Network Sparsification for Steiner Problems on Planar and Bounded-Genus Graphs

Marcin Pilipczuk Michał Pilipczuk Piotr Sankowski
Erik Jan van Leeuwen

30th July 2013

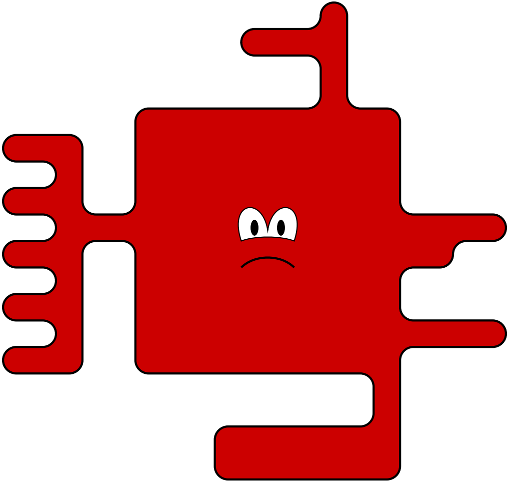
Outline

1. Introduction and background
    - Parameterized complexity and kernelization
    - Planar graphs and bidimensionality
2. Our contribution
    - Our results
    - Our techniques

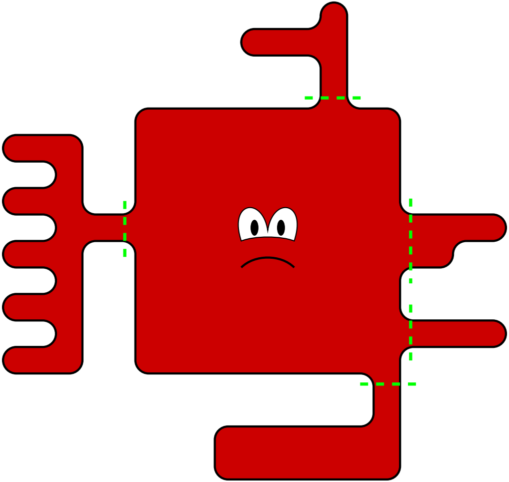
Outline

1. Introduction and background
   - Parameterized complexity and kernelization
   - Planar graphs and bidimensionality
2. Our contribution
   - Our results
   - Our techniques

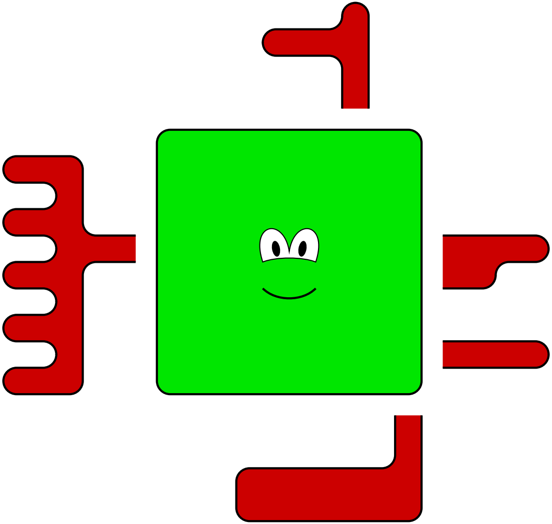
Preprocessing



Preprocessing

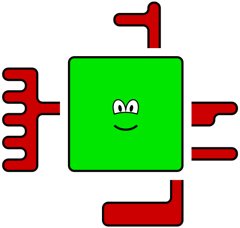


Preprocessing



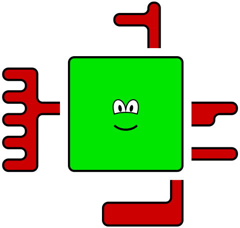
Preprocessing theoretically

- Very useful in practice.



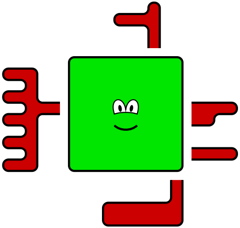
Preprocessing theoretically

- Very useful in practice.
- Not obvious how to analyze from theoretical point of view.



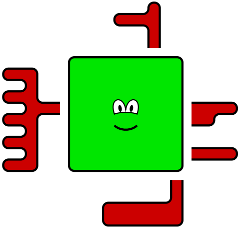
Preprocessing theoretically

- Very useful in practice.
- Not obvious how to analyze from theoretical point of view.
- *Fast* algorithm ⇒ *Polynomial-time* algorithm.

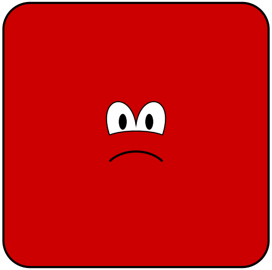


Preprocessing theoretically

- Very useful in practice.
- Not obvious how to analyze from theoretical point of view.
- *Fast* algorithm $\Rightarrow$ *Polynomial-time* algorithm.
- How to measure how well the instance is preprocessed?



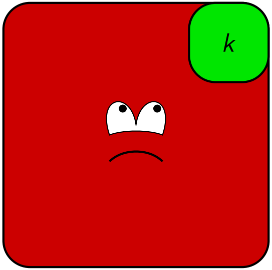
Parameterized complexity



instance of NP-hard problem

- Multidimensional analysis.

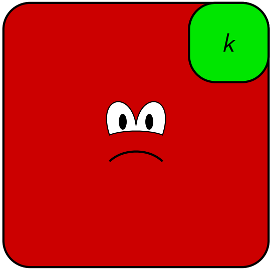
Parameterized complexity



instance of NP-hard problem

- Multidimensional analysis.
- Specify parameter.

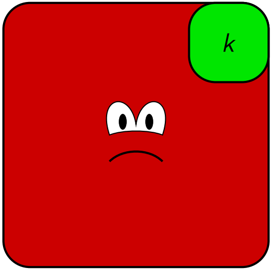
Parameterized complexity



instance of NP-hard problem

- Multidimensional analysis.
- Specify parameter.
- Instead of VERTEX COVER, we have

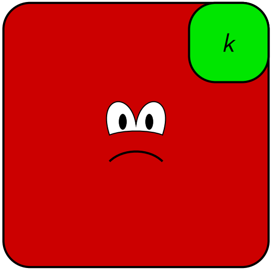
Parameterized complexity



instance of NP-hard problem

- Multidimensional analysis.
- Specify parameter.
- Instead of VERTEX COVER, we have
- VERTEX COVER / SOL. SIZE

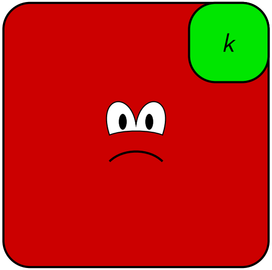
Parameterized complexity



instance of NP-hard problem

- Multidimensional analysis.
- Specify parameter.
- Instead of VERTEX COVER, we have
- VERTEX COVER / SOL. SIZE
- VERTEX COVER / TREEWIDTH

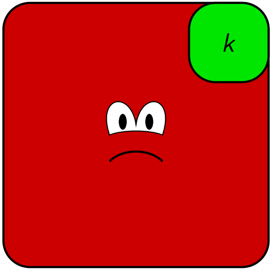
Parameterized complexity



instance of NP-hard problem

- Multidimensional analysis.
- Specify parameter.
- Instead of Vertex Cover, we have
- Vertex Cover / sol. size
- Vertex Cover / treewidth
- Vertex Cover / edition distance to cluster graphs

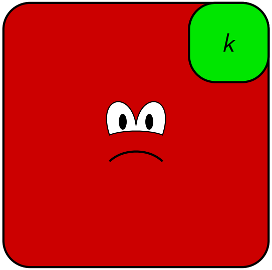
Parameterized complexity



instance of NP-hard problem

- Multidimensional analysis.
- Specify parameter.
- Instead of VERTEX COVER, we have
- VERTEX COVER / SOL. SIZE
- VERTEX COVER / TREEWIDTH
- VERTEX COVER / EDITION DISTANCE TO CLUSTER GRAPHS
- VERTEX COVER / PATHWIDTH AND MAXIMUM DEGREE

Parameterized complexity

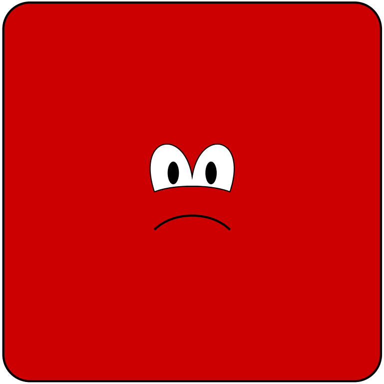


instance of NP-hard problem

- Multidimensional analysis.
- Specify parameter.
- Instead of VERTEX COVER, we have
- VERTEX COVER / SOL. SIZE
- VERTEX COVER / TREEWIDTH
- VERTEX COVER / EDITION DISTANCE TO CLUSTER GRAPHS
- VERTEX COVER / PATHWIDTH AND MAXIMUM DEGREE

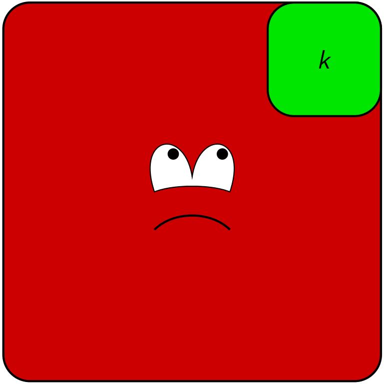
Goal: Do something clever when the parameter is small.

Kernelization



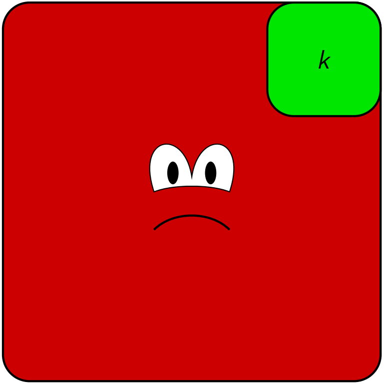
instance of NP-hard problem

Kernelization



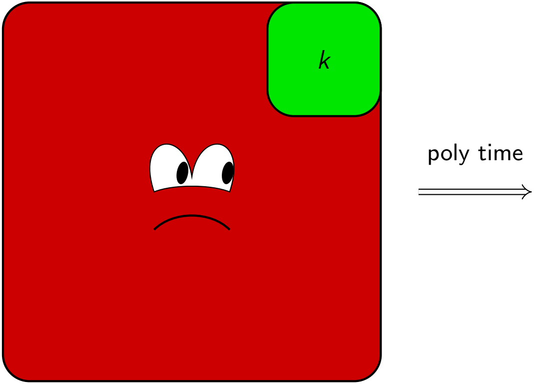
instance of NP-hard problem

Kernelization



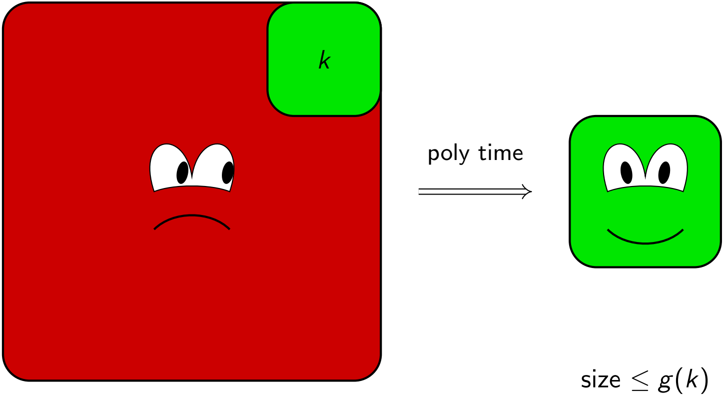
instance of NP-hard problem

Kernelization



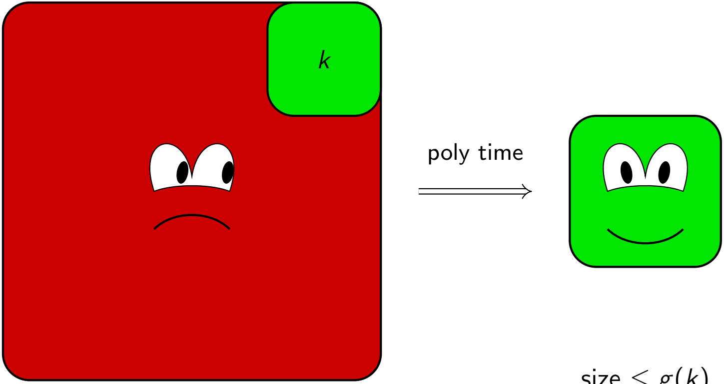
instance of NP-hard problem

Kernelization



instance of NP-hard problem

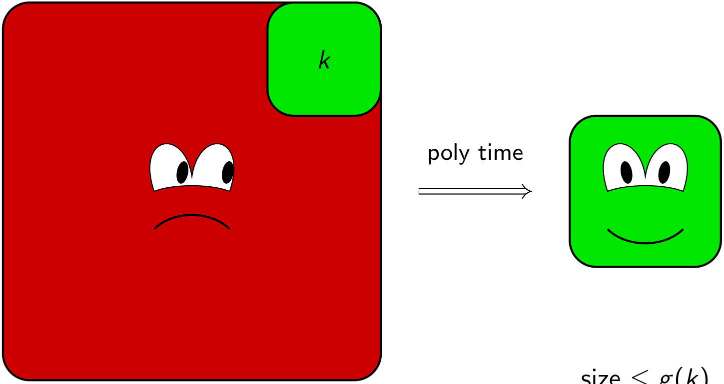
Kernelization



instance of NP-hard problem

size $\leq g(k)$
small: $g(k)$ polynomial

Kernelization



instance of NP-hard problem

size $\leq g(k)$

small: $g(k)$ polynomial

or even linear

Outline

Planar graphs

- Planar graphs have cool algorithmic properties.

Planar graphs

- Planar graphs have cool algorithmic properties.
- Many tools are reincarnations of Lipton-Tarjan separation theorem.

Planar graphs

- Planar graphs have cool algorithmic properties.
- Many tools are reincarnations of Lipton-Tarjan separation theorem.

Theorem (Grid minor theorem for planar graphs)

A planar graph without a $k \times k$ grid as a minor has treewidth $\mathcal{O}(k)$.

Planar graphs

- Planar graphs have cool algorithmic properties.
- Many tools are reincarnations of Lipton-Tarjan separation theorem.

Theorem (Grid minor theorem for planar graphs)

A planar graph without a $k \times k$ grid as a minor has treewidth $\mathcal{O}(k)$.

- Basic algorithmic usage:

Planar graphs

- Planar graphs have cool algorithmic properties.
- Many tools are reincarnations of Lipton-Tarjan separation theorem.

Theorem (Grid minor theorem for planar graphs)

A planar graph without a $k \times k$ grid as a minor has treewidth $\mathcal{O}(k)$.

- Basic algorithmic usage:
  - VERTEX COVER $< k$

Planar graphs

- Planar graphs have cool algorithmic properties.
- Many tools are reincarnations of Lipton-Tarjan separation theorem.

Theorem (Grid minor theorem for planar graphs)

A planar graph without a $k \times k$ grid as a minor has treewidth $\mathcal{O}(k)$.

- Basic algorithmic usage:
  - Vertex Cover $< k$
  - $\Rightarrow$ no $\sqrt{2k} \times \sqrt{2k}$ grid as a minor

Planar graphs

- Planar graphs have cool algorithmic properties.
- Many tools are reincarnations of Lipton-Tarjan separation theorem.

Theorem (Grid minor theorem for planar graphs)

A planar graph without a $k \times k$ grid as a minor has treewidth $\mathcal{O}(k)$.

- Basic algorithmic usage:
  - Vertex Cover $< k$
  - $\Rightarrow$ no $\sqrt{2k} \times \sqrt{2k}$ grid as a minor
  - $\Rightarrow$ treewidth $\in \mathcal{O}(\sqrt{k})$

Planar graphs

- Planar graphs have cool algorithmic properties.
- Many tools are reincarnations of Lipton-Tarjan separation theorem.

Theorem (Grid minor theorem for planar graphs)

A planar graph without a $k \times k$ grid as a minor has treewidth $\mathcal{O}(k)$.

- Basic algorithmic usage:
  - Vertex Cover $< k$
  - $\Rightarrow$ no $\sqrt{2k} \times \sqrt{2k}$ grid as a minor
  - $\Rightarrow$ treewidth $\in \mathcal{O}(\sqrt{k})$
  - $\Rightarrow$ subexponential (in k) algorithm.

Bidimesionality framework

Bidimensionality framework

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - VERTEX COVER,

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - VERTEX COVER,
  - DOMINATING SET,

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - VERTEX COVER,
  - DOMINATING SET,
  - FEEDBACK VERTEX SET,

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - Vertex Cover,
  - Dominating Set,
  - Feedback Vertex Set,
  - Connected Vertex Cover, ...

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - Vertex Cover,
  - Dominating Set,
  - Feedback Vertex Set,
  - Connected Vertex Cover, . . .
- in sparse graph classes, such as

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - Vertex Cover,
  - Dominating Set,
  - Feedback Vertex Set,
  - Connected Vertex Cover, ...
- in sparse graph classes, such as
  - planar graphs,

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - Vertex Cover,
  - Dominating Set,
  - Feedback Vertex Set,
  - Connected Vertex Cover, . . .
- in sparse graph classes, such as
  - planar graphs,
  - bounded-genus graphs,

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - Vertex Cover,
  - Dominating Set,
  - Feedback Vertex Set,
  - Connected Vertex Cover, ...
- in sparse graph classes, such as
  - planar graphs,
  - bounded-genus graphs,
  - graphs with a fixed excluded minor,

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - Vertex Cover,
  - Dominating Set,
  - Feedback Vertex Set,
  - Connected Vertex Cover, ...
- in sparse graph classes, such as
  - planar graphs,
  - bounded-genus graphs,
  - graphs with a fixed excluded minor,
  - excluded topological minor, bounded expansion, ...

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - Vertex Cover,
  - Dominating Set,
  - Feedback Vertex Set,
  - Connected Vertex Cover, ...
- in sparse graph classes, such as
  - planar graphs,
  - bounded-genus graphs,
  - graphs with a fixed excluded minor,
  - excluded topological minor, bounded expansion, ...
- and provides

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - Vertex Cover,
  - Dominating Set,
  - Feedback Vertex Set,
  - Connected Vertex Cover, ...
- in sparse graph classes, such as
  - planar graphs,
  - bounded-genus graphs,
  - graphs with a fixed excluded minor,
  - excluded topological minor, bounded expansion, ...
- and provides
  - subexponential fixed-parameter algorithms,

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - Vertex Cover,
  - Dominating Set,
  - Feedback Vertex Set,
  - Connected Vertex Cover, ...
- in sparse graph classes, such as
  - planar graphs,
  - bounded-genus graphs,
  - graphs with a fixed excluded minor,
  - excluded topological minor, bounded expansion, ...
- and provides
  - subexponential fixed-parameter algorithms,
  - linear kernels,

Bidimesionality framework

Bidimensionality framework

- applies to a variety of problems, e.g.,
  - VERTEX COVER,
  - DOMINATING SET,
  - FEEDBACK VERTEX SET,
  - CONNECTED VERTEX COVER, ...
- in sparse graph classes, such as
  - planar graphs,
  - bounded-genus graphs,
  - graphs with a fixed excluded minor,
  - excluded topological minor, bounded expansion, ...
- and provides
  - subexponential fixed-parameter algorithms,
  - linear kernels,
  - EPTASes.

Caveats

- Recent years: many efforts to generalize in terms of graph class:

Caveats

- Recent years: many efforts to generalize in terms of graph class:
  - fixed excluded minor, fixed excluded topological minor, bounded expansion, . . .

Caveats

- Recent years: many efforts to generalize in terms of graph class:
  - fixed excluded minor, fixed excluded topological minor, bounded expansion, . . .
- But they are important problems not covered at all.

Caveats

- Recent years: many efforts to generalize in terms of graph class:
  - fixed excluded minor, fixed excluded topological minor, bounded expansion, ...
- But they are important problems not covered at all.
  - recall: Vertex Cover $< k \Rightarrow$ no $\sqrt{2k} \times \sqrt{2k}$ grid as a minor;

Caveats

- Recent years: many efforts to generalize in terms of graph class:
  - fixed excluded minor, fixed excluded topological minor, bounded expansion, ...
- But they are important problems not covered at all.
  - recall: Vertex Cover $< k \Rightarrow$ no $\sqrt{2k} \times \sqrt{2k}$ grid as a minor;
  - fails for e.g. Steiner Tree / number of edges of the tree,

Caveats

- Recent years: many efforts to generalize in terms of graph class:
  - fixed excluded minor, fixed excluded topological minor, bounded expansion, ...
- But they are important problems not covered at all.
  - recall: VERTEX COVER $< k \Rightarrow$ no $\sqrt{2k} \times \sqrt{2k}$ grid as a minor;
  - fails for e.g. STEINER TREE / NUMBER OF EDGES OF THE TREE,
  - (and many problems in directed graphs).

Caveats

- Recent years: many efforts to generalize in terms of graph class:
  - fixed excluded minor, fixed excluded topological minor, bounded expansion, . . .
- But they are important problems not covered at all.
  - recall: Vertex Cover $< k \Rightarrow$ no $\sqrt{2k} \times \sqrt{2k}$ grid as a minor;
  - fails for e.g. Steiner Tree / number of edges of the tree,
  - (and many problems in directed graphs).
- Subexponential algorithms? Polynomial or linear kernels?

Outline

1. Introduction and background
   - Parameterized complexity and kernelization
   - Planar graphs and bidimensionality
2. Our contribution
   - Our results
   - Our techniques

Our results

Theorem (Polynomial kernel for Planar Steiner Tree)

STEINER TREE *in planar graphs, parameterized by the number of edges in the solution, has a kernel of size* $\mathcal{O}(k^{142})$.

Our results

Theorem (Polynomial kernel for Steiner Tree and Steiner Forest)

STEINER TREE *and* STEINER FOREST *in bounded-genus graphs, parameterized by the number of edges in the solution, have polynomial kernels.*

Our results

Theorem (Polynomial kernel for Steiner Tree and Steiner Forest)

STEINER TREE *and* STEINER FOREST *in bounded-genus graphs, parameterized by the number of edges in the solution, have polynomial kernels.*

Theorem (Polynomial kernel for Planar Edge Multiway Cut)

EDGE MULTIWAY CUT *in planar graphs, parameterized by the size of the solution, has a polynomial kernel.*

Our results

Theorem (Polynomial kernel for Steiner Tree and Steiner Forest)

STEINER TREE *and* STEINER FOREST *in bounded-genus graphs, parameterized by the number of edges in the solution, have polynomial kernels.*

Theorem (Polynomial kernel for Planar Edge Multiway Cut)

EDGE MULTIWAY CUT *in planar graphs, parameterized by the size of the solution, has a polynomial kernel.*

Corollary (Subexponential algorithms)

STEINER TREE *in bounded-genus graphs and* EDGE MULTIWAY CUT *in planar graphs admit subexponential algorithms.*

Outline

1. Introduction and background
   - Parameterized complexity and kernelization
   - Planar graphs and bidimensionality

2. Our contribution
   - Our results
   - Our techniques

Backbone theorem

Statement that stands behind:

Backbone theorem

Statement that stands behind:

Definition

A brick is a connected plane graph G with outer face surrounded by a simple cycle ∂G, called the perimeter of the brick.

Backbone theorem

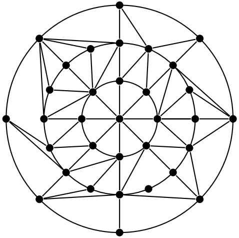
Statement that stands behind:

Definition

A brick is a connected plane graph G with outer face surrounded by a simple cycle ∂G, called the perimeter of the brick.

Theorem (Steiner Tree in bricks)

Given a brick G with perimeter ∂G of length k,



Backbone theorem

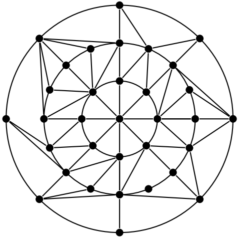
Statement that stands behind:

Definition

A brick is a connected plane graph G with outer face surrounded by a simple cycle ∂G, called the perimeter of the brick.

Theorem (Steiner Tree in bricks)

Given a brick G with perimeter ∂G of length k, one can in time $\mathcal{O}(k^{142}|G|)$



Backbone theorem

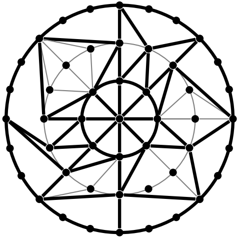
Statement that stands behind:

Definition

A brick is a connected plane graph G with outer face surrounded by a simple cycle ∂G, called the perimeter of the brick.

Theorem (Steiner Tree in bricks)

Given a brick G with perimeter ∂G of length k, one can in time $\mathcal{O}(k^{142}|G|)$ find **a set** *F of edges in G*



Backbone theorem

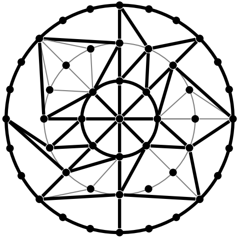
Statement that stands behind:

Definition

A brick is a connected plane graph G with outer face surrounded by a simple cycle ∂G, called the perimeter of the brick.

Theorem (Steiner Tree in bricks)

Given a brick G with perimeter ∂G of length k, one can in time $\mathcal{O}(k^{142}|G|)$ find **a set** *F of edges in G of size $\mathcal{O}(k^{142})$*



Backbone theorem

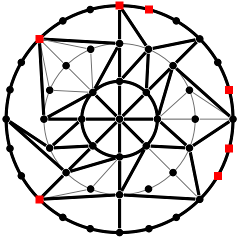
Statement that stands behind:

Definition

A brick is a connected plane graph G with outer face surrounded by a simple cycle ∂G, called the perimeter of the brick.

Theorem (Steiner Tree in bricks)

Given a brick G with perimeter ∂G of length k, one can in time $\mathcal{O}(k^{142}|G|)$ find **a set** *F of edges in G of size $\mathcal{O}(k^{142})$ such that for any* **set of terminals** *$S \subseteq \partial G$*



Backbone theorem

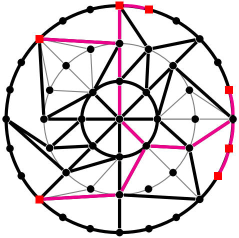
Statement that stands behind:

Definition

A brick is a connected plane graph G with outer face surrounded by a simple cycle ∂G, called the perimeter of the brick.

Theorem (Steiner Tree in bricks)

Given a brick G with perimeter ∂G of length k, one can in time $\mathcal{O}(k^{142}|G|)$ find **a set** *F of edges in G of size $\mathcal{O}(k^{142})$ such that for any* **set of terminals** *$S \subseteq \partial G$ there exists* **a Steiner tree connecting** *S*



Backbone theorem

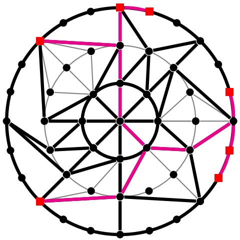
Statement that stands behind:

Definition

A brick is a connected plane graph G with outer face surrounded by a simple cycle ∂G, called the perimeter of the brick.

Theorem (Steiner Tree in bricks)

*Given a brick G with perimeter ∂G of length k, one can in time $\mathcal{O}(k^{142}|G|)$ find **a set** F of edges in G of size $\mathcal{O}(k^{142})$ such that for any **set of terminals** $S \subseteq \partial G$ there exists **a Steiner tree connecting** S that is contained in F*



Backbone theorem

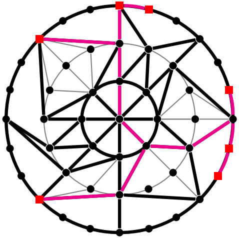
Statement that stands behind:

Definition

A brick is a connected plane graph G with outer face surrounded by a simple cycle ∂G, called the perimeter of the brick.

Theorem (Steiner Tree in bricks)

Given a brick G with perimeter ∂G of length k,
one can in time $\mathcal{O}(k^{142}|G|)$
find **a set** *F of edges in G of size $\mathcal{O}(k^{142})$*
such that for any **set of terminals** $S \subseteq \partial G$
there exists **a Steiner tree connecting** S
that is contained in F and optimal in the entire G.

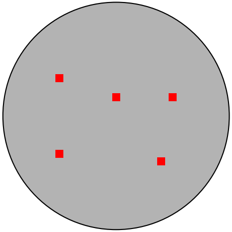


Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool!

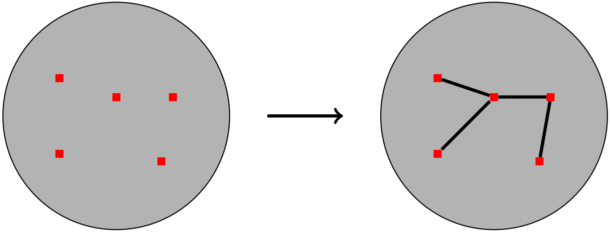
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! More robust!



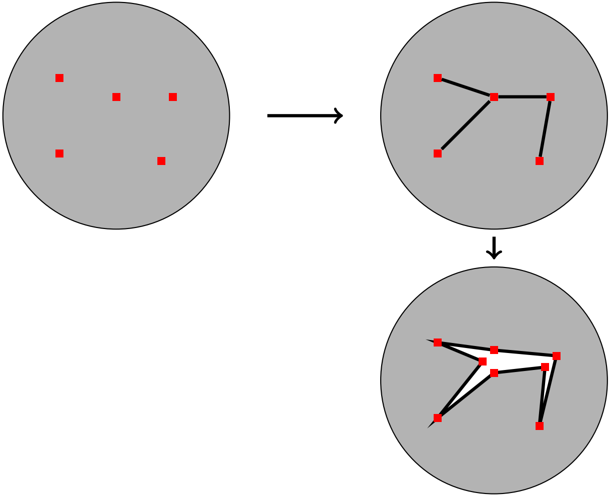
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! More robust!



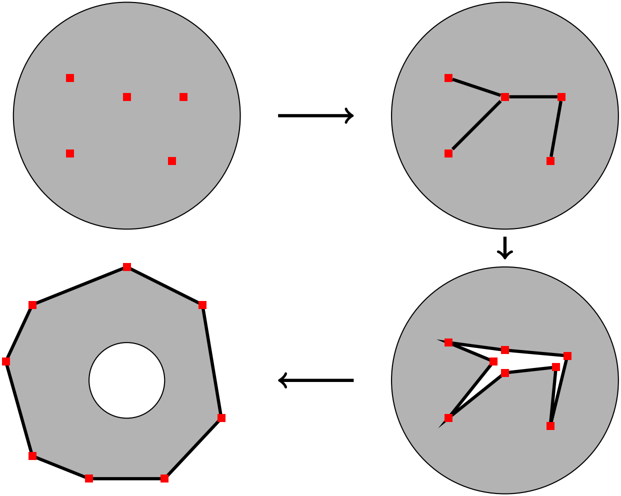
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! More robust!



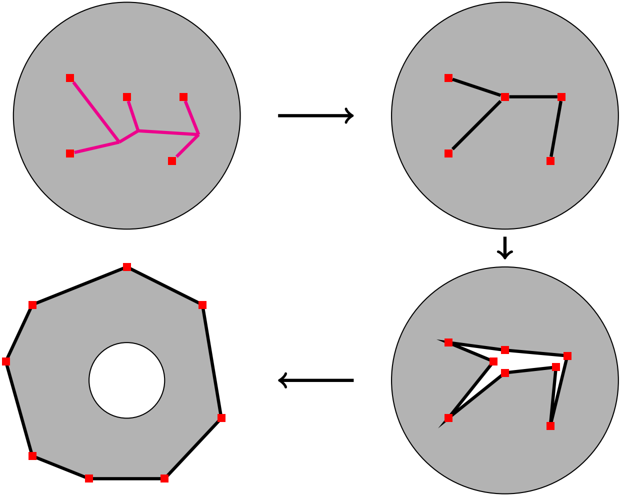
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! More robust!



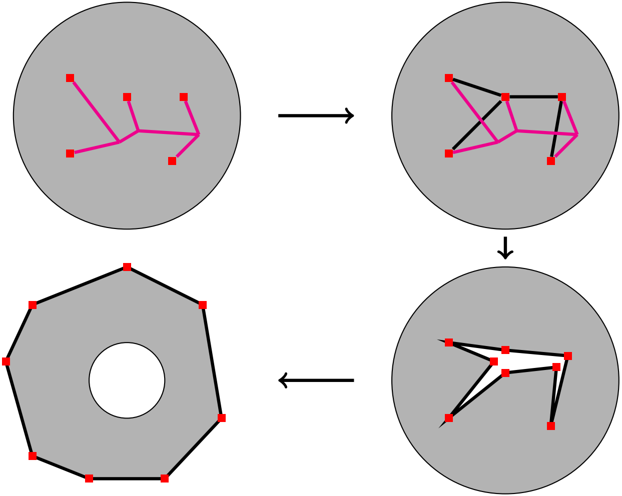
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! More robust!



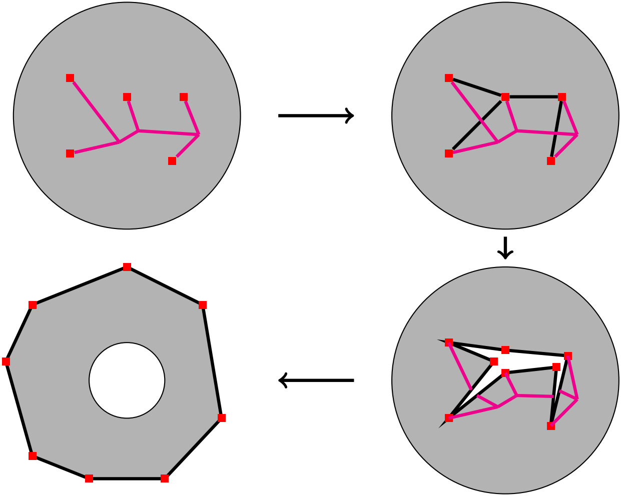
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! More robust!



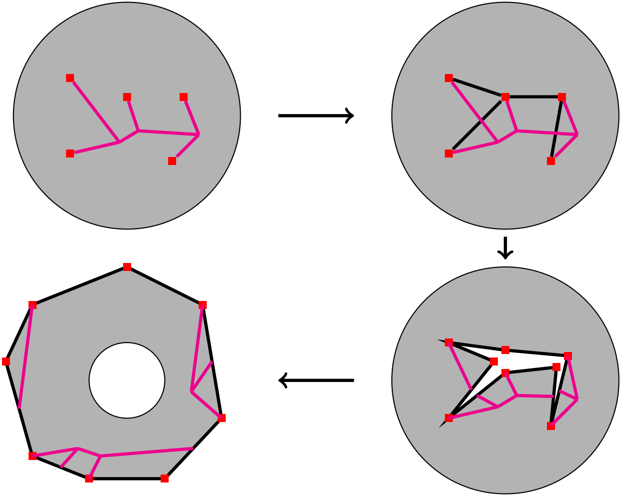
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! More robust!



Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! More robust!

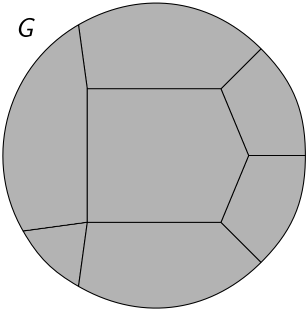


Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool!

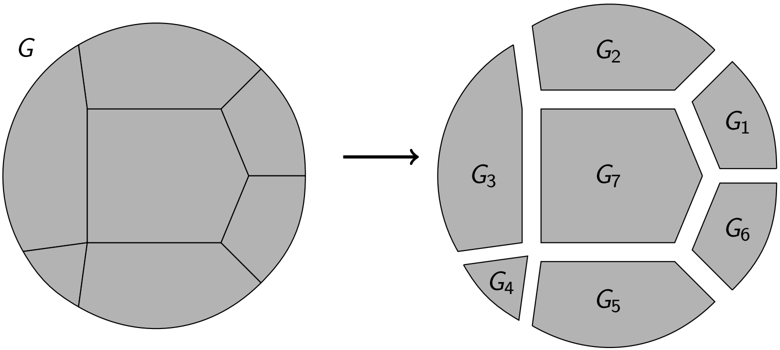
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! Divide and conquer!



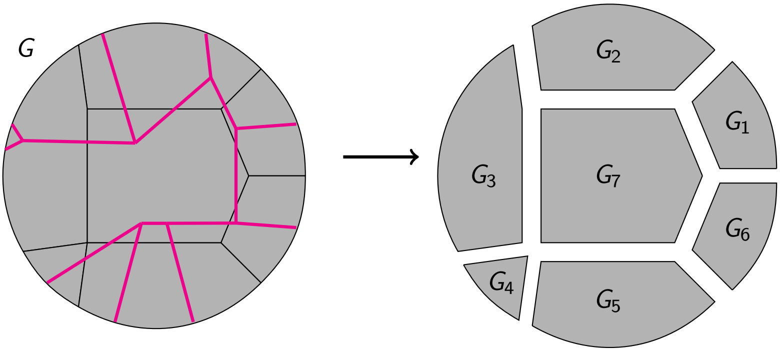
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! Divide and conquer!



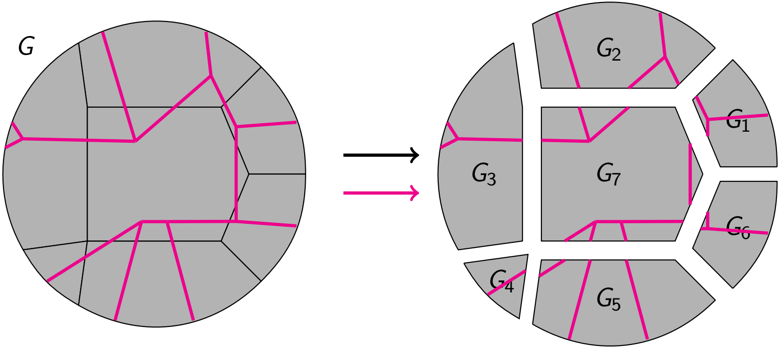
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! Divide and conquer!



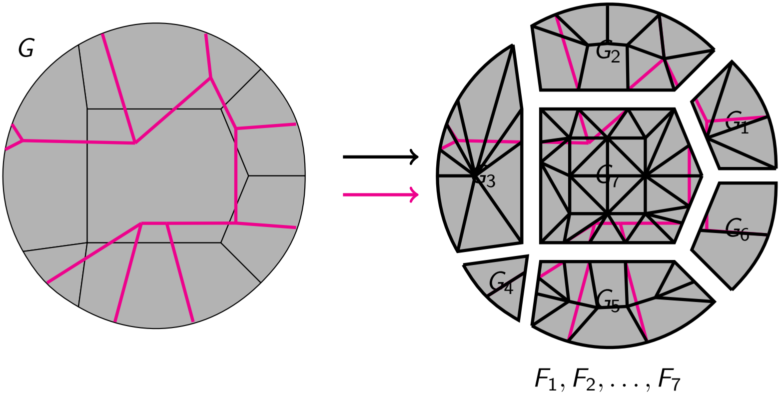
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! Divide and conquer!



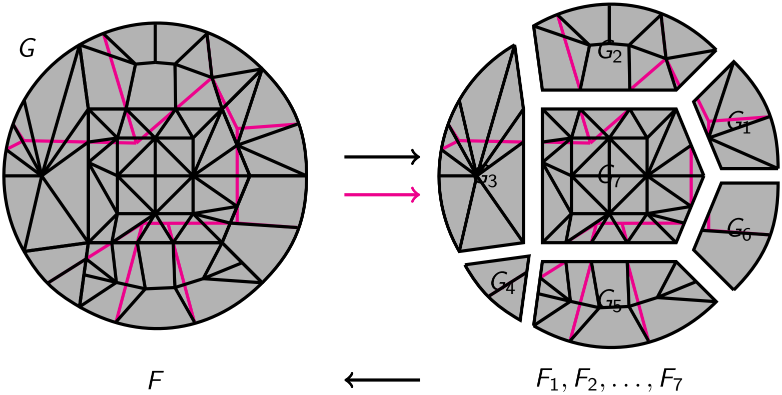
Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! Divide and conquer!

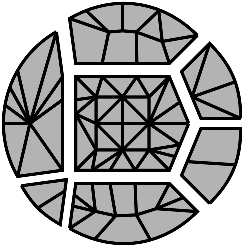


Bricks

[Borradaile, Klein, Mathieu, 2009] Bricks are cool! Divide and conquer!

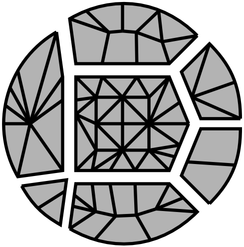


Polynomial bound



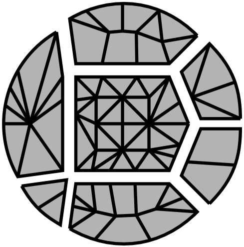
Define: $F := \bigcup_i F_i$

Polynomial bound



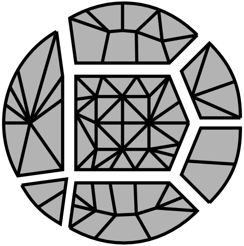
Define: $F := \bigcup_i F_i$
Want: $|F| \leq \text{poly}(|\partial G|)$

Polynomial bound



Define: $F := \bigcup_i F_i$
Want: $|F| \leq \mathrm{poly}(|\partial G|)$
Idea: $|\partial G|$ is an excellent potential

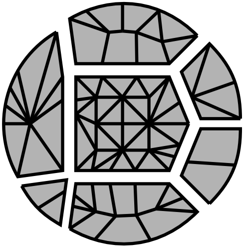
Polynomial bound



Define: $F := \bigcup_i F_i$
Want: $|F| \leq \mathrm{poly}(|\partial G|)$
Idea: $|\partial G|$ is an excellent potential

Need:

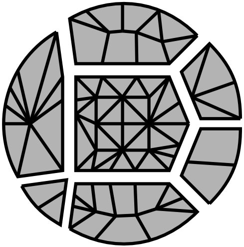
Polynomial bound



Define: $F := \bigcup_i F_i$
Want: $|F| \leq \mathrm{poly}(|\partial G|)$
Idea: $|\partial G|$ is an excellent potential

Need:
(i) $\sum_i |\partial G_i| \leq C|\partial G|$ for some C,

Polynomial bound



Define: $F := \bigcup_i F_i$
Want: $|F| \le \mathrm{poly}(|\partial G|)$
Idea: $|\partial G|$ is an excellent potential

Need:
(i) $\sum_i |\partial G_i| \le C|\partial G|$ for some C,
(ii) $\forall_i |\partial G_i| \le (1-\varepsilon)|\partial G|$ for some $\varepsilon > 0$.

Steiner tree as a separator

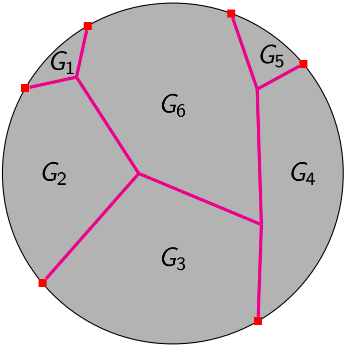
(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

Steiner tree as a separator

(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

Crucial observation:

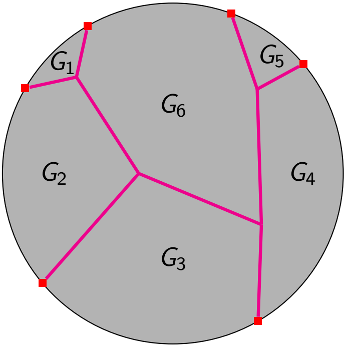
Steiner tree as a separator



(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

Crucial observation:
An optimal Steiner tree is usually a good separator!

Steiner tree as a separator

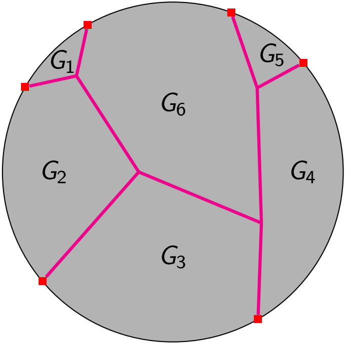


(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

Crucial observation:
An optimal Steiner tree is usually a good separator!

Cond. (i) is for free with $C = 3$

Steiner tree as a separator



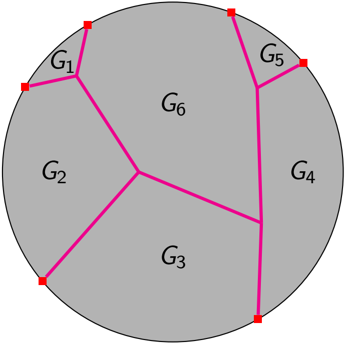
(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

Crucial observation:
An optimal Steiner tree is usually a good separator!

Cond. (i) is for free with $C = 3$

Proof:

Steiner tree as a separator



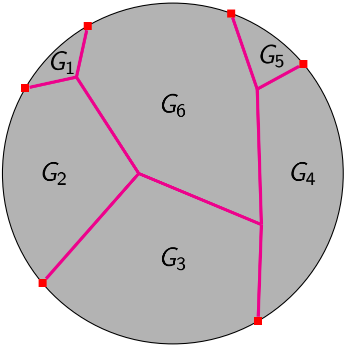
(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

Crucial observation:
An optimal Steiner tree is usually a good separator!

Cond. (i) is for free with $C = 3$

Proof:
∂G is an excellent Steiner tree

Steiner tree as a separator



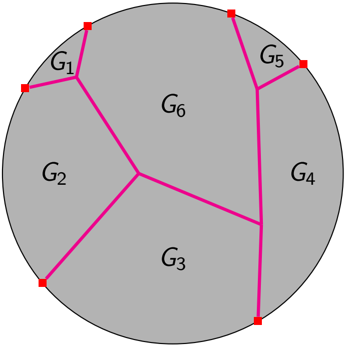
(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

Crucial observation:
An optimal Steiner tree is usually a good separator!

Cond. (i) is for free with $C = 3$

Proof:
∂G is an excellent Steiner tree
$\Rightarrow |T| \leq |\partial G|$

Steiner tree as a separator



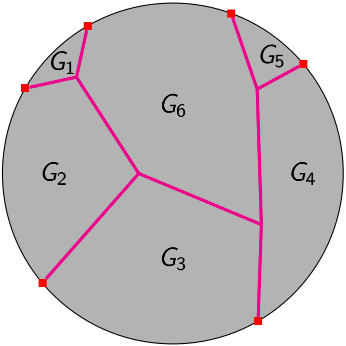
(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

Crucial observation:
An optimal Steiner tree is usually a good separator!

Cond. (i) is for free with $C = 3$

Proof:
∂G is an excellent Steiner tree
$\Rightarrow |T| \leq |\partial G|$
$\Rightarrow \sum_i |\partial G_i|$

Steiner tree as a separator



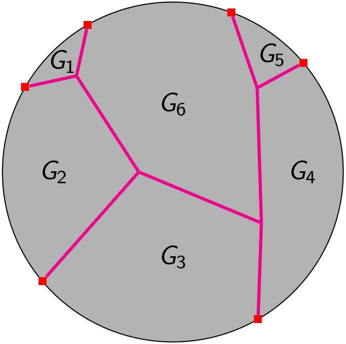
(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

Crucial observation:
An optimal Steiner tree is usually a good separator!

Cond. (i) is for free with $C = 3$

Proof:
∂G is an excellent Steiner tree
$\Rightarrow |T| \leq |\partial G|$
$\Rightarrow \sum_i |\partial G_i|$
$= |\partial G| + 2|T|$

Steiner tree as a separator



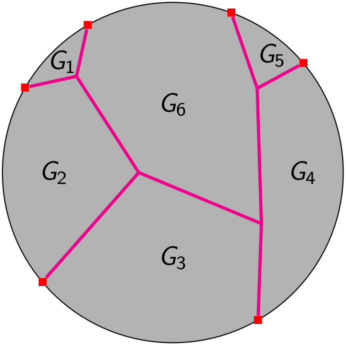
(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

Crucial observation:
An optimal Steiner tree is usually a good separator!

Cond. (i) is for free with $C = 3$

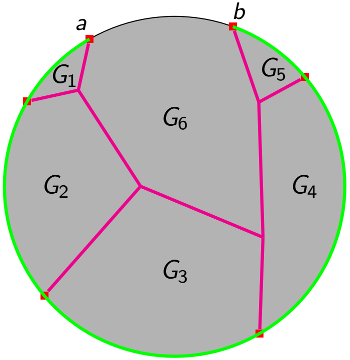
Proof:
∂G is an excellent Steiner tree
$\Rightarrow |T| \leq |\partial G|$
$\Rightarrow \sum_i |\partial G_i|$
$= |\partial G| + 2|T|$
$\leq 3|\partial G|$

Steiner tree as a separator



(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

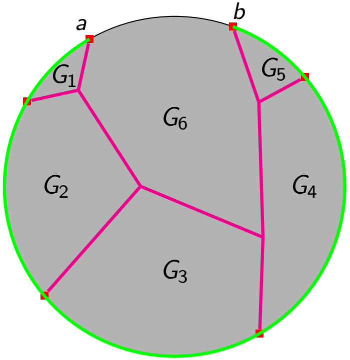
Steiner tree as a separator



(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

$\partial G[a, b]$ is an excellent tree

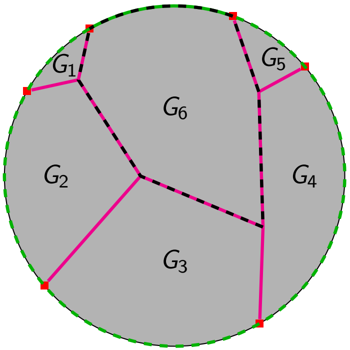
Steiner tree as a separator



(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

$\partial G[a, b]$ is an excellent tree
$\Rightarrow |T| \leq |\partial G[a, b]|$

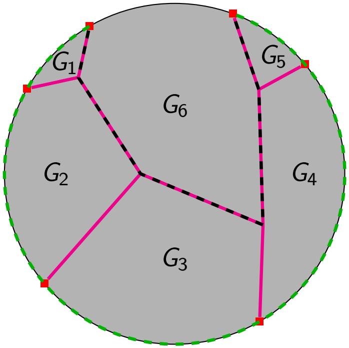
Steiner tree as a separator



(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

$\partial G[a,b]$ is an excellent tree
$\Rightarrow |T| \leq |\partial G[a,b]|$
$\Rightarrow |\partial G| - |\partial G_6|$

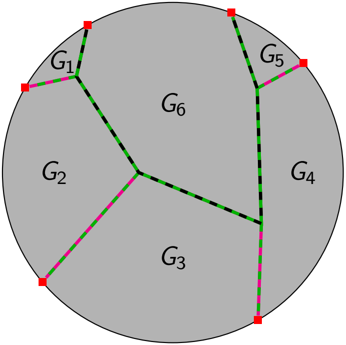
Steiner tree as a separator



(i) $\sum_i |\partial G_i| \le C|\partial G|$,
(ii) $\forall_i |\partial G_i| \le (1-\varepsilon)|\partial G|$.

$\partial G[a,b]$ is an excellent tree
$\Rightarrow |T| \le |\partial G[a,b]|$
$\Rightarrow |\partial G| - |\partial G_6|$
$= |\partial G[a,b]| - |T \cap \partial G_6|$

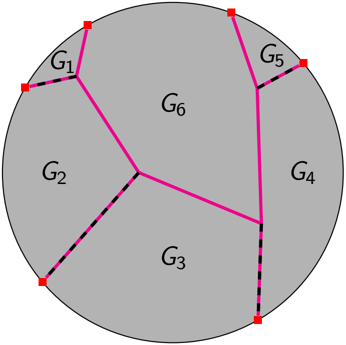
Steiner tree as a separator



(i) $\sum_i |\partial G_i| \leq C|\partial G|$,
(ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

$\partial G[a,b]$ is an excellent tree
$\Rightarrow |T| \leq |\partial G[a,b]|$
$\Rightarrow |\partial G| - |\partial G_6|$
$= |\partial G[a,b]| - |T \cap \partial G_6|$
$\geq |T| - |T \cap \partial G_6|$

Steiner tree as a separator



(i) $\sum_i |\partial G_i| \le C|\partial G|$,
(ii) $\forall_i |\partial G_i| \le (1-\varepsilon)|\partial G|$.

$\partial G[a,b]$ is an excellent tree

$$\Rightarrow |T| \le |\partial G[a,b]|$$

$$\Rightarrow |\partial G| - |\partial G_6|$$

$$= |\partial G[a,b]| - |T \cap \partial G_6|$$

$$\ge |T| - |T \cap \partial G_6|$$

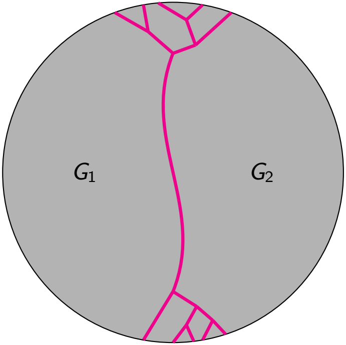
$$= |T \setminus \partial G_6|$$

Bad cases

Bad cases for (ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.

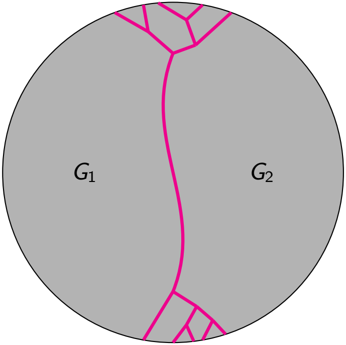
Bad cases

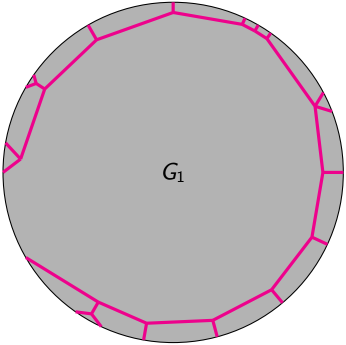
Bad cases for (ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.



Bad cases

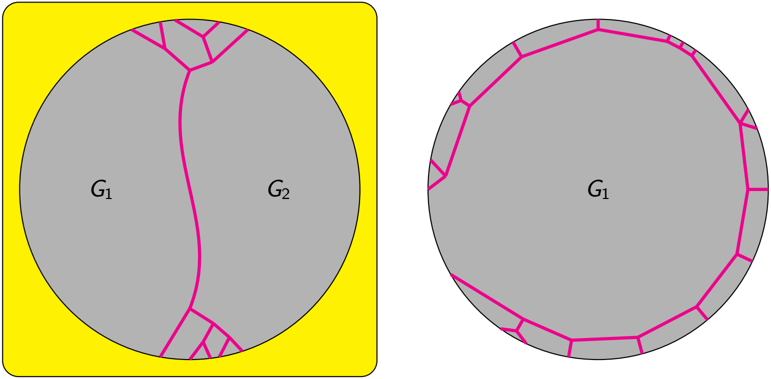
Bad cases for (ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.



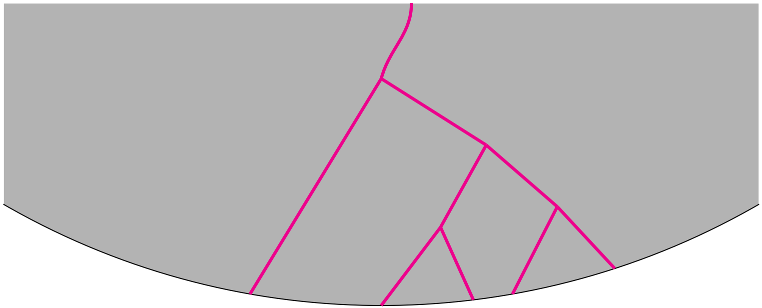


Bad cases

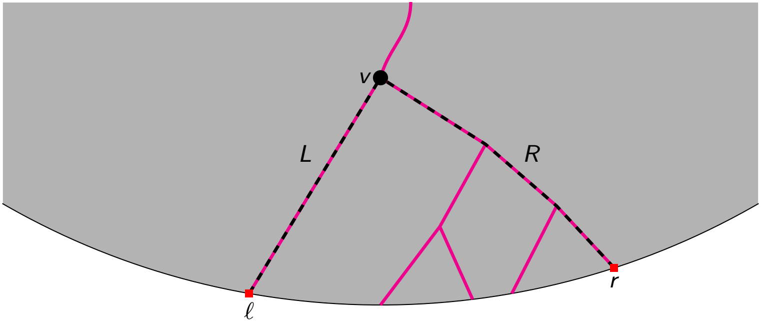
Bad cases for (ii) $\forall_i |\partial G_i| \leq (1-\varepsilon)|\partial G|$.



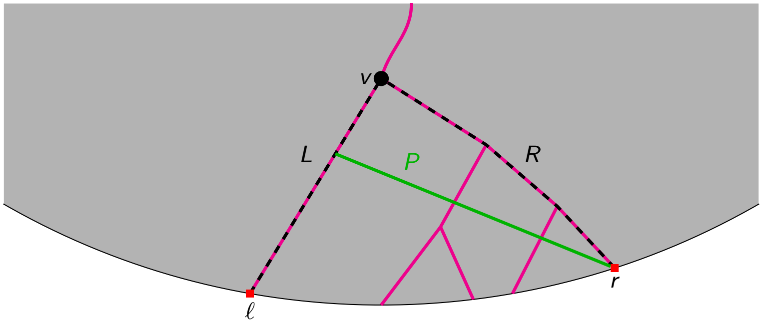
Mountains



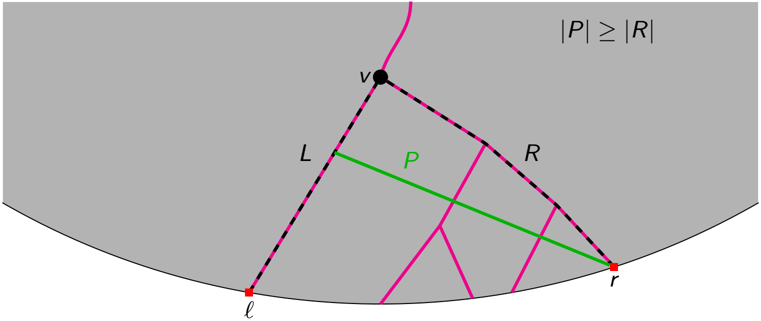
Mountains



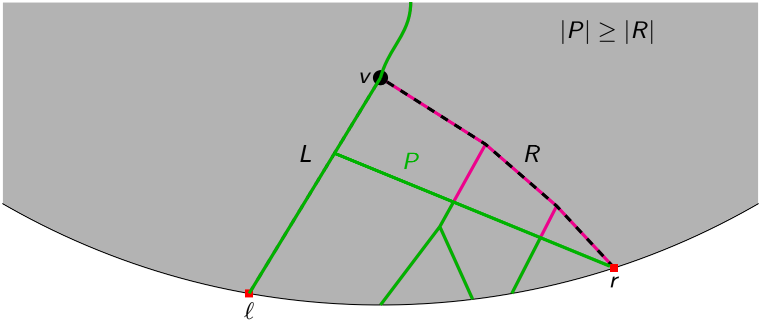
Mountains



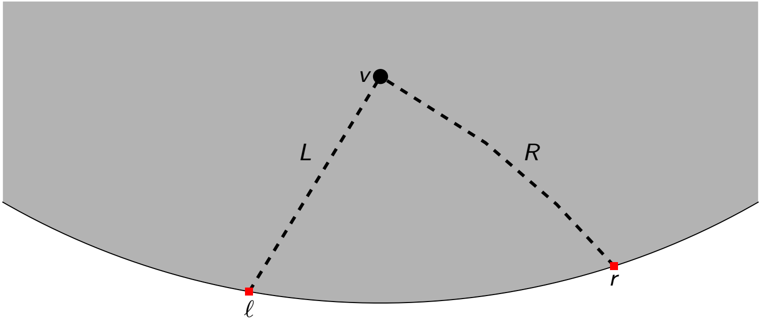
Mountains



Mountains



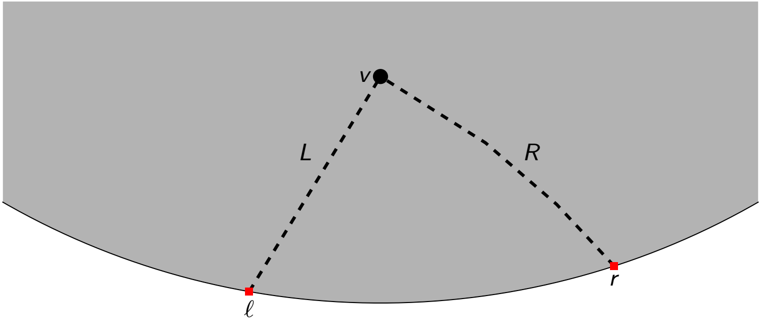
Mountains



Definition (Mountain)

$M = (L, R)$ is a mountain if

Mountains

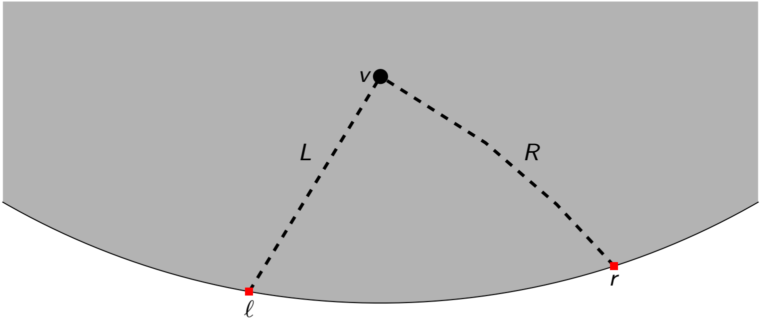


Definition (Mountain)

$M = (L, R)$ is a mountain if

(i) L is a shortest $\ell - R$ path inside M, and

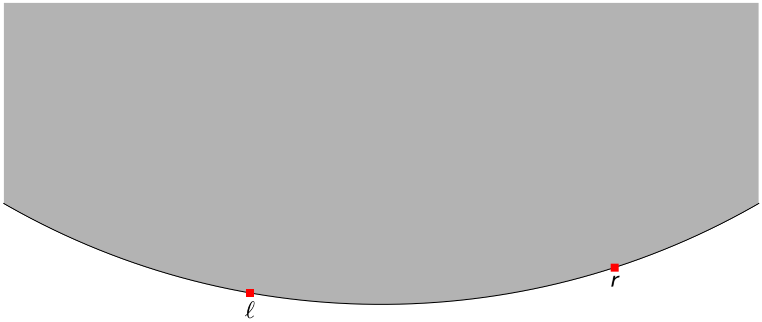
Mountains



Definition (Mountain)

$M = (L, R)$ is a mountain if
(i) L is a shortest $\ell - R$ path inside M, and
(ii) R is a shortest $r - L$ path inside M.

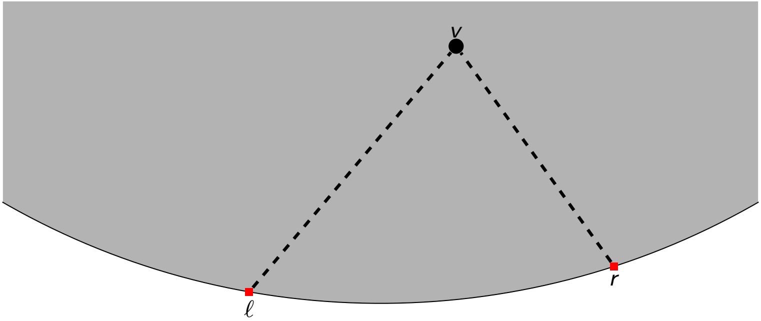
Mountain range theorem



Theorem (Mountain range theorem)

For fixed $\delta < 1/2$ and endpoints ℓ and r,

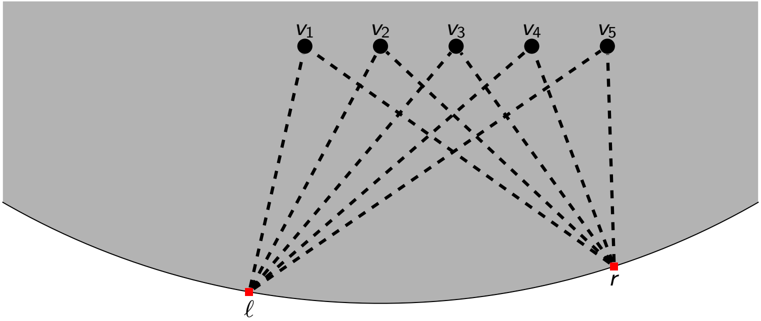
Mountain range theorem



Theorem (Mountain range theorem)

For fixed $\delta < 1/2$ and endpoints ℓ and r,
all maximal mountains of length at most $\delta|\partial G|$,

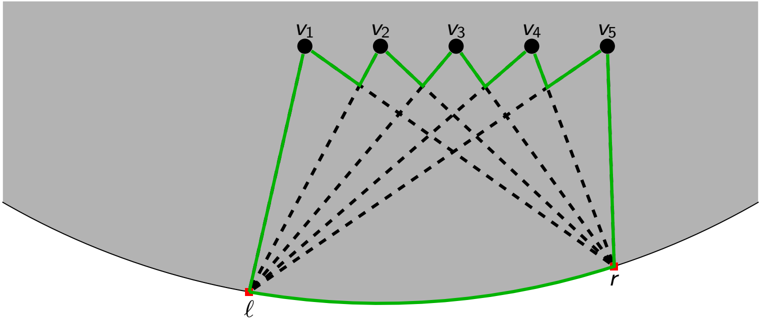
Mountain range theorem



Theorem (Mountain range theorem)

For fixed $\delta < 1/2$ and endpoints ℓ and r,
all maximal mountains of length at most $\delta|\partial G|$,
look like on the figure

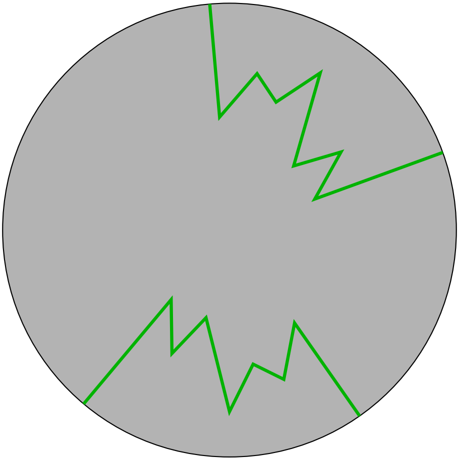
Mountain range theorem



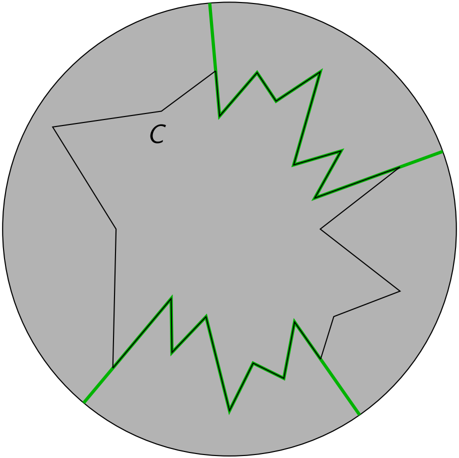
Theorem (Mountain range theorem)

For fixed $\delta < 1/2$ and endpoints ℓ and r,
all maximal mountains of length at most $\delta|\partial G|$,
look like on the figure
and have total perimeter at most $3|\partial G[\ell, r]|$.

All mountain ranges

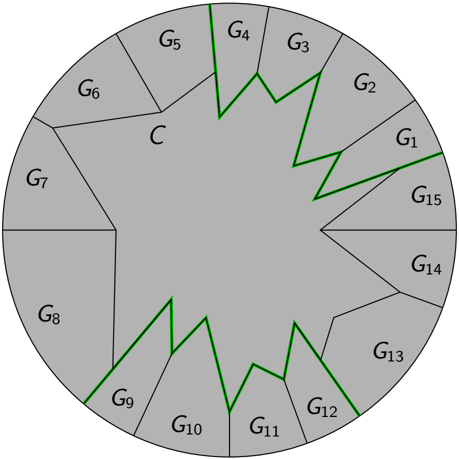


All mountain ranges



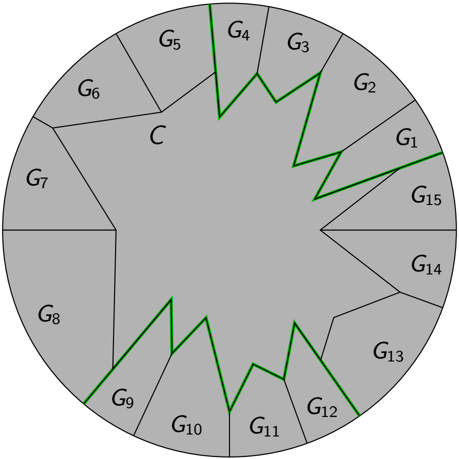
- bound on the perimeter of a mountain range $\Rightarrow |C| = \mathcal{O}(|\partial G|)$

All mountain ranges



- bound on the perimeter of a mountain range $\Rightarrow |C| = \mathcal{O}(|\partial G|)$

All mountain ranges

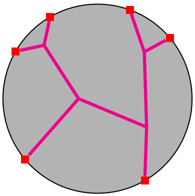


- bound on the perimeter of a mountain range $\Rightarrow |C| = \mathcal{O}(|\partial G|)$
- inside C mark only shortest paths between points on the perimeter.

Recap

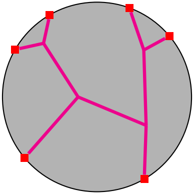
- If $|\partial G| = \mathcal{O}(1)$, do brute-force.

Recap



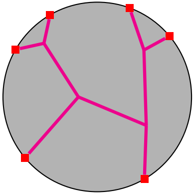
- If $|\partial G| = \mathcal{O}(1)$, do brute-force.
- If there exists an optimal Steiner tree T that is a good separator, split with T and recurse.

Recap



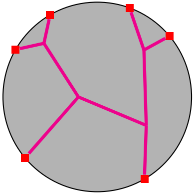
- If $|\partial G| = \mathcal{O}(1)$, do brute-force.
- If there exists an optimal Steiner tree T that is a good separator, split with T and recurse.
  - Finding such T is a technical, but natural DP.

Recap

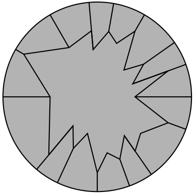


- If $|\partial G| = \mathcal{O}(1)$, do brute-force.
- If there exists an optimal Steiner tree T that is a good separator, split with T and recurse.
  - Finding such T is a technical, but natural DP.
  - Fix $\varepsilon = 1/36$.

Recap

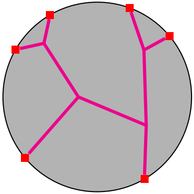


- If $|\partial G| = \mathcal{O}(1)$, do brute-force.
- If there exists an optimal Steiner tree T that is a good separator, split with T and recurse.
  - Finding such T is a technical, but natural DP.
  - Fix $\varepsilon = 1/36$.

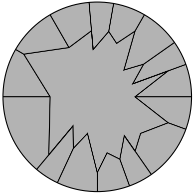


- Otherwise, compute the union of $\mathcal{O}(1/\varepsilon^2)$ mountain ranges and the cycle C. Perform decomposition of the second type and recurse.

Recap

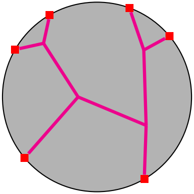


- If $|\partial G| = \mathcal{O}(1)$, do brute-force.
- If there exists an optimal Steiner tree T that is a good separator, split with T and recurse.
  - Finding such T is a technical, but natural DP.
  - Fix $\varepsilon = 1/36$.

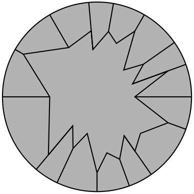


- Otherwise, compute the union of $\mathcal{O}(1/\varepsilon^2)$ mountain ranges and the cycle C. Perform decomposition of the second type and recurse.
  - Some technical massage to cope with the second bad case.

Recap



- If $|\partial G| = \mathcal{O}(1)$, do brute-force.
- If there exists an optimal Steiner tree T that is a good separator, split with T and recurse.
  - Finding such T is a technical, but natural DP.
  - Fix $\varepsilon = 1/36$.



- Otherwise, compute the union of $\mathcal{O}(1/\varepsilon^2)$ mountain ranges and the cycle C. Perform decomposition of the second type and recurse.
  - Some technical massage to cope with the second bad case.

- Big exponent due to small ε and large C in the decomposition of the second type.

Open problems

Some open problems:

Open problems

Some open problems:

- Get better exponent!

Open problems

Some open problems:

- Get better exponent!
  - For the brick theorem, a grid yields an $\Omega(k^2)$ lower bound.

Open problems

Some open problems:

- Get better exponent!
  - For the brick theorem, a grid yields an $\Omega(k^2)$ lower bound.
- What about parameter *number of terminals*?

Open problems

Some open problems:

- Get better exponent!
  - For the brick theorem, a grid yields an $\Omega(k^2)$ lower bound.
- What about parameter *number of terminals*?
- What about problems with vertex-based measures, such as NODE MULTIWAY CUT?

Open problems

Some open problems:

- Get better exponent!
  - For the brick theorem, a grid yields an $\Omega(k^2)$ lower bound.
- What about parameter *number of terminals*?
- What about problems with vertex-based measures, such as NODE MULTIWAY CUT?
  - Also a combinatorial kernel for PLANAR ODD CYCLE TRANSVERSAL would be nice.

Open problems

Some open problems:

- Get better exponent!
  - For the brick theorem, a grid yields an $\Omega(k^2)$ lower bound.
- What about parameter *number of terminals*?
- What about problems with vertex-based measures, such as NODE MULTIWAY CUT?
  - Also a combinatorial kernel for PLANAR ODD CYCLE TRANSVERSAL would be nice.
- Lift to graphs with fixed excluded minor?

Open problems

Some open problems:

- Get better exponent!
  - For the brick theorem, a grid yields an $\Omega(k^2)$ lower bound.
- What about parameter *number of terminals*?
- What about problems with vertex-based measures, such as NODE MULTIWAY CUT?
  - Also a combinatorial kernel for PLANAR ODD CYCLE TRANSVERSAL would be nice.
- Lift to graphs with fixed excluded minor?
  - EDGE MULTIWAY CUT for e.g. torus is also open.

Open problems

Some open problems:

- Get better exponent!
  - For the brick theorem, a grid yields an $\Omega(k^2)$ lower bound.
- What about parameter *number of terminals*?
- What about problems with vertex-based measures, such as NODE MULTIWAY CUT?
  - Also a combinatorial kernel for PLANAR ODD CYCLE TRANSVERSAL would be nice.
- Lift to graphs with fixed excluded minor?
  - EDGE MULTIWAY CUT for e.g. torus is also open.
- Subexponential algorithm for PLANAR STEINER FOREST?

Open problems

Some open problems:

- Get better exponent!
  - For the brick theorem, a grid yields an $\Omega(k^2)$ lower bound.
- What about parameter *number of terminals*?
- What about problems with vertex-based measures, such as Node Multiway Cut?
  - Also a combinatorial kernel for Planar Odd Cycle Transversal would be nice.
- Lift to graphs with fixed excluded minor?
  - Edge Multiway Cut for e.g. torus is also open.
- Subexponential algorithm for Planar Steiner Forest?
  - Main obstacle: NP-hard for treewidth 3.

Thank you

Questions?