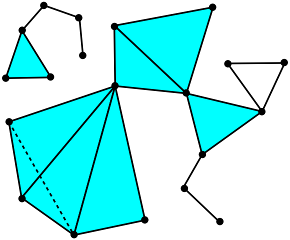
# Embeddability of Graphs into 2-Dimensional Simplicial Complexes

Éric Colin de Verdière

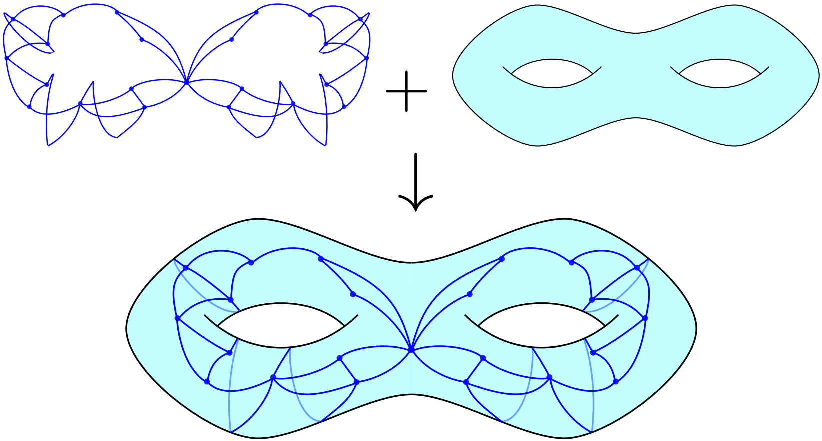
CNRS, LIGM, Marne-la-Vallée, France

Joint work with Thomas Magnard



# Embedding graphs on surfaces

- Input: A graph $G$ with $n$ vertices and edges; a surface $S$ specified by its genus $g$ and its orientability
- Question: Decide whether $G$ has a topological embedding (a crossing-free drawing) into $S$.



# Embedding graphs on surfaces

- Input: A graph $G$ with $n$ vertices and edges; a surface $S$ specified by its genus $g$ and its orientability
- Question: Decide whether $G$ has a topological embedding (a crossing-free drawing) into $S$.

**Motivation: Algorithms for graphs on surfaces**

Many problems can be solved faster for graphs embedded 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 on surfaces

- Input: A graph $G$ with $n$ vertices and edges; a surface $S$ specified by its genus $g$ and its orientability
- Question: Decide whether $G$ has a topological embedding (a crossing-free drawing) into $S$.

## Motivation: Algorithms for graphs on surfaces

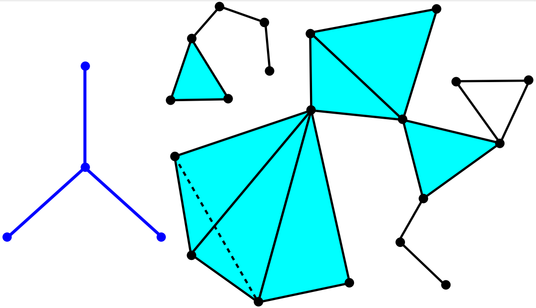
Many problems can be solved faster for graphs embedded on a fixed surface than for general graphs (shortest paths, (multi)flows and (multi)cuts, disjoint paths, (sub)graph isomorphism, TSP, Steiner trees, etc.)

## Existing results

- [Thomassen, 1989]: NP-hard (when $g$ is part of the input)
- [Mohar, 1999]: $f(g) \cdot n$ (very technical)
- [Kawarabayashi et al., 2008]: $2^{\mathrm{poly}(g)} \cdot n$ (only appeared in extended abstract)
- Graph minor theory: $f(g) \cdot n^3$ [Robertson and Seymour, 1995]+[Adler et al., 2008].

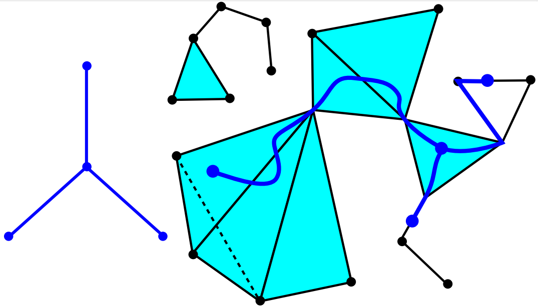
# Our goal: beyond surfaces

- Input: A graph $G$; a 2-dimensional simplicial complex $C$
- Question: Decide whether $G$ embeds into $C$.



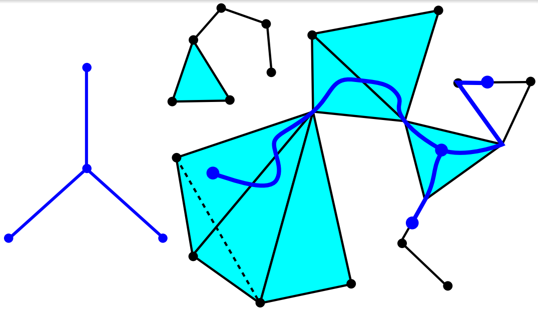
# Our goal: beyond surfaces

- Input: A graph $G$; a 2-dimensional simplicial complex $C$
- Question: Decide whether $G$ embeds into $C$.



# Our goal: beyond surfaces

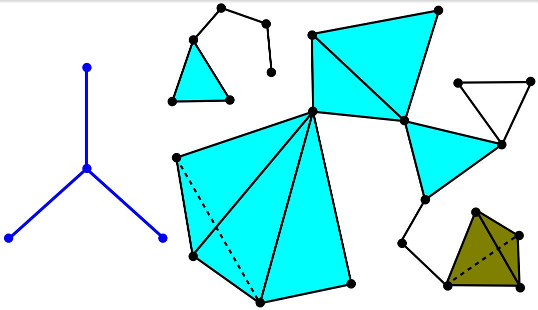
- Input: A graph $G$; a 2-dimensional simplicial complex $C$
- Question: Decide whether $G$ embeds into $C$.



- everything is topological: no constraint on the embedding;
- actually, “2-dimensional” is an unnecessary restriction;
- NP-hard (surfaces are 2-complexes);
- the set of graphs embeddable on $C$ is not minor-closed;
- encompasses other known problems, e.g., crossing number.

# Our goal: beyond surfaces

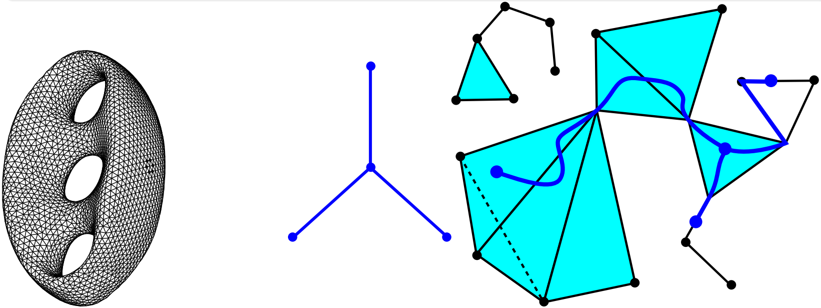
- Input: A graph $G$; a 2-dimensional simplicial complex $C$
- Question: Decide whether $G$ embeds into $C$.



- everything is topological: no constraint on the embedding;
- actually, "2-dimensional" is an unnecessary restriction;
- NP-hard (surfaces are 2-complexes);
- the set of graphs embeddable on $C$ is not minor-closed;
- encompasses other known problems, e.g., crossing number.

# Our goal: beyond surfaces

- Input: A graph $G$; a 2-dimensional simplicial complex $C$
- Question: Decide whether $G$ embeds into $C$.



- everything is topological: no constraint on the embedding;
- actually, "2-dimensional" is an unnecessary restriction;
- NP-hard (surfaces are 2-complexes);
- the set of graphs embeddable on $C$ is not minor-closed;
- encompasses other known problems, e.g., crossing number.

# Our result

## Our result

An algorithm with running time $2^{\mathrm{poly}(c)} \cdot n^2$ where

- $n$ is the number of vertices and edges of $G$;
- $c$ is the number of simplices of $C$.

# Our result

## Our result

An algorithm with running time $2^{\text{poly}(c)} \cdot n^2$ where

- $n$ is the number of vertices and edges of $G$;
- $c$ is the number of simplices of $C$.

## Features

- Our algorithm is independent from the existing algorithms for surfaces, and simpler. . .
- but quadratic in $n$ instead of linear.
- Main strategy of the algorithm:
  - reduce to the case where $G$ has branchwidth poly($c$) (*irrelevant vertex method*),
  - use dynamic programming.

# Our result

## Our result

An algorithm with running time $2^{\mathrm{poly}(c)} \cdot n^2$ where

- $n$ is the number of vertices and edges of $G$;
- $c$ is the number of simplices of $C$.

## Features

- Our algorithm is independent from the existing algorithms for surfaces, and simpler...
- but quadratic in $n$ instead of linear.
- Main strategy of the algorithm:
  - reduce to the case where $G$ has branchwidth poly($c$) (*irrelevant vertex method*),
  - use dynamic programming.

**THANKS FOR YOUR ATTENTION!**