Minimum Bisection is fixed-parameter tractable

Marek Cygan[1], Daniel Lokshtanov[2], Marcin Pilipczuk[2],
Michał Pilipczuk[2], Saket Saurabh[2,3]

[1] Institute of Informatics, University of Warsaw, Poland

[2] Department of Informatics, University of Bergen, Norway

[3] Institute of Mathematical Sciences, Chennai, India

Bertinoro Workshop on Graphs,
December 17$^{\text{th}}$, 2013

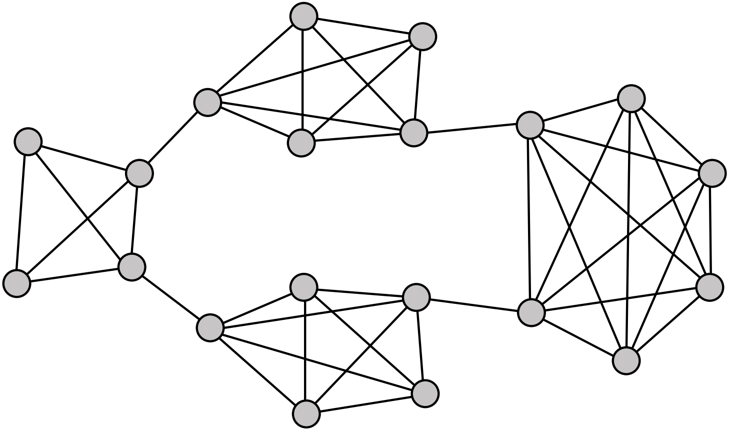
The problem

Mininimum Bisection

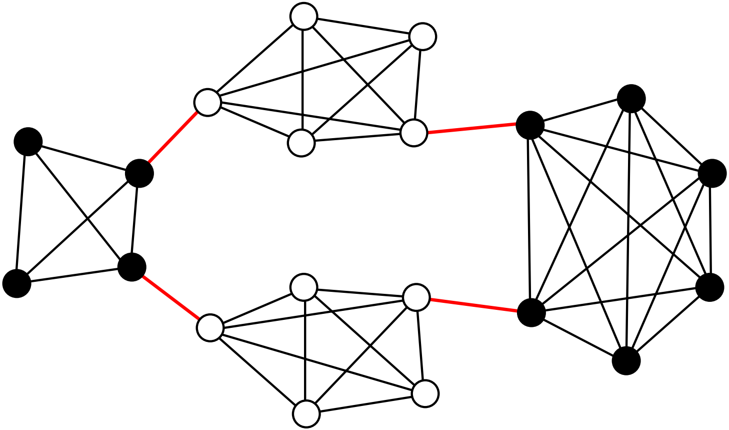
Input: A graph G with $|V(G)| = 2n$ and an integer k

Question: Does there exist a partition (A, B) of $V(G)$ such that $|A| = |B| = n$ and $|E(A, B)| \leq k$?

Example



Example



About the problem

- MinBisection is one of the most classic NP-hard problems,

About the problem

- MinBisection is one of the most classic NP-hard problems,
- but also one of the most mysterious ones.

About the problem

- MinBisection is one of the most classic NP-hard problems,
- but also one of the most mysterious ones.
  - Only $\mathcal{O}(\log n)$-approximation is known [Räcke], but a constant factor approximation is not refuted.

About the problem

- MinBisection is one of the most classic NP-hard problems,
- but also one of the most mysterious ones.
  - Only $\mathcal{O}(\log n)$-approximation is known [Räcke], but a constant factor approximation is not refuted.
  - It is open whether MinBisection is NP-hard on planar graphs.

About the problem

- MinBisection is one of the most classic NP-hard problems,
- but also one of the most mysterious ones.
  - Only $\mathcal{O}(\log n)$-approximation is known [Räcke], but a constant factor approximation is not refuted.
  - It is open whether MinBisection is NP-hard on planar graphs.
- **Our result**: MinBisection is fixed-parameter tractable when parameterized by k.

About the problem

- MinBisection is one of the most classic NP-hard problems,
- but also one of the most mysterious ones.
  - Only $\mathcal{O}(\log n)$-approximation is known [Räcke], but a constant factor approximation is not refuted.
  - It is open whether MinBisection is NP-hard on planar graphs.
- **Our result**: MinBisection is fixed-parameter tractable when parameterized by k.
- **Running time**: $2^{\mathcal{O}(k^3)} \cdot n^3 \log^3 n$

The approach

- **Decomposition**: a new decomposition theorem for graphs

The approach

- **Decomposition**: a new decomposition theorem for graphs
  - **Intuition**: A graph can be decomposed in a tree-like manner by separators of size roughly k into pieces that are highly connected in terms of k, and cannot be broken further.

The approach

- **Decomposition**: a new decomposition theorem for graphs
  - **Intuition**: A graph can be decomposed in a tree-like manner by separators of size roughly k into pieces that are highly connected in terms of k, and cannot be broken further.
  - More formally, each bag of the tree decomposition is (q, k)-unbreakable for $q = f(k)$.

The approach

- **Decomposition**: a new decomposition theorem for graphs
  - **Intuition**: A graph can be decomposed in a tree-like manner by separators of size roughly k into pieces that are highly connected in terms of k, and cannot be broken further.
  - More formally, each bag of the tree decomposition is (q, k)-unbreakable for $q = f(k)$.
- **Dynamic programming**: run a bottom-up DP on the obtained tree decomposition.

The approach

- **Decomposition**: a new decomposition theorem for graphs
  - **Intuition**: A graph can be decomposed in a tree-like manner by separators of size roughly k into pieces that are highly connected in terms of k, and cannot be broken further.
  - More formally, each bag of the tree decomposition is (q, k)-unbreakable for $q = f(k)$.
- **Dynamic programming**: run a bottom-up DP on the obtained tree decomposition.
  - MinBisection can be robustly solved in FPT time on (q, k)-unbreakable graphs using **randomized contractions**.

The approach

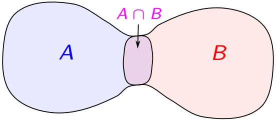
- **Decomposition**: a new decomposition theorem for graphs
  - **Intuition**: A graph can be decomposed in a tree-like manner by separators of size roughly k into pieces that are highly connected in terms of k, and cannot be broken further.
  - More formally, each bag of the tree decomposition is (q, k)-unbreakable for $q = f(k)$.
- **Dynamic programming**: run a bottom-up DP on the obtained tree decomposition.
  - MinBisection can be robustly solved in FPT time on (q, k)-unbreakable graphs using **randomized contractions**.
  - Use this algorithm as a subroutine at each bag of the decomposition.

The approach

- **Decomposition**: a new **decomposition theorem** for graphs
  - **Intuition**: A graph can be decomposed in a tree-like manner by separators of size roughly k into pieces that are highly connected in terms of k, and cannot be broken further.
  - More formally, each bag of the tree decomposition is (q, k)-unbreakable for $q = f(k)$.
- **Dynamic programming**: run a bottom-up DP on the obtained tree decomposition.
  - MinBisection can be robustly solved in FPT time on (q, k)-unbreakable graphs using **randomized contractions**.
  - Use this algorithm as a subroutine at each bag of the decomposition.

Unbreakable sets

- (A, B) is a *separation* if $A \cup B = V(G)$ and $E(A \setminus B, B \setminus A) = \emptyset$. $A \cap B$ is the *separator*, and $|A \cap B|$ is the *order* of the separation.

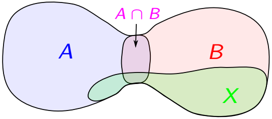


Unbreakable sets

- (A, B) is a *separation* if $A \cup B = V(G)$ and $E(A \setminus B, B \setminus A) = \emptyset$. $A \cap B$ is the *separator*, and $|A \cap B|$ is the *order* of the separation.

(q, k)-unbreakability

We say that a subset $X \subseteq V(G)$ is (q, k)-*unbreakable* if for every separation (A, B) of G of order at most k, either $|(A \setminus B) \cap X| \leq q$ or $|(B \setminus A) \cap X| \leq q$.



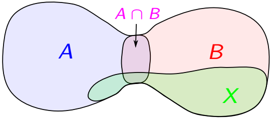
Unbreakable sets

- (A, B) is a *separation* if $A \cup B = V(G)$ and $E(A \setminus B, B \setminus A) = \emptyset$. $A \cap B$ is the *separator*, and $|A \cap B|$ is the *order* of the separation.

(q, k)-unbreakability

We say that a subset $X \subseteq V(G)$ is (q, k)-*unbreakable* if for every separation (A, B) of G of order at most k, either $|(A \setminus B) \cap X| \leq q$ or $|(B \setminus A) \cap X| \leq q$.

- **Intuition**: a separation of small order can carve out only a small portion of X.



The decomposition theorem

The decomposition theorem

Let G be a graph and k be an integer. Then there exists a tree decomposition $\mathcal{T} = (T, \{B_u\}_{u \in V(T)})$ of G such that:

The decomposition theorem

The decomposition theorem

Let G be a graph and k be an integer. Then there exists a tree decomposition $\mathcal{T} = (T, \{B_u\}_{u \in V(T)})$ of G such that:

- Each B_u is (τ, k)-unbreakable in G, where $\tau = 2^{\mathcal{O}(k)}$.

The decomposition theorem

The decomposition theorem

Let G be a graph and k be an integer. Then there exists a tree decomposition $\mathcal{T} = (T, \{B_u\}_{u \in V(T)})$ of G such that:

- Each B_u is (τ, k)-unbreakable in G, where $\tau = 2^{\mathcal{O}(k)}$.
- $|B_u \cap B_v| \leq \eta$ for each $uv \in E(T)$, where $\eta = 2^{\mathcal{O}(k)}$.

The decomposition theorem

The decomposition theorem

Let G be a graph and k be an integer. Then there exists a tree decomposition $\mathcal{T} = (T, \{B_u\}_{u \in V(T)})$ of G such that:

- Each B_u is (τ, k)-unbreakable in G, where $\tau = 2^{\mathcal{O}(k)}$.
- $|B_u \cap B_v| \leq \eta$ for each $uv \in E(T)$, where $\eta = 2^{\mathcal{O}(k)}$.
- $B_u \cap B_v$ is $(2k, k)$-unbreakable in G, for each $uv \in E(T)$.

The decomposition theorem

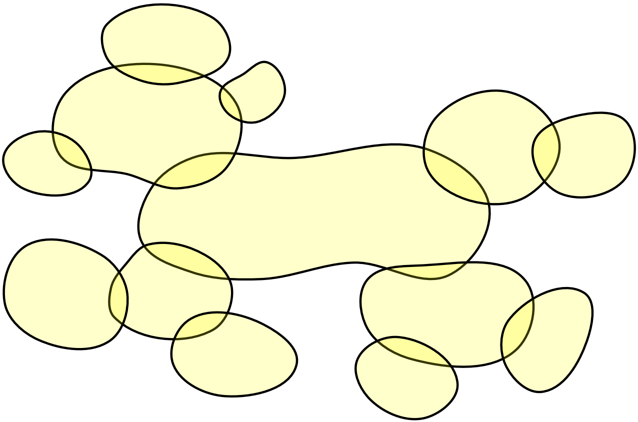
The decomposition theorem

Let G be a graph and k be an integer. Then there exists a tree decomposition $\mathcal{T} = (T, \{B_u\}_{u \in V(T)})$ of G such that:

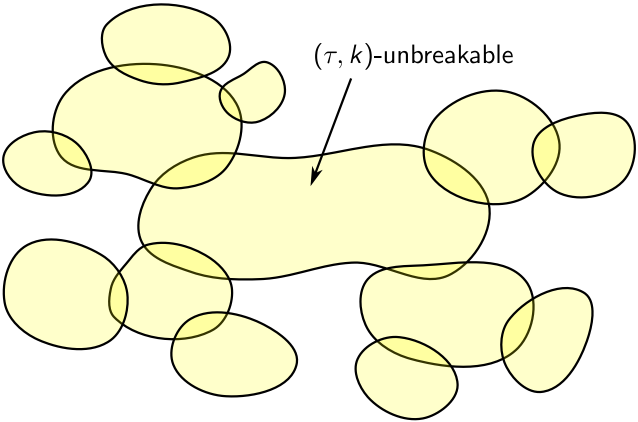
- Each B_u is (τ, k)-unbreakable in G, where $\tau = 2^{\mathcal{O}(k)}$.
- $|B_u \cap B_v| \leq \eta$ for each $uv \in E(T)$, where $\eta = 2^{\mathcal{O}(k)}$.
- $B_u \cap B_v$ is $(2k, k)$-unbreakable in G, for each $uv \in E(T)$.

Moreover, this decomposition can be computed in $2^{\mathcal{O}(k^2)} \cdot n^2 m$ time.

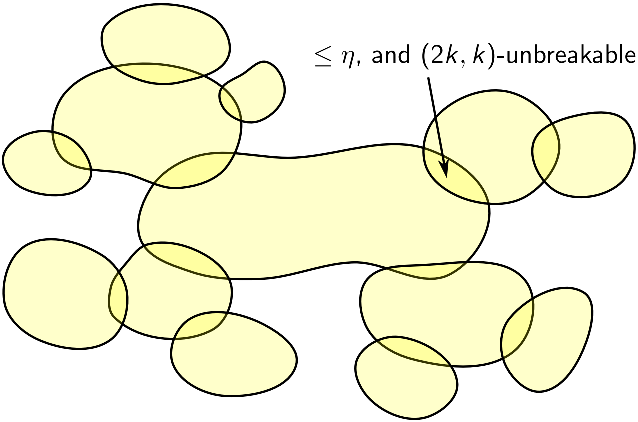
On a picture



On a picture



On a picture



A glimpse into the proof

- **Now**: A sketch of the proof, without unbreakability of adhesions.

A glimpse into the proof

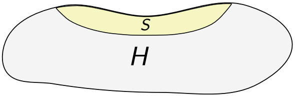
- **Now**: A sketch of the proof, without unbreakability of adhesions.
- Construct the decomposition in a standard top-down manner.

A glimpse into the proof

- **Now**: A sketch of the proof, without unbreakability of adhesions.
- Construct the decomposition in a standard top-down manner.
- Problem to solve at one step:

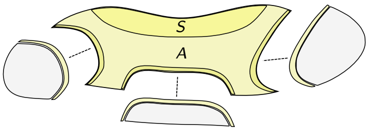
A glimpse into the proof

- **Now**: A sketch of the proof, without unbreakability of adhesions.
- Construct the decomposition in a standard top-down manner.
- Problem to solve at one step:
  - **Input**: Subgraph $H \subseteq G$ to be decomposed, and a set $S \subseteq V(H)$, $|S| \leq \eta$, that will be the top adhesion.



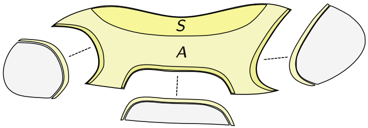
A glimpse into the proof

- **Now**: A sketch of the proof, without unbreakability of adhesions.
- Construct the decomposition in a standard top-down manner.
- Problem to solve at one step:
  - **Input**: Subgraph $H \subseteq G$ to be decomposed, and a set $S \subseteq V(H)$, $|S| \leq \eta$, that will be the top adhesion.
  - **Output**: A candidate unbreakable bag $A \supseteq S$, such that every connected component of $H \setminus A$ sees only $\leq \eta$ vertices in A.



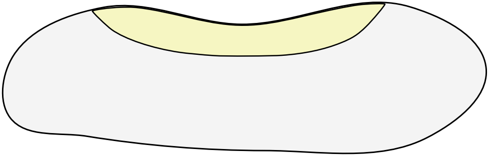
A glimpse into the proof

- **Now**: A sketch of the proof, without unbreakability of adhesions.
- Construct the decomposition in a standard top-down manner.
- Problem to solve at one step:
  - **Input**: Subgraph $H \subseteq G$ to be decomposed, and a set $S \subseteq V(H)$, $|S| \leq \eta$, that will be the top adhesion.
  - **Output**: A candidate unbreakable bag $A \supseteq S$, such that every connected component of $H \setminus A$ sees only $\leq \eta$ vertices in A.
- Then recurse into the connected components below.



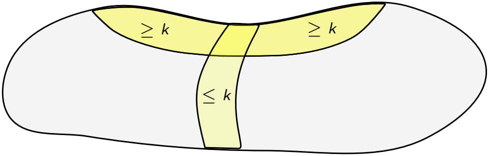
Breakable S

- Test whether S is (k, k)-breakable in H.



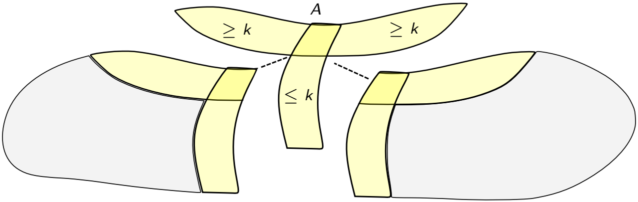
Breakable S

- Test whether S is (k, k)-breakable in H.
- If so, then just break.



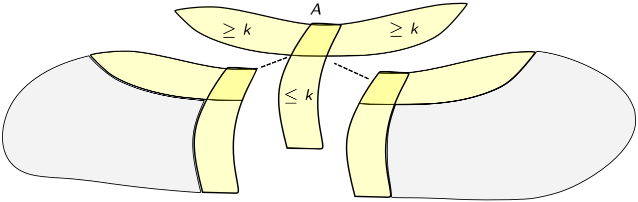
Breakable S

- Test whether S is (k, k)-breakable in H.
- If so, then just break.
- $|A| \leq \eta + k$, so A is (τ, k)-unbreakable for any $\tau \geq \eta + k$.



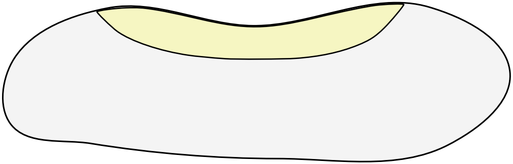
Breakable S

- Test whether S is (k, k)-breakable in H.
- If so, then just break.
- $|A| \leq \eta + k$, so A is (τ, k)-unbreakable for any $\tau \geq \eta + k$.
- Every connected component below is either on the left or on the right, so it sees only $\leq \eta$ vertices from A.



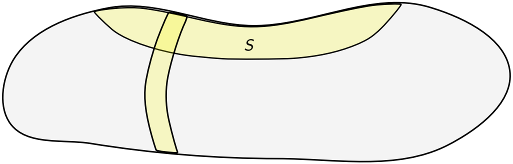
Unbreakable S

- Assume then that S is (k, k)-unbreakable.



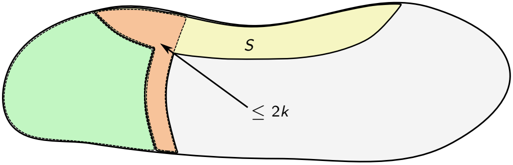
Unbreakable S

- Assume then that S is (k, k)-unbreakable.
- Every separation (A, B) of order $\leq k$ in H has a small side (with at most k vertices from S), and a large size (with the rest).



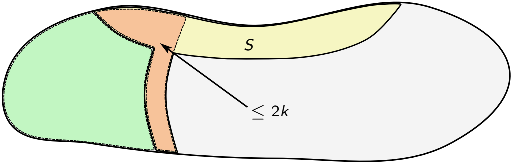
Unbreakable S

- Assume then that S is (k, k)-unbreakable.
- Every separation (A, B) of order $\leq k$ in H has a small side (with at most k vertices from S), and a large size (with the rest).
- Adding these k vertices to the large side gives a separation of order $\leq 2k$ that carves out a bunch of vertices from S.

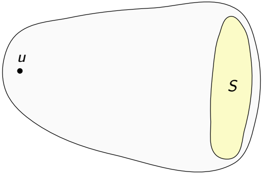


Unbreakable S

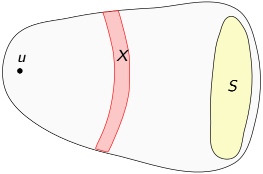
- Assume then that S is (k, k)-unbreakable.
- Every separation (A, B) of order $\leq k$ in H has a small side (with at most k vertices from S), and a large size (with the rest).
- Adding these k vertices to the large side gives a separation of order $\leq 2k$ that carves out a bunch of vertices from S.
- **Idea**: Mark greedily **all** possible such carvings; what is left constitutes A.



Important separators

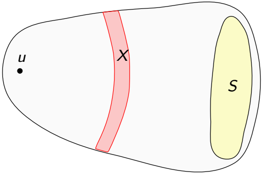


Important separators



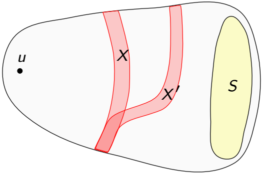
- *X* is an *important u-S separator* if

Important separators



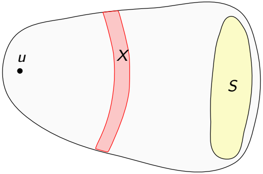
- X is an *important u-S separator* if
  - X is a minimal u-S separator, and

Important separators



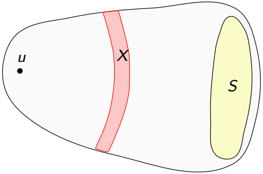
- X is an *important u-S separator* if
  - X is a minimal u-S separator, and
  - there is no u-S separator X' such that
    (i) $|X'| \leq |X|$ and
    (ii) $\mathrm{reach}(u, H \setminus X) \subsetneq \mathrm{reach}(u, H \setminus X')$.

Important separators



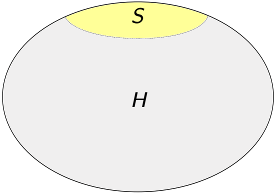
- X is an *important u-S separator* if
  - X is a minimal u-S separator, and
  - there is no u-S separator X' such that
    - (i) $|X'| \leq |X|$ and
    - (ii) $\text{reach}(u, H \setminus X) \subsetneq \text{reach}(u, H \setminus X')$.
- **Note**: minimality $\Rightarrow X = N(\text{reach}(u, H \setminus X))$.

Important separators

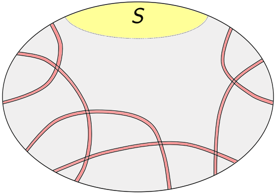


- X is an *important u-S separator* if
  - X is a minimal u-S separator, and
  - there is no u-S separator X' such that
    - (i) $|X'| \leq |X|$ and
    - (ii) $\text{reach}(u, H \setminus X) \subsetneq \text{reach}(u, H \setminus X')$.
- **Note**: minimality $\Rightarrow X = N(\text{reach}(u, H \setminus X))$.
- For fixed u and S, there are at most 4^k important u-S separators of size at most k. [Chen et al., Marx and Razgon]

Chips and the decomposition

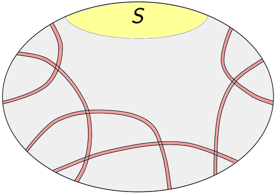


Chips and the decomposition



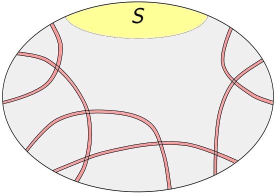
- For every $u \in V(H)$, list all the important separators u-S of size at most $2k$.

Chips and the decomposition



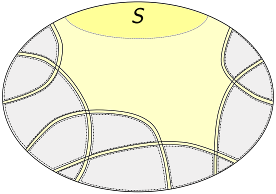
- For every $u \in V(H)$, list all the important separators u-S of size at most $2k$.
- For each separator, take the set of vertices reachable from u. Filter out all such sets that are not inclusion-wise maximal.

Chips and the decomposition



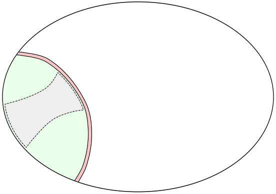
- For every $u \in V(H)$, list all the important separators u-S of size at most $2k$.
- For each separator, take the set of vertices reachable from u. Filter out all such sets that are not inclusion-wise maximal.
- The obtained sets will be called **chips**, and the family of chips is called $\mathcal{C}$. Note that $|\mathcal{C}| \leq 4^{2k} \cdot n$.

Chips and the decomposition



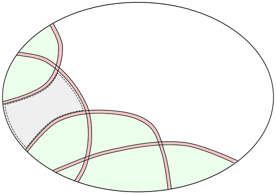
- For every $u \in V(H)$, list all the important separators u-S of size at most $2k$.
- For each separator, take the set of vertices reachable from u. Filter out all such sets that are not inclusion-wise maximal.
- The obtained sets will be called **chips**, and the family of chips is called $\mathcal{C}$. Note that $|\mathcal{C}| \leq 4^{2k} \cdot n$.
- Set $A = \bigcup_{C \in \mathcal{C}} N(C) \cup \bigcap_{C \in \mathcal{C}} V(H) \setminus N[C]$.

Bounding the adhesions



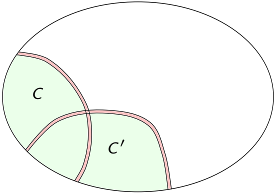
- Examine one connected component D of $H \setminus A$ and suppose that it is contained in chip C.

Bounding the adhesions



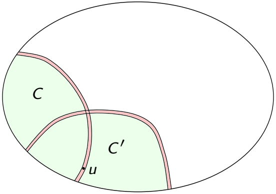
- Examine one connected component D of $H \setminus A$ and suppose that it is contained in chip C.
- $N(D)$ is contained in $N(C)$ plus the neighbourhoods of all chips C' that interfere with C.

Bounding the adhesions



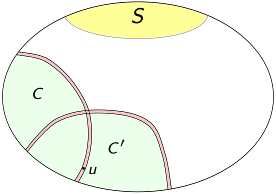
- Examine one connected component D of $H \setminus A$ and suppose that it is contained in chip C.
- $N(D)$ is contained in $N(C)$ plus the neighbourhoods of all chips C' that interfere with C.
- C and C' *touch* if $C \cap C' \neq \emptyset$ or $E(C, C') \neq \emptyset$.

Bounding the adhesions



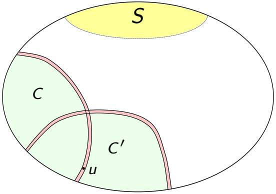
- Examine one connected component D of $H \setminus A$ and suppose that it is contained in chip C.
- $N(D)$ is contained in $N(C)$ plus the neighbourhoods of all chips C' that interfere with C.
- C and C' *touch* if $C \cap C' \neq \emptyset$ or $E(C, C') \neq \emptyset$.
- Connectivity $\Rightarrow$ There exists $u \in N(C) \cap C'$.

Bounding the adhesions, continued



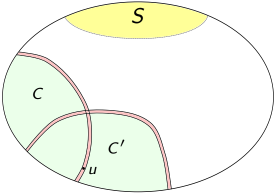
- **Observation**: $N(C')$ is an important u-S separator.

Bounding the adhesions, continued



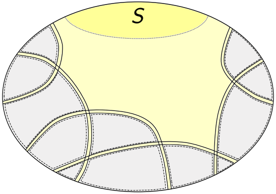
- **Observation**: $N(C')$ is an important u-S separator.
- Hence, every chip touching C is raised by an important separator from a vertex of $N(C)$.

Bounding the adhesions, continued



- **Observation**: $N(C')$ is an important u-S separator.
- Hence, every chip touching C is raised by an important separator from a vertex of $N(C)$.
- Ergo every chip touches at most $2k \cdot 4^{2k}$ other chips, and every component of $H \setminus A$ sees at most $(2k) \cdot (2k+1) \cdot 4^{2k}$ vertices.

Bounding the adhesions, continued



- **Observation**: $N(C')$ is an important u-S separator.
- Hence, every chip touching C is raised by an important separator from a vertex of $N(C)$.
- Ergo every chip touches at most $2k \cdot 4^{2k}$ other chips, and every component of $H \setminus A$ sees at most $(2k) \cdot (2k+1) \cdot 4^{2k}$ vertices.
- **Check**: A is (τ, k)-unbreakable in H, for some $\tau = 2^{\mathcal{O}(k)}$.

Open problems

- Many open questions:

Open problems

- Many open questions:
  - Are exponential-size adhesions necessary?

Open problems

- Many open questions:
  - Are exponential-size adhesions necessary?
  - Does unbreakability need to depend exponentially on k?

Open problems

- Many open questions:
  - Are exponential-size adhesions necessary?
  - Does unbreakability need to depend exponentially on k?
  - Can we compute the decomposition quicker?

Open problems

- Many open questions:
  - Are exponential-size adhesions necessary?
  - Does unbreakability need to depend exponentially on k?
  - Can we compute the decomposition quicker?
    - Both terms $2^{\mathcal{O}(k^2)}$ and $n^2 m$ seem suboptimal.

Open problems

- Many open questions:
  - Are exponential-size adhesions necessary?
  - Does unbreakability need to depend exponentially on k?
  - Can we compute the decomposition quicker?
    - Both terms $2^{\mathcal{O}(k^2)}$ and $n^2 m$ seem suboptimal.
- Can we find other applications of the decomposition theorem?

Open problems

- Many open questions:
  - Are exponential-size adhesions necessary?
  - Does unbreakability need to depend exponentially on k?
  - Can we compute the decomposition quicker?
    - Both terms $2^{\mathcal{O}(k^2)}$ and $n^2 m$ seem suboptimal.
- Can we find other applications of the decomposition theorem?
- **Thanks for attention!**