/look harder to handle than surfaces

The graphs embeddable on a given 2-complex is not minor-closed.



Why 2-complexes are/look harder to handle than surfaces

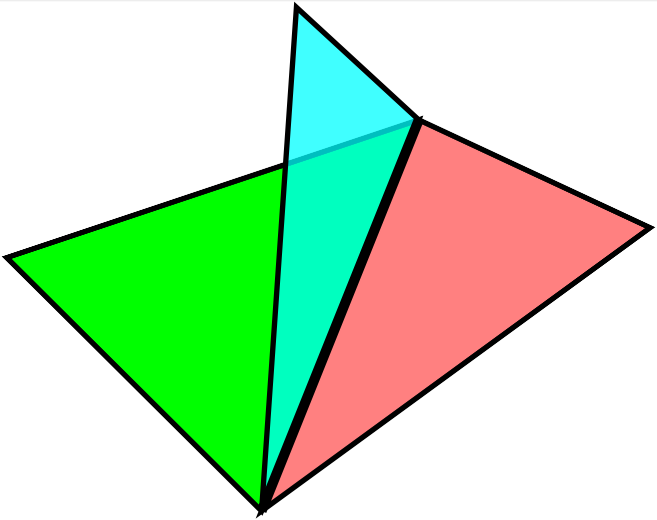
The graphs embeddable on a given 2-complex is not minor-closed.

Some problems are harder on 2-complexes than on surfaces:

problem	surfaces	2-complexes
homeomorphism	linear-time	same as graph isomorphism [Ó Dúnlaing et al., 2000]
deciding contractibility of curves	linear-time [Dey and Guha, 1999]	undecidable [Boone, 1959]

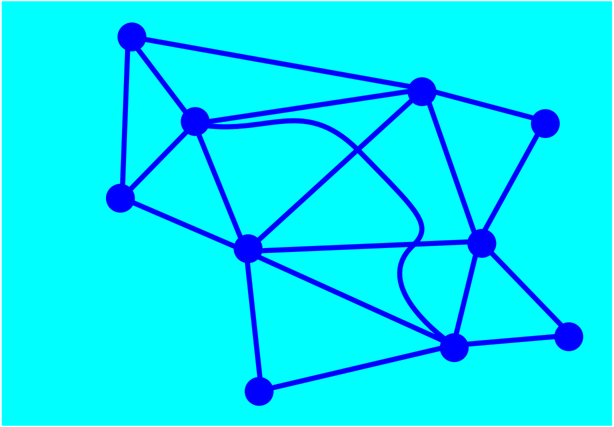
Why 2-complexes are still manageable

Every graph is embeddable in a 3-book. So wlog C contains no 3-book.

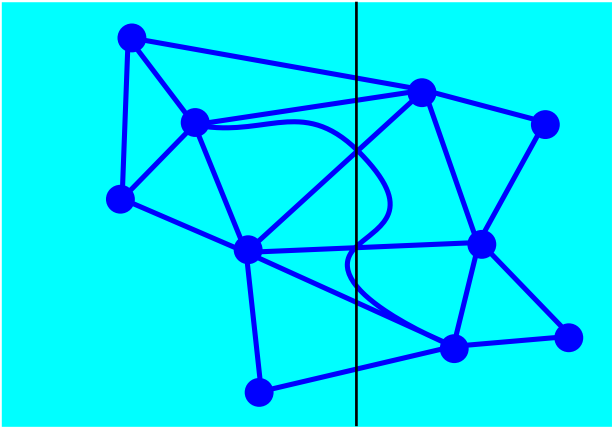


Why 2-complexes are still manageable

Every graph is embeddable in a 3-book. So wlog C contains no 3-book.

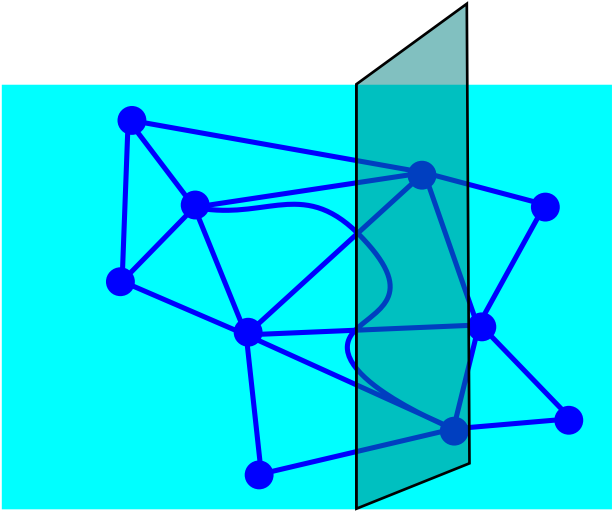


Every graph is embeddable in a 3-book. So wlog C contains no 3-book.



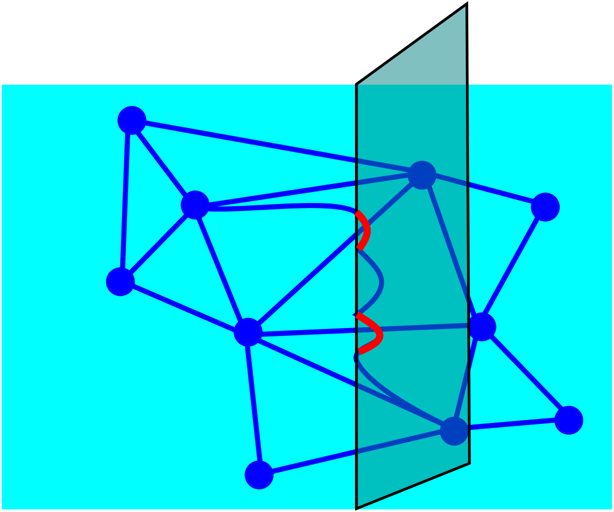
Why 2-complexes are still manageable

Every graph is embeddable in a 3-book. So wlog C contains no 3-book.



Why 2-complexes are still manageable

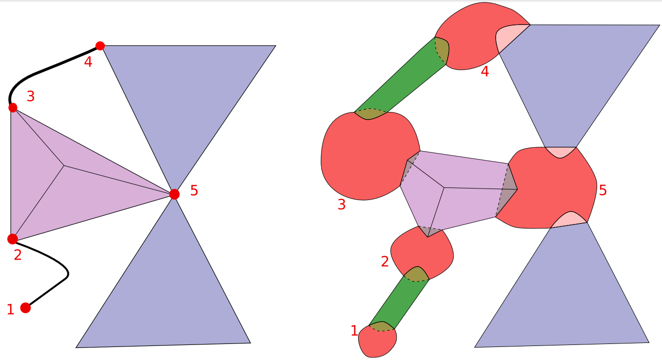
Every graph is embeddable in a 3-book. So wlog C contains no 3-book.



Why 2-complexes are still manageable

Every graph is embeddable in a 3-book. So wlog C contains no 3-book.

Let C be a 2-complex without 3-book. Every graph embeddable on C is embeddable on a surface of genus $O(c)$.

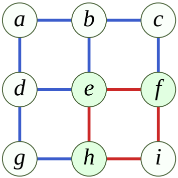


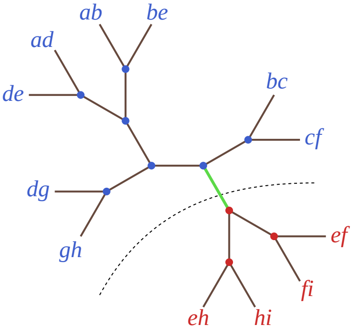
Sketch of the algorithm

Overview

Branch decomposition of G

- Branch decomposition B: unrooted binary tree with leaves in bijection with the edges of G;
- each edge of B induces a bipartition of the edges of G.
- B has width $\leq k$ if the middle set, the set of vertices appearing on both sides of an induced bipartition, is always $\leq k$.

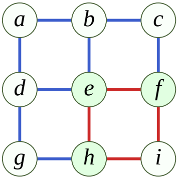


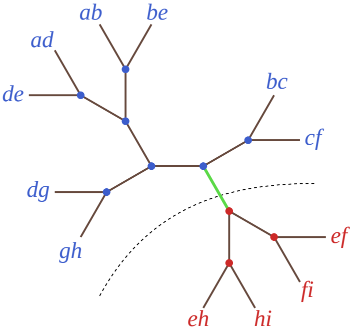


Credits: wikipedia

Branch decomposition of G

- Branch decomposition B: unrooted binary tree with leaves in bijection with the edges of G;
- each edge of B induces a bipartition of the edges of G.
- B has width $\leq k$ if the middle set, the set of vertices appearing on both sides of an induced bipartition, is always $\leq k$.





Credits: wikipedia

Overview

Branch decomposition of G

- Branch decomposition B: unrooted binary tree with leaves in bijection with the edges of G;
- each edge of B induces a bipartition of the edges of G.
- B has width $\leq k$ if the middle set, the set of vertices appearing on both sides of an induced bipartition, is always $\leq k$.

Standard strategy

1. Reduce to the case where G has a branch decomposition of width $w = \text{poly}(c)$;
2. use dynamic programming on a branch decomposition of G.

Irrelevant vertex method

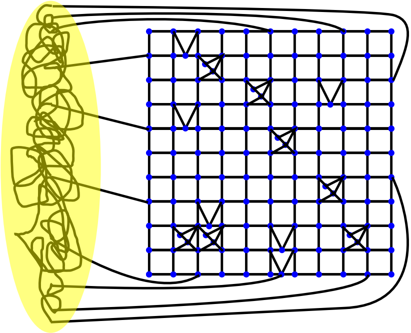
Inspired from [Kociumaka and Ma. Pilipczuk, 2019]

If G has no branch decomposition of small width, then it has (a subdivision of) a large grid [Robertson and Seymour, 1995].

In $O(n)$ time, find a *planar* subgraph of G

- containing a subdivision of a large grid
- connected to the rest of the graph only via its outside cycle.

The central vertex is irrelevant.



Irrelevant vertex method

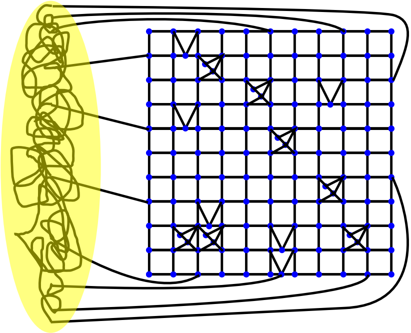
Inspired from [Kociumaka and Ma. Pilipczuk, 2019]

If G has no branch decomposition of small width, then it has (a subdivision of) a large grid [Robertson and Seymour, 1995].

In $O(n)$ time, find a *planar* subgraph of G

- containing a subdivision of a large grid
- connected to the rest of the graph only via its outside cycle.

The central vertex is irrelevant.



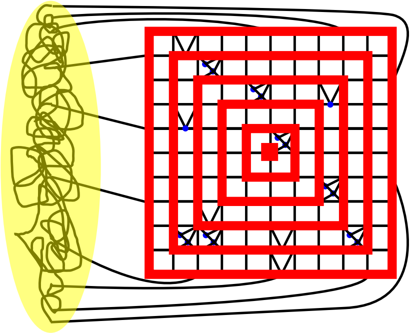
Irrelevant vertex method

Inspired from [Kociumaka and Ma. Pilipczuk, 2019]

If G has no branch decomposition of small width, then it has (a subdivision of) a large grid [Robertson and Seymour, 1995].
In $O(n)$ time, find a *planar* subgraph of G

- containing a subdivision of a large grid
- connected to the rest of the graph only via its outside cycle.

The central vertex is irrelevant.



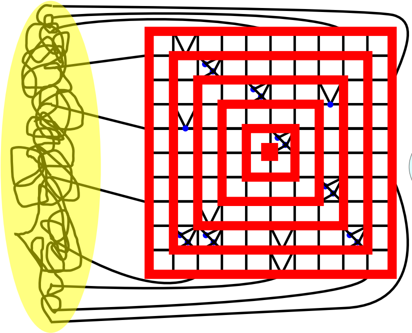
Irrelevant vertex method

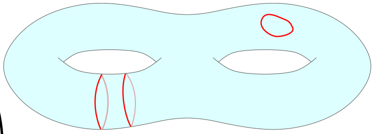
Inspired from [Kociumaka and Ma. Pilipczuk, 2019]

If G has no branch decomposition of small width, then it has (a subdivision of) a large grid [Robertson and Seymour, 1995].
In $O(n)$ time, find a *planar* subgraph of G

- containing a subdivision of a large grid
- connected to the rest of the graph only via its outside cycle.

The central vertex is irrelevant.





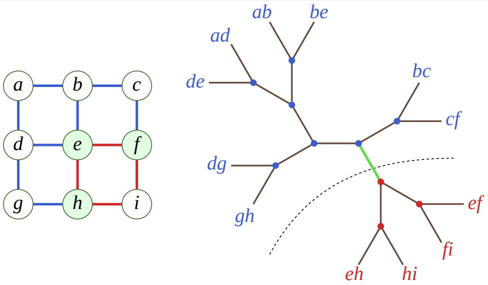
Bottom-up dynamic programming

Intuition

Choose a root of the branch decomposition B and apply bottom-up dynamic programming.

For every induced bipartition (E_1, E_2), memoize

- all the possible shapes of "regions" of C that can be occupied by E_2
- and, for each such shape, the location of the vertices of the middle set of (E_1, E_2) (the "boundary" of the region).



Credits: wikipedia

Bottom-up dynamic programming

Intuition

Choose a root of the branch decomposition B and apply bottom-up dynamic programming.

For every induced bipartition (E_1, E_2), memoize

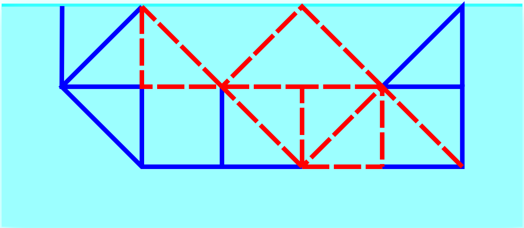
- all the possible shapes of "regions" of C that can be occupied by E_2
- and, for each such shape, the location of the vertices of the middle set of (E_1, E_2) (the "boundary" of the region).

Problems

1. represent such regions,
2. prove that there are not too many such possibilities.

Representing regions: Partitioning graphs

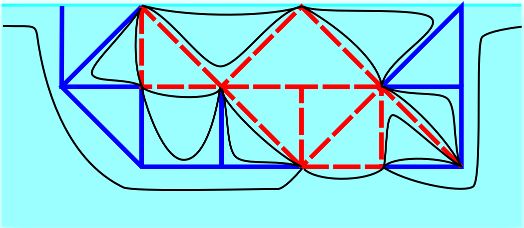
Assume G is embedded on C. To every induced bipartition (E_1, E_2), we define a partitioning graph $P(E_1, E_2)$ separating E_1 and E_2.



- Regions are labelled 0 (no part of the graph), 1 (E_1), 2 (E_2);
- vertices are labelled essentially by vertices of the middle set.

Representing regions: Partitioning graphs

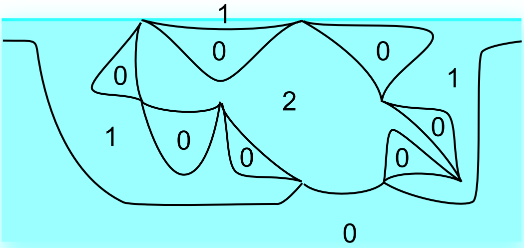
Assume G is embedded on C. To every induced bipartition (E_1, E_2), we define a partitioning graph $P(E_1, E_2)$ separating E_1 and E_2.



- Regions are labelled 0 (no part of the graph), 1 (E_1), 2 (E_2);
- vertices are labelled essentially by vertices of the middle set.

Representing regions: Partitioning graphs

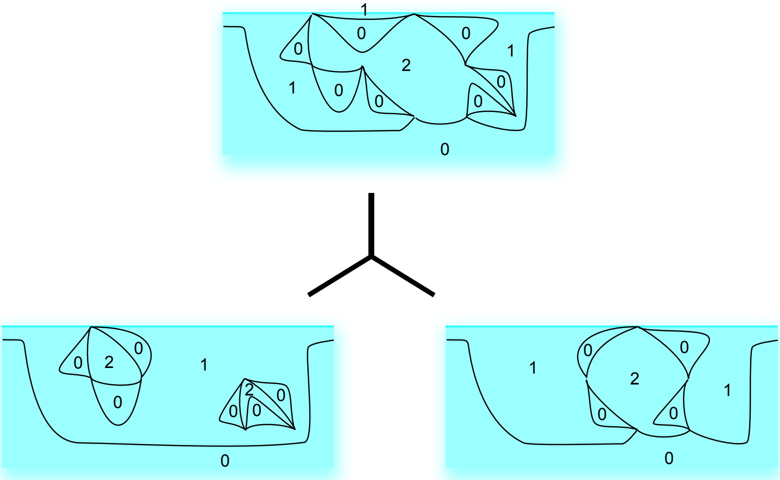
Assume G is embedded on C. To every induced bipartition (E_1, E_2), we define a partitioning graph $P(E_1, E_2)$ separating E_1 and E_2.



- Regions are labelled 0 (no part of the graph), 1 (E_1), 2 (E_2);
- vertices are labelled essentially by vertices of the middle set.

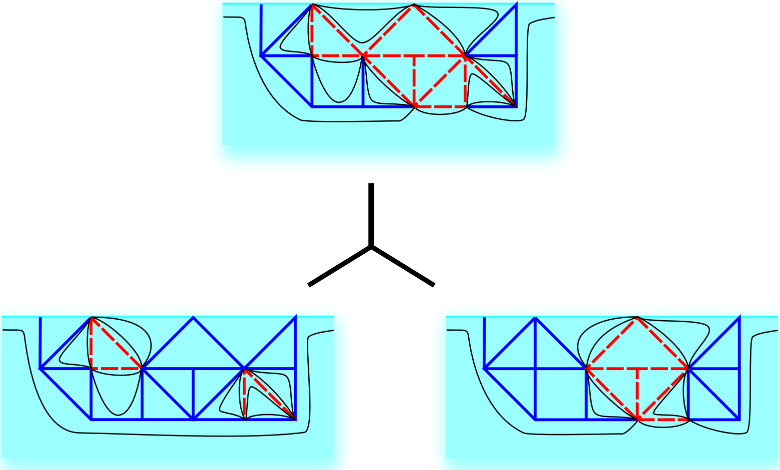
Representing regions: Partitioning graphs

Assume G is embedded on C. To every induced bipartition (E_1, E_2), we define a partitioning graph $P(E_1, E_2)$ separating E_1 and E_2.



Representing regions: Partitioning graphs

Assume G is embedded on C. To every induced bipartition (E_1, E_2), we define a partitioning graph $P(E_1, E_2)$ separating E_1 and E_2.



Number of possibilities in the dynamic program

Main Lemma

If G embeds on C, then it has an embedding in which every partitioning graph (w.r.t. B) has $O(c + w)$ vertices, edges, and faces.

Sketch of proof

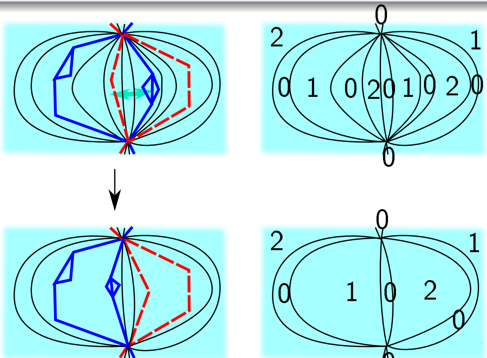
- By moving around monogons and bigons: $P(E_1, E_2)$ does not have too many monogons or bigons.

Main Lemma

If G embeds on C, then it has an embedding in which every partitioning graph (w.r.t. B) has $O(c+w)$ vertices, edges, and faces.

Sketch of proof

- By moving around monogons and bigons: $P(E_1, E_2)$ does not have too many monogons or bigons.



Number of possibilities in the dynamic program

Main Lemma

If G embeds on C, then it has an embedding in which every partitioning graph (w.r.t. B) has $O(c + w)$ vertices, edges, and faces.

Sketch of proof

- By moving around monogons and bigons: $P(E_1, E_2)$ does not have too many monogons or bigons.
- Need to check that such operations are compatible for all induced bipartitions (E_1, E_2).

Number of possibilities in the dynamic program

Main Lemma

If G embeds on C, then it has an embedding in which every partitioning graph (w.r.t. B) has $O(c+w)$ vertices, edges, and faces.

Sketch of proof

- By moving around monogons and bigons: $P(E_1, E_2)$ does not have too many monogons or bigons.
- Need to check that such operations are compatible for all induced bipartitions (E_1, E_2).

Fact

Up to homeomorphism, there are $(c+w)^{O(c+w)}$ embeddings of graphs with $O(c+w)$ vertices, edges, and faces into C.

Number of possibilities in the dynamic program

Main Lemma

If G embeds on C, then it has an embedding in which every partitioning graph (w.r.t. B) has $O(c+w)$ vertices, edges, and faces.

Sketch of proof

- By moving around monogons and bigons: $P(E_1, E_2)$ does not have too many monogons or bigons.
- Need to check that such operations are compatible for all induced bipartitions (E_1, E_2).

Fact

Up to homeomorphism, there are $(c+w)^{O(c+w)}$ embeddings of graphs with $O(c+w)$ vertices, edges, and faces into C.

+ many other details (data structures for graphs on 2-complexes; definition of partitioning graph; assuming cellular embeddings. . .).

Thank you for your attention!
Questions?