Aalto University
School of Science

Using Tutte Paths for Finding Long Cycles in Planar Graphs

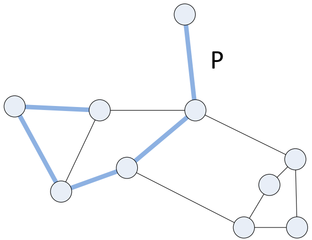
Andreas Schmid

August 12, 2019

Overview

But first... P-Bridges

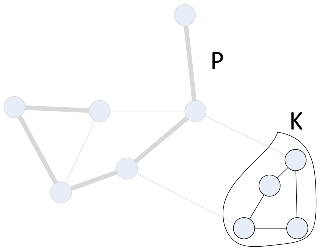
For a path *P* in *G*, a *P-bridge J* is



But first... P-Bridges

For a path P in G, a *P-bridge* J is

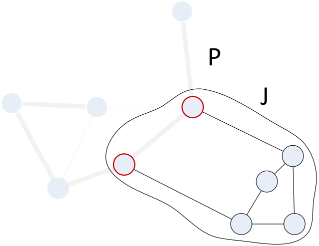
- a component K of $G \setminus V(P)$



But first... P-Bridges

For a path P in G, a *P-bridge* J is

- a component K of $G \setminus V(P)$
- **plus** the edges connecting K to P and their end-vertices (**attachments**).



Tutte paths - Definition

Let G be a 2-connected plane graph with outer face C_G. Then a path T is called a **Tutte path** if:

- every T-bridge has at most **three** attachments and
- any T-bridge containing an edge of C_G has exactly **two** attachments **(outer bridges)**.

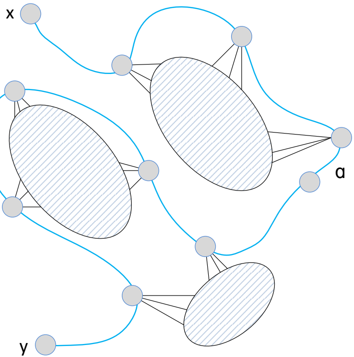
Tutte paths - Definition

Let G be a 2-connected plane graph with outer face C_G. Then a path T is called a **Tutte path** if:

- every T-bridge has at most **three** attachments and
- any T-bridge containing an edge of C_G has exactly **two** attachments **(outer bridges)**.

Given endvertices x and y and intermediate edge α, let $x - \alpha - y$ **path** denote a Tutte path from x to y through α.

Tutte Paths - Example



Tutte paths! What are they good for?

Tutte paths! What are they good for?

Often used to show Hamiltonicity of a graph class:

- 4-connected planar graphs [Tutte56].
- 4-connected projective planar graphs [Thomas94].
- 5-connected toroidal graphs [Thomas97].

Tutte paths! What are they good for?

Often used to show Hamiltonicity of a graph class:

- 4-connected planar graphs [Tutte56].
- 4-connected projective planar graphs [Thomas94].
- 5-connected toroidal graphs [Thomas97].

Or to make even stronger statements:

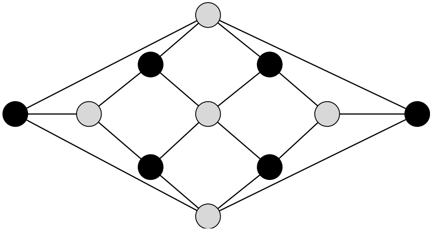
- 4-connected planar graphs are Hamiltonian-connected [Thomassen83].
- 4-connected planar graphs are 2-edge Hamiltonian-connected [Ozeki2014].
- 4-connected projective planar graphs are Hamiltonian-connected [Kawarabayaschi2015].

Using Tutte paths right!

Big open question: Which 3-connected planar graphs are Hamiltonian?

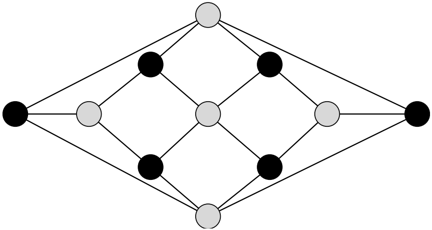
Using Tutte paths right!

Big open question: Which 3-connected planar graphs are Hamiltonian?



Using Tutte paths right!

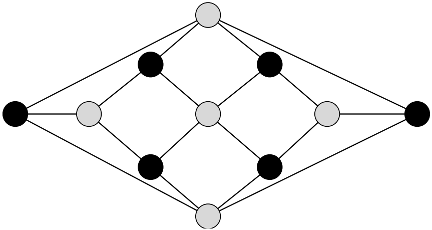
Big open question: Which 3-connected planar graphs are Hamiltonian?



- If G has no internal 3-separator [Thomassen83].

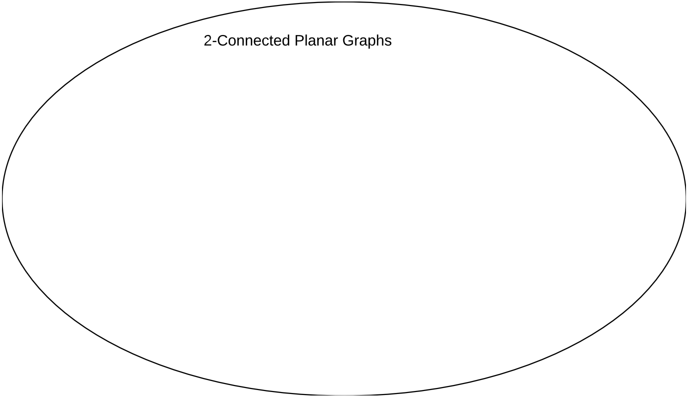
Using Tutte paths right!

Big open question: Which 3-connected planar graphs are Hamiltonian?

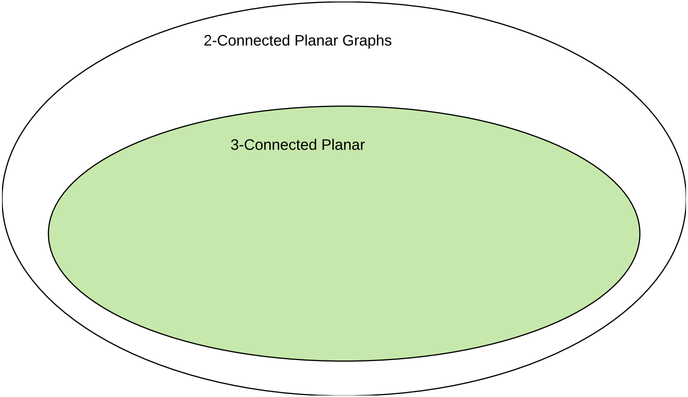


- If G has no internal 3-separator [Thomassen83].
- If G has at most three 3-separators [Brinkmann2016].

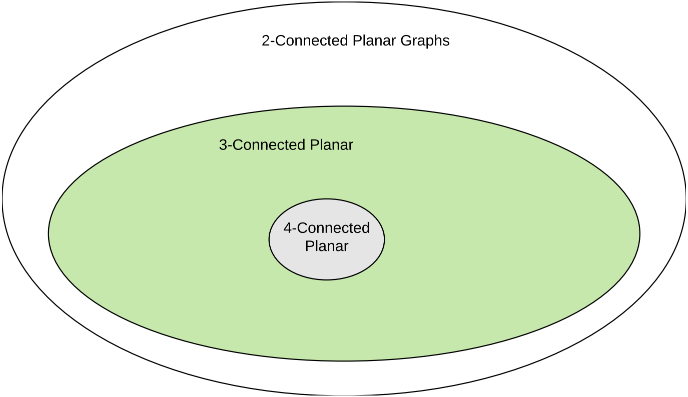
Using Tutte paths right!



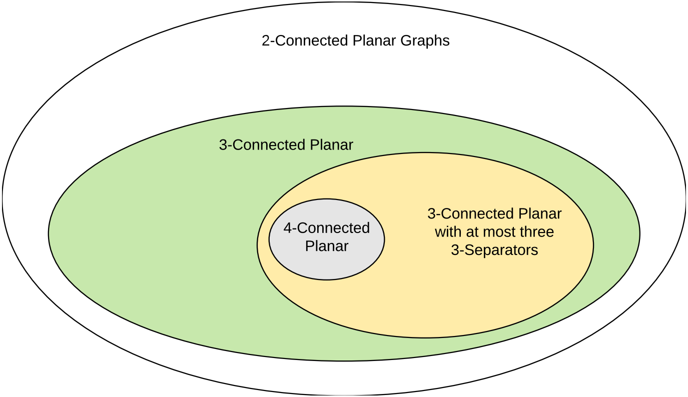
Using Tutte paths right!



Using Tutte paths right!



Using Tutte paths right!



Using Tutte paths right!

Also used to show relaxations of Hamiltonicity:

Using Tutte paths right!

Also used to show relaxations of Hamiltonicity:

- A cycle of length at least $\frac{5(n+2)}{8}$ in essentially 4-connected planar graphs [Fabrici18].

Using Tutte paths right!

Also used to show relaxations of Hamiltonicity:

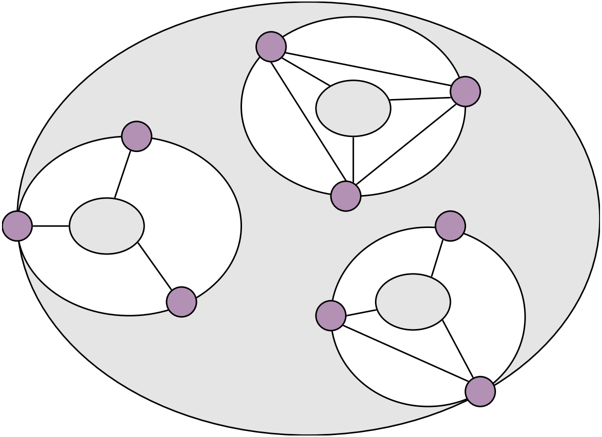
- A cycle of length at least $\frac{5(n+2)}{8}$ in essentially 4-connected planar graphs [Fabrici18].
- Every circuit graph contains a closed 2-walk [Gao95].

Using Tutte paths right!

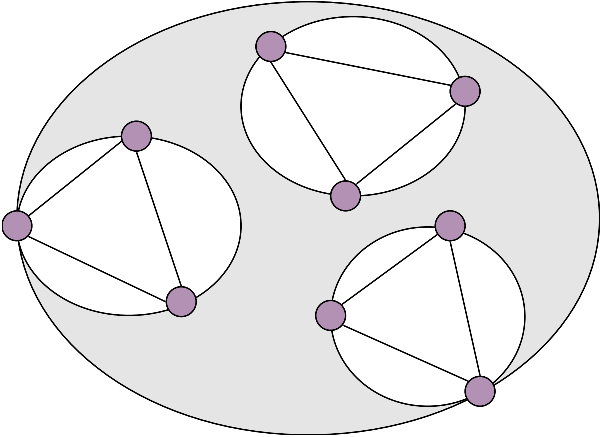
Also used to show relaxations of Hamiltonicity:

- A cycle of length at least $\frac{5(n+2)}{8}$ in essentially 4-connected planar graphs [Fabrici18].
- Every circuit graph contains a closed 2-walk [Gao95].
- Circuit graphs always have a 3-tree with few degree three vertices [Nakamoto2009].

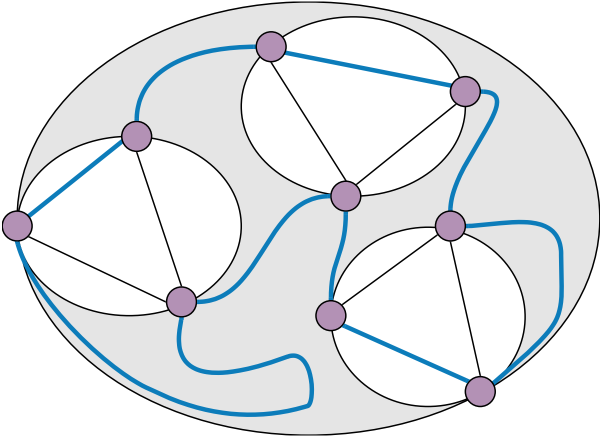
Graphs with at most three 3-separators



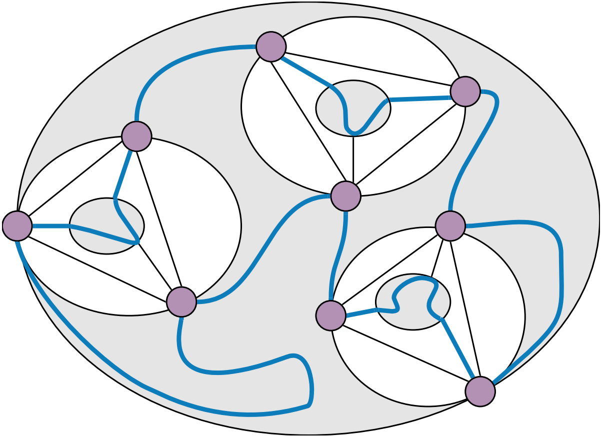
Graphs with at most three 3-separators



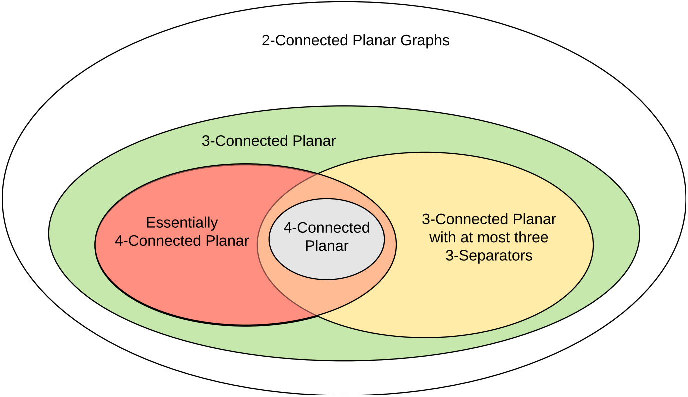
Graphs with at most three 3-separators



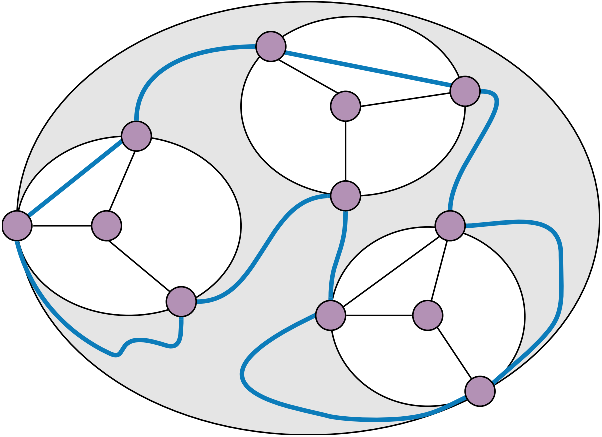
Graphs with at most three 3-separators



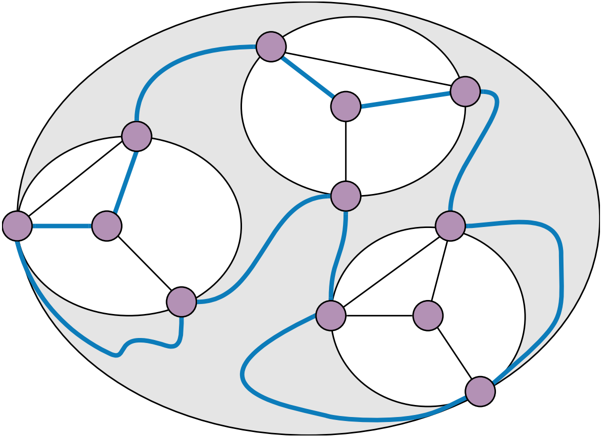
Essentially 4-connected planar graphs



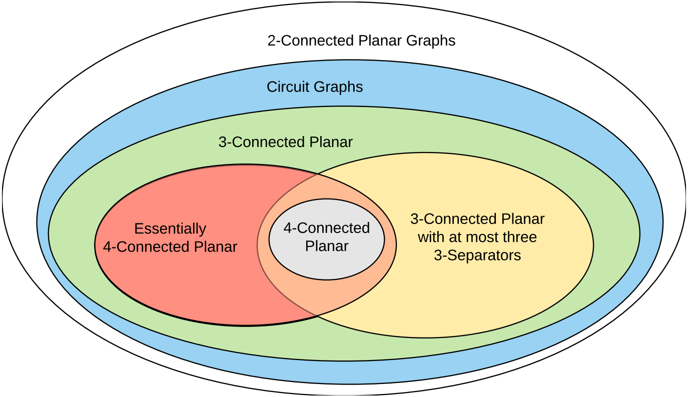
Essentially 4-connected planar graphs



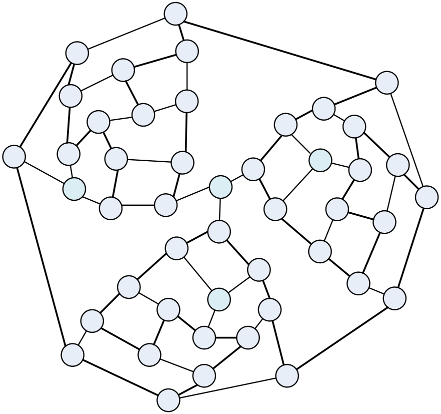
Essentially 4-connected planar graphs



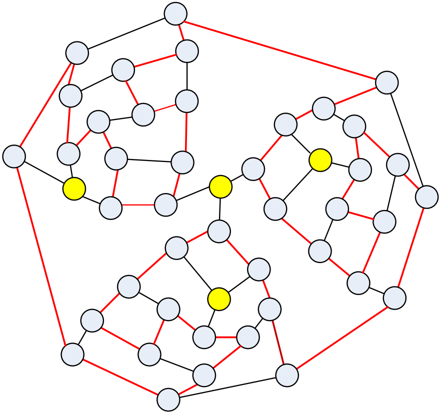
Closed 2-Walks in Circuit Graphs



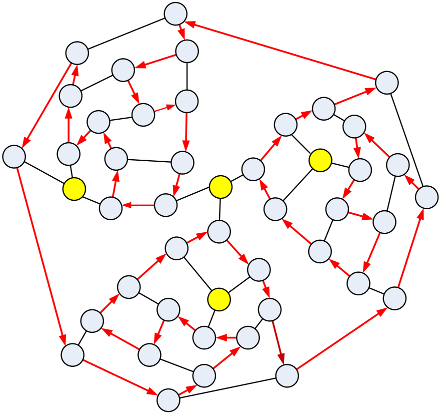
Closed 2-Walks in Circuit Graphs



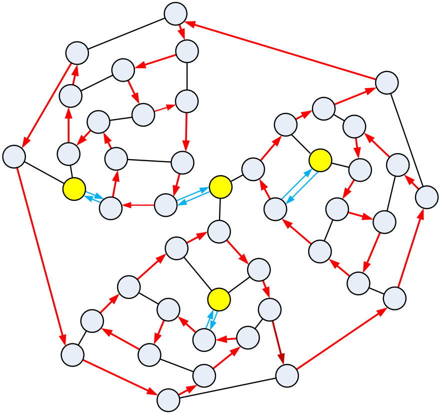
Closed 2-Walks in Circuit Graphs



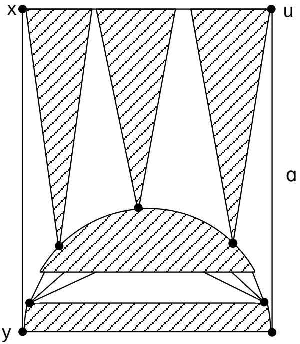
Closed 2-Walks in Circuit Graphs



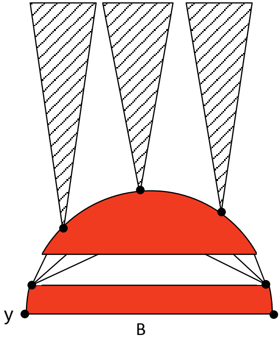
Closed 2-Walks in Circuit Graphs



Algorithmic Difficulty in their Proofs

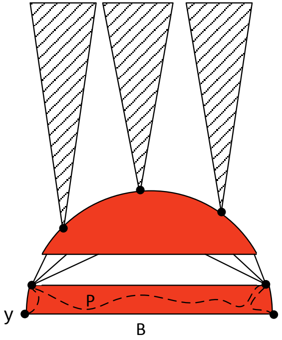


Algorithmic Difficulty in their Proofs



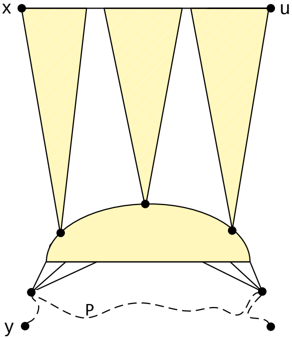
- Decompose the graph into blocks.

Algorithmic Difficulty in their Proofs



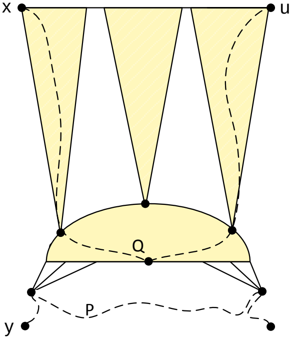
- Decompose the graph into blocks.
- Recurse on block B to compute P.

Algorithmic Difficulty in their Proofs



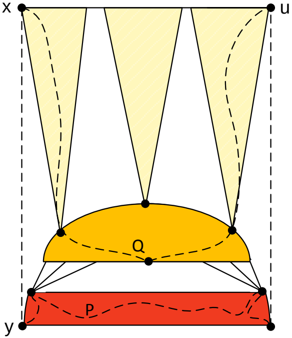
- Decompose the graph into blocks.
- Recurse on block B to compute P.
- Construct subgraph F.

Algorithmic Difficulty in their Proofs



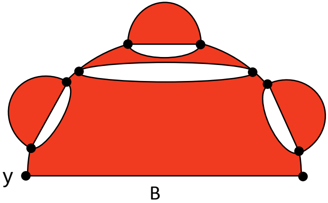
- Decompose the graph into blocks.
- Recurse on block B to compute P.
- Construct subgraph F.
- Recurse on F to compute Q.

Algorithmic Difficulty in their Proofs



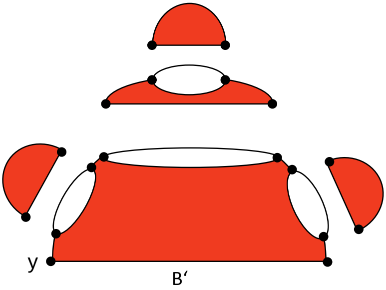
- Decompose the graph into blocks.
- Recurse on block *B* to compute *P*.
- Construct subgraph *F*.
- Recurse on *F* to compute *Q*.

Our Algorithm



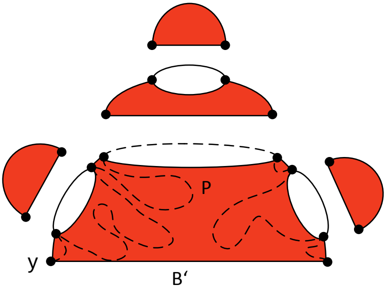
- Decompose the graph into blocks.

Our Algorithm



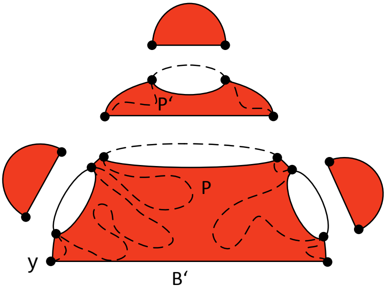
- Decompose the graph into blocks.
- We modify a block B to B'. depending on its 2-separators

Our Algorithm



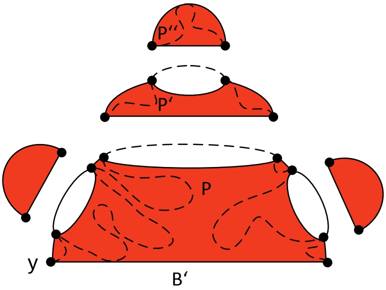
- Decompose the graph into blocks.
- We modify a block B to B'. depending on its 2-separators
- Recurse on B' to compute P.

Our Algorithm



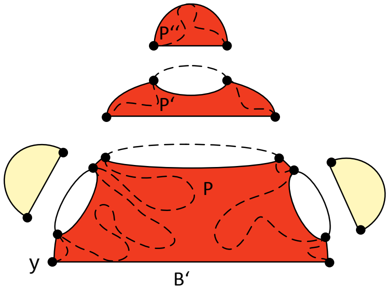
- Decompose the graph into blocks.
- We modify a block B to B'. depending on its 2-separators
- Recurse on B' to compute P.
- If P contains a virtual edge: recurse on split off graph.

Our Algorithm



- Decompose the graph into blocks.
- We modify a block B to B'. depending on its 2-separators
- Recurse on B' to compute P.
- If P contains a virtual edge: recurse on split off graph.

Our Algorithm



- Decompose the graph into blocks.
- We modify a block B to B'. depending on its 2-separators
- Recurse on B' to compute P.
- If P contains a virtual edge: recurse on split off graph.
- Merge the resulting path with paths from other blocks.

Running Time

- Decomposing G into blocks is in $O(n)$.

Running Time

- Decomposing G into blocks is in $O(n)$.
- All critical 2-separators for the incremental computation of P can be found in $O(n)$.

Running Time

- Decomposing G into blocks is in $O(n)$.
- All critical 2-separators for the incremental computation of P can be found in $O(n)$.
- Our Algorithm does at most $O(n)$ recursions.

Running Time

- Decomposing G into blocks is in $O(n)$.
- All critical 2-separators for the incremental computation of P can be found in $O(n)$.
- Our Algorithm does at most $O(n)$ recursions.

Therefore:

Theorem

Let G be a 2-connected plane graph and let $x, y \in V(G)$ and $\alpha \in E(G)$. A $x - \alpha - y$ path in G can be found in $O(n^2)$ time.

Tutte Paths in Linear Time

In ICALP'19, Biedl and Kindermann gave the first linear time algorithm

- Based on decomposing the graph into 3-connected subgraphs.

Tutte Paths in Linear Time

In ICALP'19, Biedl and Kindermann gave the first linear time algorithm

- Based on decomposing the graph into 3-connected subgraphs.
- Both end-vertices have to lie on the outer face.
- The authors say this cannot be generalized for arbitrary end-vertex position.

Future Work

- Find more applications...
- Improve the running time
  - Can our algorithm be improved to $O(n)$ running time?

Future Work

- Find more applications...
- Improve the running time
  - Can our algorithm be improved to $O(n)$ running time?
- Find an algorithm for other variants of Tutte paths (for example in circuit graphs [Jackson2002]).

Future Work

- Find more applications...
- Improve the running time
  - Can our algorithm be improved to $O(n)$ running time?
- Find an algorithm for other variants of Tutte paths (for example in circuit graphs [Jackson2002]).

Thank You!
Questions?