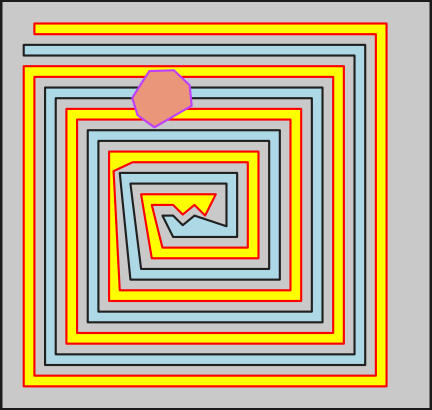
Quasi-Polynomial Time Approximation Scheme for Sparse Subsets of Polygons

Sariel Har-Peled[1]

[1]UIUC, Illinois, USA

2: The problem

- $\mathcal{P}$: m polygons in the plane.
- n: Total complexity of $\mathcal{P}$.

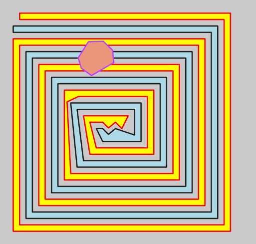


2: The problem

- $\mathcal{P}$: m polygons in the plane.
- n: Total complexity of $\mathcal{P}$.

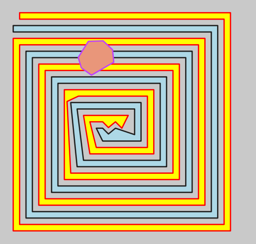
Task...

Compute largest independent set of polygons in $\mathcal{P}$.



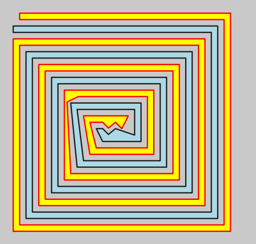
2: The problem

- $\mathcal{P}$: m polygons in the plane.
- n: Total complexity of $\mathcal{P}$.

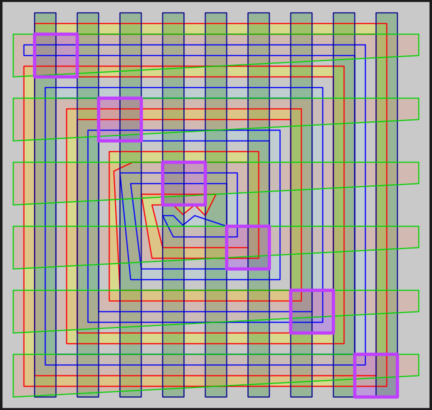


Task...

Compute largest independent set of polygons in $\mathcal{P}$.



3: The problem - another example



4: The main result

Theorem

1. $\mathcal{P}$: set m simple weighted polygons.
2. n: total complexity.
3. Compute an independent set weight $\geq (1-\varepsilon)W_{\mathrm{opt}}$.
4. W_{opt}: maximum weight of optimal independent set.
5. Running time: $O\left(m^{\mathrm{poly}(\log m, 1/\varepsilon)} + m^{O(1)}n\right)$.

4: The main result

Theorem

1. $\mathcal{P}$: set m simple weighted polygons.
2. n: total complexity.
3. Compute an independent set weight $\geq (1-\varepsilon)W_{\mathrm{opt}}$.
4. W_{opt}: maximum weight of optimal independent set.
5. Running time: $O\left(m^{\mathrm{poly}(\log m, 1/\varepsilon)} + m^{O(1)} n\right)$.

4: The main result

Theorem

1. $\mathcal{P}$: set m simple weighted polygons.
2. n: total complexity.
3. Compute an independent set weight $\geq (1-\varepsilon)W_{\mathrm{opt}}$.
4. W_{opt}: maximum weight of optimal independent set.
5. Running time: $O\left(m^{\mathrm{poly}(\log m, 1/\varepsilon)} + m^{O(1)}n\right)$.

4: The main result

Theorem

1. $\mathcal{P}$: set m simple weighted polygons.
2. n: total complexity.
3. Compute an independent set weight $\geq (1-\varepsilon)W_{\mathrm{opt}}$.
4. W_{opt}: maximum weight of optimal independent set.
5. Running time: $O\left(m^{\mathrm{poly}(\log m, 1/\varepsilon)} + m^{O(1)}n\right)$.

5: Approximation algorithms running time

Definition (PTAS)

$\varepsilon > 0$: Parameter
n: Input sizes
$(1 - \varepsilon)$-approximation
For fixed $\varepsilon > 0$, running time $n^{O(1)}$.
PTAS $=$ Polynomial time approximation scheme.

Definition (QPTAS)

$(1 - \varepsilon)$-approximation
Fixed $\varepsilon > 0$, running time $n^{\log^{O(1)} n}$.
QPTAS $=$ **Quasi** polynomial time approximation scheme.

5: Approximation algorithms running time

Definition (PTAS)

$\varepsilon > 0$: Parameter

n: Input sizes

$(1 - \varepsilon)$-approximation

For fixed $\varepsilon > 0$, running time $n^{O(1)}$.

PTAS $=$ Polynomial time approximation scheme.

Definition (QPTAS)

$(1 - \varepsilon)$-approximation

Fixed $\varepsilon > 0$, running time $n^{\log^{O(1)} n}$.

QPTAS $=$ **Quasi** polynomial time approximation scheme.

6: Why care about QPTAS

"There is no such thing as quasicrystals, only quasi-scientists." – Linus Pauling

---

Intuition

$\exists$ **QPTAS** "$\implies$" $\exists$ **PTAS**.

6: Why care about QPTAS

"There is no such thing as quasicrystals, only quasi-scientists." – Linus Pauling

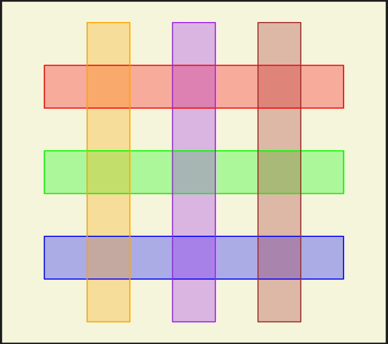
---

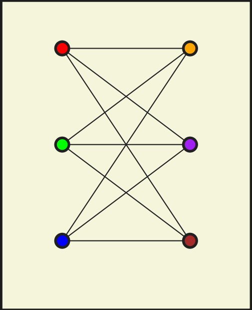
Intuition

$\exists$ **QPTAS** "$\Longrightarrow$" $\exists$ **PTAS**.

7: Intersection graph

Find independent set in the graph.





8: Computing independent set

What is known...

1. **NP-Complete**.
2. No $|V|^{1-\varepsilon}$-approx if **NP** $\neq$ **ZPP** **[Hastad, 1996]**.
3. Maximum degree of graph is $\leq \mathbf{3}$: no **PTAS** possible.

9: Geometric settings

What is known...

1. PTASes for fat objects (disks/squares).
   1. [Chan, 2003, Erlebach, Jansen, and Seidel, 2005]: Quadtrees + random shifting
   2. [Chan, 2003]: Uses planar separator (unweighted).
2. Axis-parallel rectangles.
   1. [Chalermsook and Chuzhoy, 2009]: $O(\log\log n)$ approximation.
   2. [Chan and Har-Peled, 2012]: $O(\log n / \log\log n)$-approx for weighted case.
3. Pseudo-disks
   [Chan and Har-Peled, 2012]:
   $(1+\varepsilon)$-approximation (PTAS).
   Local search + separator theorem.

10: Geometric settings

Continued...

1. Line segments
   1. **[Agarwal and Mustafa, 2006]**: $O\left(\sqrt{n_{\text{opt}}}\right)$-approx.
      Dilworth's theorem.
   2. **[Fox and Pach, 2011]**: n^{ε}-approx.
      Idea: Large biclique if dense, and cheap separator if sparse.
2. New results: **[Adamaszek and Wiese, 2013, 2014]**.
   1. Good news: $(1+\varepsilon)$-approximation for rectangles.
   2. Bad news: Running time $n^{(\log n/\varepsilon)^{O(1)}}$.
      QPTAS: Quasi-polynomial time approximation scheme
   3. Main idea: Cuttings + planar separator.

10: Geometric settings

Continued...

1. Line segments
   1. [Agarwal and Mustafa, 2006]: $O\left(\sqrt{n_{\text{opt}}}\right)$-approx. Dilworth's theorem.
   2. [Fox and Pach, 2011]: n^{ε}-approx. Idea: Large biclique if dense, and cheap separator if sparse.
2. New results: [Adamaszek and Wiese, 2013, 2014].
   1. Good news: $(1+\varepsilon)$-approximation for rectangles.
   2. Bad news: Running time $n^{(\log n/\varepsilon)^{O(1)}}$. **QPTAS**: Quasi-polynomial time approximation scheme
   3. Main idea: Cuttings + planar separator.

11: Why **QPTAS** are interesting?

A soup of complexity classes.

- X is **APXHard** $\implies$ no **PTAS** for X (no ε-approx).
- PCP Theorem ($\forall \varepsilon < 1/8$):
  $(1+\varepsilon)$-approx to **Max3SAT** $\implies$ solving **3SAT** exactly.
- **ETH** (**Exp. time hypothesis**):
  3SAT not solvable in $2^{o(n)}$.
- X has a **QPTAS** & X is **APXHard** $\implies$ **ETH** is false.
- X has a **QPTAS** $\implies$ X is not **APXHard**.
  ...hope for an efficient $(1+\varepsilon)$-approx.

- X is **APXHard** $\implies$ no **PTAS** for X.
- X is **APXHard** $\implies$ no **QPTAS** for X.

11: Why QPTAS are interesting?

A soup of complexity classes.

- X is **APXHard** $\implies$ no **PTAS** for X (no ε-approx).
- PCP Theorem ($\forall \varepsilon < 1/8$):
  $(1+\varepsilon)$-approx to **Max3SAT** $\implies$ solving **3SAT** exactly.
- **ETH** (**Exp. time hypothesis**):
  3SAT not solvable in $2^{o(n)}$.
- X has a **QPTAS** & X is **APXHard** $\implies$ **ETH** is false.
- X has a **QPTAS** $\implies$ X is not **APXHard**.
  ...hope for an efficient $(1+\varepsilon)$-approx.

- X is **APXHard** $\implies$ no **PTAS** for X.
- X is **APXHard** $\implies$ no **QPTAS** for X.

11: Why QPTAS are interesting?

A soup of complexity classes.

1. X is **APXHard** $\implies$ no **PTAS** for X (no ε-approx).
2. PCP Theorem ($\forall \varepsilon < 1/8$):
   $(1+\varepsilon)$-approx to **Max3SAT** $\implies$ solving **3SAT** exactly.
3. **ETH** (**Exp. time hypothesis**):
   3SAT not solvable in $2^{o(n)}$.
4. X has a **QPTAS** & X is **APXHard** $\implies$ **ETH** is false.
5. X has a **QPTAS** $\implies$ X is not **APXHard**.
   ...hope for an efficient $(1+\varepsilon)$-approx.

- X is **APXHard** $\implies$ no **PTAS** for X.
- X is **APXHard** $\implies$ no **QPTAS** for X.

11: Why QPTAS are interesting?

A soup of complexity classes.

1. X is **APXHard** $\implies$ no **PTAS** for X (no ε-approx).
2. PCP Theorem ($\forall \varepsilon < 1/8$):
   $(1+\varepsilon)$-approx to **Max3SAT** $\implies$ solving **3SAT** exactly.
3. **ETH** (**Exp. time hypothesis**):
   3SAT not solvable in $2^{o(n)}$.
4. X has a **QPTAS** & X is **APXHard** $\implies$ **ETH** is false.
5. X has a **QPTAS** $\implies$ X is not **APXHard**.
   ...hope for an efficient $(1+\varepsilon)$-approx.

- X is **APXHard** $\implies$ no **PTAS** for X.
- X is **APXHard** $\implies$ no **QPTAS** for X.

11: Why QPTAS are interesting?

A soup of complexity classes.

1. X is **APXHard** $\implies$ no **PTAS** for X (no ε-approx).
2. PCP Theorem ($\forall \varepsilon < 1/8$):
   $(1+\varepsilon)$-approx to **Max3SAT** $\implies$ solving **3SAT** exactly.
3. **ETH** (**Exp. time hypothesis**):
   3SAT not solvable in $2^{o(n)}$.
4. X has a **QPTAS** & X is **APXHard** $\implies$ **ETH** is false.
5. X has a **QPTAS** $\implies$ X is not **APXHard**.
   ...hope for an efficient $(1+\varepsilon)$-approx.

- X is **APXHard** $\implies$ no **PTAS** for X.
- X is **APXHard** $\implies$ no **QPTAS** for X.

11: Why QPTAS are interesting?

A soup of complexity classes.

1. X is **APXHard** $\implies$ no **PTAS** for X (no ε-approx).
2. PCP Theorem ($\forall \varepsilon < 1/8$):
   $(1+\varepsilon)$-approx to **Max3SAT** $\implies$ solving **3SAT** exactly.
3. **ETH** (**Exp. time hypothesis**):
   3SAT not solvable in $2^{o(n)}$.
4. X has a **QPTAS** & X is **APXHard** $\implies$ **ETH** is false.
5. X has a **QPTAS** $\implies$ X is not **APXHard**.
   ...hope for an efficient $(1+\varepsilon)$-approx.

---

1. X is **APXHard** $\implies$ no **PTAS** for X.
2. X is **APXHard** $\implies$ no **QPTAS** for X.

12: Thinking about these complexity classes

Polynomial time	$\exp(O(\log n))$
FPTAS	$\exp\left(O\left(\log \frac{n}{\varepsilon}\right)\right)$
PTAS	$\exp\left(O\left(f(\varepsilon) \log n\right)\right)$
QPTAS	$\exp\left(f(\varepsilon) \log^{O(1)} n\right)$
ETH	3SAT $\in \exp\left(n^{\Omega(1)}\right)$
EXP	$\exp\left(n^{O(1)}\right)$

Part I

Extremal graphs, Separators and Sparsity

13: A quicky on planar separators

1. **[Lipton and Tarjan, 1977]**
   Every planar graph has a separator of size $O(\sqrt{n})$.
2. **[Miller, 1986]**
   2-connected planar graph: $\exists$ **cycle** separator $O(\sqrt{n})$.
3. **[Miller, Teng, Thurston, and Vavasis, 1997]**
   Circle packing theorem $\implies$ planar separator.
4. [Har-Peled, 2013]: Circle packing easily implies planar cycle separator.
5. Circle packing theorem **[Koebe, 1936]**.

14: When are graphs sparse?

- $G = (V, E)$, $n = |V|$.
- When $|E|$ is near linear?
- Planar graphs, low genus graphs. Etc.
- If $K_{t,t}$ is a forbidden subgraph $\implies |E| \le O(n^{2-1/t})$

Theorem ([Fox and Pach, 2008])

- $\mathcal{P}$: *Set of n curves in the plane ($O(1)$ intersections for pair).*
- $G = (\mathcal{P}, E)$: *Intersection graph.*
- *For any subset $X \subseteq \mathcal{P}$: $|E(G_X)| \le |X|^{2-\delta}$. δ: Constant*
- $\implies |E(G_X)| = O(n)$.

14: When are graphs sparse?

- $G = (V, E)$, $n = |V|$.
- When $|E|$ is near linear?
- Planar graphs, low genus graphs. Etc.
- If $K_{t,t}$ is a forbidden subgraph $\implies |E| \leq O(n^{2-1/t})$

Theorem ([Fox and Pach, 2008])

- *$\mathcal{P}$: Set of n curves in the plane ($O(1)$ intersections for pair).*
- *$G = (\mathcal{P}, E)$: Intersection graph.*
- *For any subset $X \subseteq \mathcal{P}$: $|E(G_X)| \leq |X|^{2-\delta}$. δ: Constant*
- $\implies |E(G_X)| = O(n)$.

14: When are graphs sparse?

- $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, $n = |\mathsf{V}|$.
- When $|\mathsf{E}|$ is near linear?
- Planar graphs, low genus graphs. Etc.
- If $K_{t,t}$ is a forbidden subgraph
  $\implies |\mathsf{E}| \leq O\left(n^{2-1/t}\right)$

Theorem ([Fox and Pach, 2008])

- *$\mathcal{P}$: Set of n curves in the plane ($O(1)$ intersections for pair).*
- *$\mathsf{G} = (\mathcal{P}, \mathsf{E})$: Intersection graph.*
- *For any subset $X \subseteq \mathcal{P}$: $|\mathsf{E}(\mathsf{G}_X)| \leq |X|^{2-\delta}$. δ: Constant*
- $\implies |\mathsf{E}(\mathsf{G}_X)| = O(n)$.

14: When are graphs sparse?

- $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, $n = |\mathbf{V}|$.
- When $|\mathbf{E}|$ is near linear?
- Planar graphs, low genus graphs. Etc.
- If $K_{t,t}$ is a forbidden subgraph
  $\implies |\mathbf{E}| \leq O\left(n^{2-1/t}\right)$

Theorem ([Fox and Pach, 2008])

- *$\mathcal{P}$: Set of n curves in the plane ($O(1)$ intersections for pair).*
- *$\mathbf{G} = (\mathcal{P}, \mathbf{E})$: Intersection graph.*
- *For any subset $\mathbf{X} \subseteq \mathcal{P}$: $|\mathbf{E}(\mathbf{G}_X)| \leq |\mathbf{X}|^{2-\delta}$. δ: Constant*
- $\implies |\mathbf{E}(\mathbf{G}_X)| = O(n)$.

14: When are graphs sparse?

- $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, $n = |\mathsf{V}|$.
- When $|\mathsf{E}|$ is near linear?
- Planar graphs, low genus graphs. Etc.
- If $K_{t,t}$ is a forbidden subgraph
  $\implies |\mathsf{E}| \leq O\left(n^{2-1/t}\right)$

Theorem ([Fox and Pach, 2008])

- $\mathcal{P}$: *Set of* n *curves in the plane* *(*$O(1)$ *intersections for pair).*
- $\mathsf{G} = (\mathcal{P}, \mathsf{E})$: *Intersection graph.*
- *For any subset* $X \subseteq \mathcal{P}$: $|\mathsf{E}(\mathsf{G}_X)| \leq |X|^{2-\delta}$.
  δ: *Constant*
- $\implies |\mathsf{E}(\mathsf{G}_X)| = O(n)$.

14: When are graphs sparse?

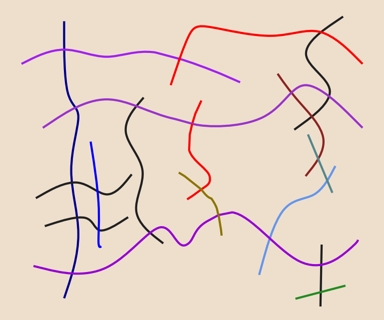
1. $G = (V, E)$, $n = |V|$.
2. When $|E|$ is near linear?
3. Planar graphs, low genus graphs. Etc.
4. If $K_{t,t}$ is a forbidden subgraph $\implies |E| \leq O\left(n^{2-1/t}\right)$

Theorem ([Fox and Pach, 2008])

1. *$\mathcal{P}$: Set of n curves in the plane ($O(1)$ intersections for pair).*
2. *$G = (\mathcal{P}, E)$: Intersection graph.*
3. *For any subset $X \subseteq \mathcal{P}$: $|E(G_X)| \leq |X|^{2-\delta}$. δ: Constant*
4. $\implies |E(G_X)| = O(n)$.

15: Sparse intersection graphs are really sparse

Proof of [Fox and Pach, 2008]

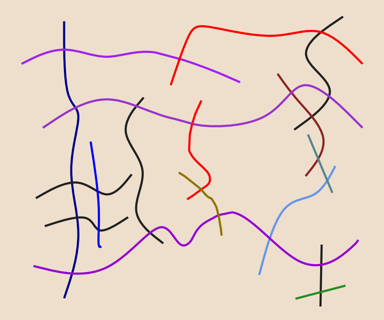


1. $\mathcal{P}$: Set of curves. $n = |\mathcal{P}|$.
2. Compute arrangement G: planar graph $|\mathbf{V(G)}|, |\mathbf{E(G)}| \leq n^{2-\delta}$.
3. S: Vertex separator $O\left(\sqrt{n^{2-\delta}}\right)$.
4. Cut at S, & recurse.

$$T(n) = n^{1-\delta/2} + 2T\left(n/2 + n^{1-\delta/2}\right) = O(n).$$

15: Sparse intersection graphs are really sparse

Proof of [Fox and Pach, 2008]

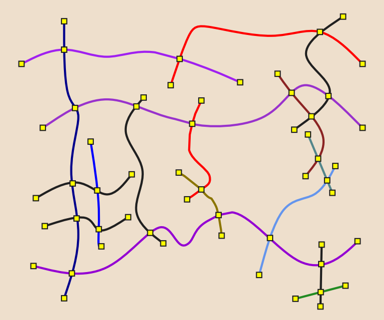


1. $\mathcal{P}$: Set of curves. $n = |\mathcal{P}|$.
2. Compute arrangement
   G: planar graph
   $|\mathbf{V(G)}|, |\mathbf{E(G)}| \leq n^{2-\delta}$.
3. S: Vertex separator
   $O\left(\sqrt{n^{2-\delta}}\right)$.
4. Cut at S, & recurse.

$$T(n) = n^{1-\delta/2} + 2T\left(n/2 + n^{1-\delta/2}\right) = O(n).$$

15: Sparse intersection graphs are really sparse

Proof of [Fox and Pach, 2008]

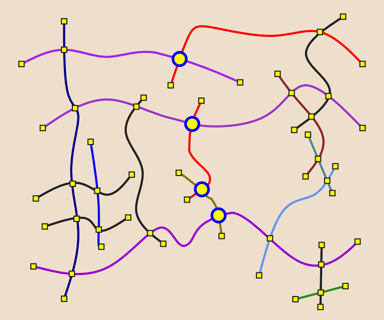


1. $\mathcal{P}$: Set of curves. $n = |\mathcal{P}|$.
2. Compute arrangement G: planar graph $|\mathbf{V}(\mathbf{G})|, |\mathbf{E}(\mathbf{G})| \leq n^{2-\delta}$.
3. S: Vertex separator $O\left(\sqrt{n^{2-\delta}}\right)$.
4. Cut at S, & recurse.

$$T(n) = n^{1-\delta/2} + 2T\left(n/2 + n^{1-\delta/2}\right) = O(n).$$

15: Sparse intersection graphs are really sparse

Proof of [Fox and Pach, 2008]

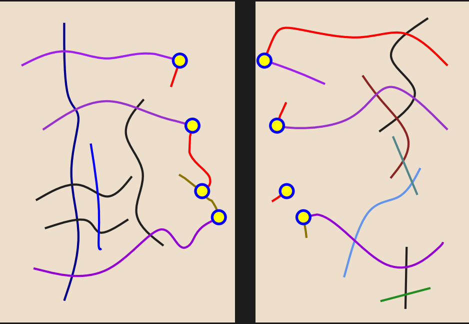


1. $\mathcal{P}$: Set of curves. $n = |\mathcal{P}|$.
2. Compute arrangement G: planar graph $|\mathbf{V(G)}|, |\mathbf{E(G)}| \leq n^{2-\delta}$.
3. S: Vertex separator $O\left(\sqrt{n^{2-\delta}}\right)$.
4. Cut at S, & recurse.

$$T(n) = n^{1-\delta/2} + 2T\left(n/2 + n^{1-\delta/2}\right) = O(n).$$

15: Sparse intersection graphs are really sparse

Proof of **[Fox and Pach, 2008]**

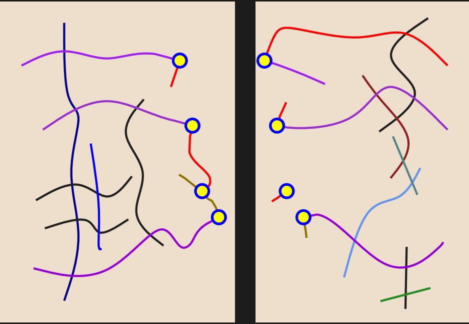


1. $\mathcal{P}$: Set of curves. $n = |\mathcal{P}|$.
2. Compute arrangement G: planar graph $|\mathbf{V(G)}|, |\mathbf{E(G)}| \leq n^{2-\delta}$.
3. S: Vertex separator $O\left(\sqrt{n^{2-\delta}}\right)$.
4. Cut at S, & recurse.

$$T(n) = n^{1-\delta/2} + 2T\left(n/2 + n^{1-\delta/2}\right) = O(n).$$

15: Sparse intersection graphs are really sparse

Proof of **[Fox and Pach, 2008]**



1. $\mathcal{P}$: Set of curves. $n = |\mathcal{P}|$.
2. Compute arrangement G: planar graph $|\mathbf{V(G)}|, |\mathbf{E(G)}| \leq n^{2-\delta}$.
3. S: Vertex separator $O\left(\sqrt{n^{2-\delta}}\right)$.
4. Cut at S, & recurse.

$$T(n) = n^{1-\delta/2} + 2T\left(n/2 + n^{1-\delta/2}\right) = O(n).$$

Part II

Independent sets are sparse subsets

16: Properties of sparse subsets

- $\mathcal{P}$: Set of polygons.
- $\mathbf{\Pi}$: Property (e.g., polygons are disjoint).
- $\mathbf{\Pi}_{\mathcal{P}} = (\mathcal{P}, \mathcal{I})$, where
  $\mathcal{I} = \left\{ X \subseteq \mathcal{P} \;\middle|\; X \text{ has property } \mathbf{\Pi} \right\}$.
- **Task:** Compute heaviest $X \in \mathbf{\Pi}_{\mathcal{P}}$.
- Require of the property:
  - **hereditary**: $\forall X \in \mathbf{\Pi}_{\mathcal{P}} \quad \forall Y \subseteq X \implies Y \in \mathbf{\Pi}_{\mathcal{P}}$.
  - **mergeable**; $\forall X, Y \in \mathbf{\Pi}_{\mathcal{P}} \quad X \cap Y = \emptyset \implies X \cup Y \in \mathbf{\Pi}_{\mathcal{P}}$.
  - **sparse**: $\forall X \in \mathbf{\Pi}_{\mathcal{P}} \implies |E(G_X)| \le c' |X|^{2-\delta}$.
  - **exponential time checkable**:
    $\forall X \subseteq \mathcal{P}$, decide if $X \in \mathbf{\Pi}_{\mathcal{P}}$ in time $2^{|X|^{O(1)}}$.

16: Properties of sparse subsets

1. $\mathcal{P}$: Set of polygons.
2. Π: Property (e.g., polygons are disjoint).
3. $\Pi_{\mathcal{P}} = (\mathcal{P}, \mathcal{I})$, where $\mathcal{I} = \left\{ X \subseteq \mathcal{P} \;\middle|\; X \text{ has property } \Pi \right\}$.
4. **Task:** Compute heaviest $X \in \Pi_{\mathcal{P}}$.
5. Require of the property:
   1. **hereditary**: $\forall X \in \Pi_{\mathcal{P}} \quad \forall Y \subseteq X \implies Y \in \Pi_{\mathcal{P}}$.
   2. **mergeable**; $\forall X, Y \in \Pi_{\mathcal{P}} \quad X \cap Y = \emptyset \implies X \cup Y \in \Pi_{\mathcal{P}}$.
   3. **sparse**: $\forall X \in \Pi_{\mathcal{P}} \implies |E(G_X)| \leq c' |X|^{2-\delta}$.
   4. **exponential time checkable**: $\forall X \subseteq \mathcal{P}$, decide if $X \in \Pi_{\mathcal{P}}$ in time $2^{|X|^{O(1)}}$.

16: Properties of sparse subsets

1. $\mathcal{P}$: Set of polygons.
2. Π: Property (e.g., polygons are disjoint).
3. $\Pi_{\mathcal{P}} = (\mathcal{P}, \mathcal{I})$, where
   $\mathcal{I} = \left\{ X \subseteq \mathcal{P} \;\middle|\; X \text{ has property } \Pi \right\}$.
4. **Task:** Compute heaviest $X \in \Pi_{\mathcal{P}}$.
5. Require of the property:
   1. **hereditary**: $\forall X \in \Pi_{\mathcal{P}} \quad \forall Y \subseteq X \implies Y \in \Pi_{\mathcal{P}}$.
   2. **mergeable**; $\forall X, Y \in \Pi_{\mathcal{P}} \quad X \cap Y = \emptyset \implies X \cup Y \in \Pi_{\mathcal{P}}$.
   3. **sparse**: $\forall X \in \Pi_{\mathcal{P}} \implies |E(G_X)| \leq c' |X|^{2-\delta}$.
   4. **exponential time checkable**: $\forall X \subseteq \mathcal{P}$, decide if $X \in \Pi_{\mathcal{P}}$ in time $2^{|X|^{O(1)}}$.

16: Properties of sparse subsets

- $\mathcal{P}$: Set of polygons.
- Π: Property (e.g., polygons are disjoint).
- $\Pi_{\mathcal{P}} = (\mathcal{P}, \mathcal{I})$, where
  $\mathcal{I} = \left\{ X \subseteq \mathcal{P} \;\middle|\; X \text{ has property } \Pi \right\}$.
- **Task:** Compute heaviest $X \in \Pi_{\mathcal{P}}$.
- Require of the property:
  - **hereditary**: $\forall X \in \Pi_{\mathcal{P}} \quad \forall Y \subseteq X \implies Y \in \Pi_{\mathcal{P}}$.
  - **mergeable**: $\forall X, Y \in \Pi_{\mathcal{P}} \quad X \cap Y = \emptyset \implies X \cup Y \in \Pi_{\mathcal{P}}$.
  - **sparse**: $\forall X \in \Pi_{\mathcal{P}} \implies |E(G_X)| \leq c' |X|^{2-\delta}$.
  - **exponential time checkable**: $\forall X \subseteq \mathcal{P}$, decide if $X \in \Pi_{\mathcal{P}}$ in time $2^{|X|^{O(1)}}$.

16: Properties of sparse subsets

1. $\mathcal{P}$: Set of polygons.
2. Π: Property (e.g., polygons are disjoint).
3. $\Pi_{\mathcal{P}} = (\mathcal{P}, \mathcal{I})$, where
   $\mathcal{I} = \left\{ X \subseteq \mathcal{P} \,\middle|\, X \text{ has property } \Pi \right\}$.
4. **Task:** Compute heaviest $X \in \Pi_{\mathcal{P}}$.
5. Require of the property:
   1. **hereditary**: $\forall X \in \Pi_{\mathcal{P}} \quad \forall Y \subseteq X \implies Y \in \Pi_{\mathcal{P}}$.
   2. **mergeable**; $\forall X, Y \in \Pi_{\mathcal{P}} \quad X \cap Y = \emptyset \implies X \cup Y \in \Pi_{\mathcal{P}}$.
   3. **sparse**: $\forall X \in \Pi_{\mathcal{P}} \implies |E(G_X)| \leq c' |X|^{2-\delta}$.
   4. **exponential time checkable**: $\forall X \subseteq \mathcal{P}$, decide if $X \in \Pi_{\mathcal{P}}$ in time $2^{|X|^{O(1)}}$.

16: Properties of sparse subsets

1. $\mathcal{P}$: Set of polygons.
2. Π: Property (e.g., polygons are disjoint).
3. $\Pi_{\mathcal{P}} = (\mathcal{P}, \mathcal{I})$, where
   $\mathcal{I} = \left\{ X \subseteq \mathcal{P} \;\middle|\; X \text{ has property } \Pi \right\}$.
4. **Task:** Compute heaviest $X \in \Pi_{\mathcal{P}}$.
5. Require of the property:
   1. **hereditary**: $\forall X \in \Pi_{\mathcal{P}} \quad \forall Y \subseteq X \implies Y \in \Pi_{\mathcal{P}}$.
   2. **mergeable**; $\forall X, Y \in \Pi_{\mathcal{P}} \quad X \cap Y = \emptyset \implies X \cup Y \in \Pi_{\mathcal{P}}$.
   3. **sparse**: $\forall X \in \Pi_{\mathcal{P}} \implies |E(G_X)| \leq c' |X|^{2-\delta}$.
   4. **exponential time checkable**:
      $\forall X \subseteq \mathcal{P}$, decide if $X \in \Pi_{\mathcal{P}}$ in time $2^{|X|^{O(1)}}$.

16: Properties of sparse subsets

1. $\mathcal{P}$: Set of polygons.
2. Π: Property (e.g., polygons are disjoint).
3. $\Pi_{\mathcal{P}} = (\mathcal{P}, \mathcal{I})$, where
   $\mathcal{I} = \left\{ X \subseteq \mathcal{P} \;\middle|\; X \text{ has property } \Pi \right\}.$
4. **Task:** Compute heaviest $X \in \Pi_{\mathcal{P}}$.
5. Require of the property:
   1. **hereditary**: $\forall X \in \Pi_{\mathcal{P}} \quad \forall Y \subseteq X \implies Y \in \Pi_{\mathcal{P}}$.
   2. **mergeable**; $\forall X, Y \in \Pi_{\mathcal{P}} \quad X \cap Y = \emptyset \implies X \cup Y \in \Pi_{\mathcal{P}}$.
   3. **sparse**: $\forall X \in \Pi_{\mathcal{P}} \implies |E(G_X)| \leq c' |X|^{2-\delta}$.
   4. **exponential time checkable**: $\forall X \subseteq \mathcal{P}$, decide if $X \in \Pi_{\mathcal{P}}$ in time $2^{|X|^{O(1)}}$.

17: Main result II

Theorem

- $\mathcal{P}$: *weighted set of* m *polygons.*
- n: *total complexity*
- *every pair of polygons intersects* $O(1)$ *times.*
- $\Pi_{\mathcal{P}}$: *hereditary* $+$ *sparse* $+$ *mergeable property* $+$ *exponential time checkable.*
- W_{opt}: *maximum weight of a set in* $\Pi_{\mathcal{P}}$.
- $\implies$ **QPTAS** *to compute a subset* $X \subseteq \mathcal{P}$ $X \in \Pi_{\mathcal{P}}$, *and* $\omega(X) \geq (1 - \varepsilon) W_{\mathrm{opt}}$,

17: Main result II

Theorem

1. $\mathcal{P}$: weighted set of m polygons.
2. n: total complexity
3. every pair of polygons intersects $O(1)$ times.
4. $\Pi_{\mathcal{P}}$: hereditary $+$ sparse $+$ mergeable property $+$ exponential time checkable.
5. W_{opt}: maximum weight of a set in $\Pi_{\mathcal{P}}$.
6. $\implies$ **QPTAS** to compute a subset $X \subseteq \mathcal{P}$ $X \in \Pi_{\mathcal{P}}$, and $\omega(X) \geq (1-\varepsilon)W_{\text{opt}}$,

17: Main result II

Theorem

- $\mathcal{P}$: weighted set of m polygons.
- n: total complexity
- every pair of polygons intersects $O(1)$ times.
- $\Pi_{\mathcal{P}}$: hereditary $+$ sparse $+$ mergeable property $+$ exponential time checkable.
- W_{opt}: maximum weight of a set in $\Pi_{\mathcal{P}}$.
- $\implies$ **QPTAS** to compute a subset $X \subseteq \mathcal{P}$ $X \in \Pi_{\mathcal{P}}$, and $\omega(X) \geq (1-\varepsilon)W_{\mathrm{opt}}$,

18: Implications of Main Result II

$\mathcal{P}$: Set of polygons/pseudo-disks.
QPTAS for $(1-\varepsilon)$-approx to heaviest subset $\mathcal{O}$ of $\mathcal{P}$ s.t.

1. Pseudo-disks: every point covered $\leq c$ times by $\mathcal{O}$, c a constant.
   $c = 1$: Independent set.
2. $G_{\mathcal{O}}$ is planar.
3. $G_{\mathcal{O}}$ has low genus.
4. $G_{\mathcal{O}}$ does not contain $K_{s,t}$ as a subgraph. s and t constants.

Main point...

Independent set is just one possible sparse subset one can compute with these techniques.

18: Implications of Main Result II

$\mathcal{P}$: Set of polygons/pseudo-disks.
QPTAS for $(1-\varepsilon)$-approx to heaviest subset $\mathcal{O}$ of $\mathcal{P}$ s.t.

1. Pseudo-disks: every point covered $\leq c$ times by $\mathcal{O}$, c a constant.
   $c = 1$: Independent set.
2. $\mathbf{G}_{\mathcal{O}}$ is planar.
3. $\mathbf{G}_{\mathcal{O}}$ has low genus.
4. $\mathbf{G}_{\mathcal{O}}$ does not contain $K_{s,t}$ as a subgraph. s and t constants.

Main point...

Independent set is just one possible sparse subset one can compute with these techniques.

18: Implications of Main Result II

$\mathcal{P}$: Set of polygons/pseudo-disks.
QPTAS for $(1-\varepsilon)$-approx to heaviest subset $\mathcal{O}$ of $\mathcal{P}$ s.t.

1. Pseudo-disks: every point covered $\leq c$ times by $\mathcal{O}$, c a constant.
   $c = 1$: Independent set.
2. $\mathbf{G}_{\mathcal{O}}$ is planar.
3. $\mathbf{G}_{\mathcal{O}}$ has low genus.
4. $\mathbf{G}_{\mathcal{O}}$ does not contain $K_{s,t}$ as a subgraph. s and t constants.

Main point...

Independent set is just one possible sparse subset one can compute with these techniques.

18: Implications of Main Result II

$\mathcal{P}$: Set of polygons/pseudo-disks.
QPTAS for $(1-\varepsilon)$-approx to heaviest subset $\mathcal{O}$ of $\mathcal{P}$ s.t.

1. Pseudo-disks: every point covered $\leq c$ times by $\mathcal{O}$, c a constant.
   $c = 1$: Independent set.
2. $\mathbf{G}_{\mathcal{O}}$ is planar.
3. $\mathbf{G}_{\mathcal{O}}$ has low genus.
4. $\mathbf{G}_{\mathcal{O}}$ does not contain $K_{s,t}$ as a subgraph. s and t constants.

Main point...

Independent set is just one possible sparse subset one can compute with these techniques.

18: Implications of Main Result II

$\mathcal{P}$: Set of polygons/pseudo-disks.
QPTAS for $(1-\varepsilon)$-approx to heaviest subset $\mathcal{O}$ of $\mathcal{P}$ s.t.

1. Pseudo-disks: every point covered $\leq c$ times by $\mathcal{O}$, c a constant.
   $c = 1$: Independent set.
2. $\mathbf{G}_{\mathcal{O}}$ is planar.
3. $\mathbf{G}_{\mathcal{O}}$ has low genus.
4. $\mathbf{G}_{\mathcal{O}}$ does not contain $K_{s,t}$ as a subgraph. s and t constants.

Main point...

Independent set is just one possible sparse subset one can compute with these techniques.

18: Implications of Main Result II

$\mathcal{P}$: Set of polygons/pseudo-disks.
QPTAS for $(1-\varepsilon)$-approx to heaviest subset $\mathcal{O}$ of $\mathcal{P}$ s.t.

1. Pseudo-disks: every point covered $\leq c$ times by $\mathcal{O}$, c a constant. $c = 1$: Independent set.
2. $\mathbf{G}_{\mathcal{O}}$ is planar.
3. $\mathbf{G}_{\mathcal{O}}$ has low genus.
4. $\mathbf{G}_{\mathcal{O}}$ does not contain $K_{s,t}$ as a subgraph. s and t constants.

Main point...

Independent set is just one possible sparse subset one can compute with these techniques.

18: Implications of Main Result II

$\mathcal{P}$: Set of polygons/pseudo-disks.
QPTAS for $(1-\varepsilon)$-approx to heaviest subset $\mathcal{O}$ of $\mathcal{P}$ s.t.

1. Pseudo-disks: every point covered $\leq c$ times by $\mathcal{O}$, c a constant.
   $c = 1$: Independent set.
2. $\mathbf{G}_{\mathcal{O}}$ is planar.
3. $\mathbf{G}_{\mathcal{O}}$ has low genus.
4. $\mathbf{G}_{\mathcal{O}}$ does not contain $K_{s,t}$ as a subgraph. s and t constants.

Main point...

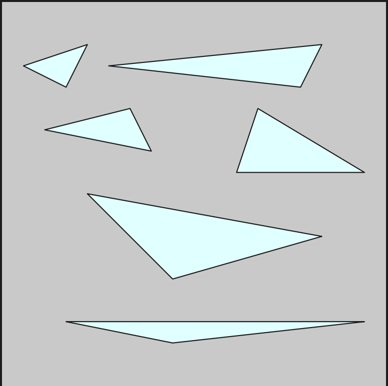
Independent set is just one possible sparse subset one can compute with these techniques.

Deep in the technicalities zone

I: Simple decompositions for polygons

19: Vertical decomposition for triangles

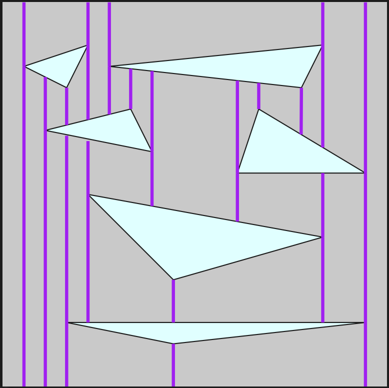
For disjoint triangles



- m disjoint triangles.
- Rays up/down vertices. Get $O(m)$ v-traps.
- Every trapezoid defined by 4 triangles.
- **Q:** How to do VD for polygons?

19: Vertical decomposition for triangles

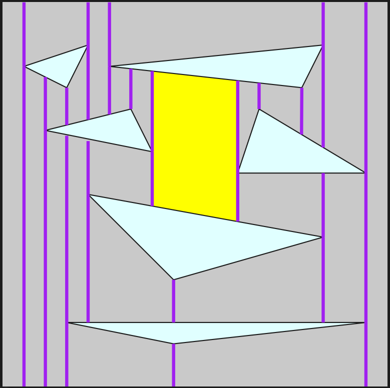
For disjoint triangles



1. m disjoint triangles.
2. Rays up/down vertices. Get $O(m)$ v-traps.
3. Every trapezoid defined by **4** triangles.
4. **Q:** How to do VD for polygons?

19: Vertical decomposition for triangles

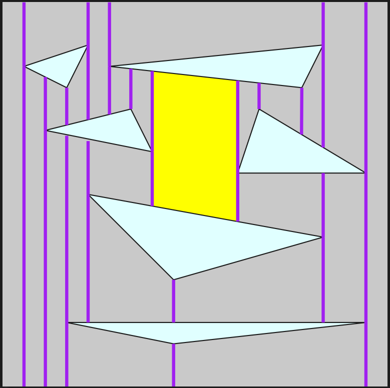
For disjoint triangles



1. m disjoint triangles.
2. Rays up/down vertices. Get $O(m)$ v-traps.
3. Every trapezoid defined by **4** triangles.
4. **Q:** How to do VD for polygons?

19: Vertical decomposition for triangles

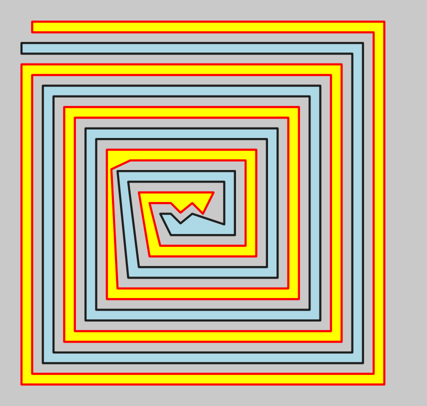
For disjoint triangles



1. m disjoint triangles.
2. Rays up/down vertices. Get $O(m)$ v-traps.
3. Every trapezoid defined by **4** triangles.
4. **Q:** How to do VD for polygons?

20: Vertical decomposition for polygons

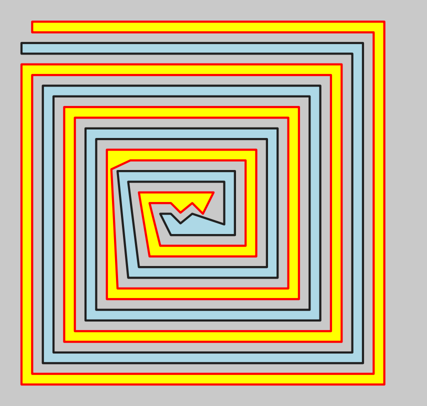
For disjoint polygons



- m disjoint polygons. n: total complexity of polygons.
- Want:
  - Decomposition into $O(m)$ cells.
  - Cell def. by $O(1)$ polygons.
- No hope with vertical decomposition.
- Need a more topological approach.

20: Vertical decomposition for polygons

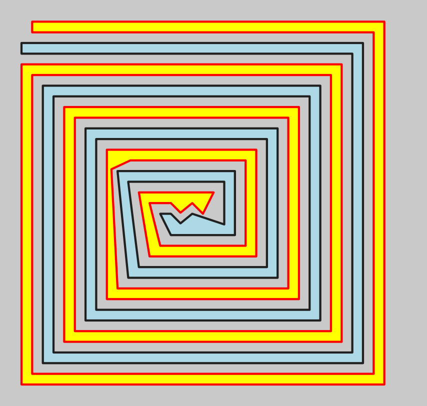
For disjoint polygons



1. m disjoint polygons. n: total complexity of polygons.
2. Want:
   1. Decomposition into $O(m)$ cells.
   2. Cell def. by $O(1)$ polygons.
3. No hope with vertical decomposition.
4. Need a more topological approach.

20: Vertical decomposition for polygons

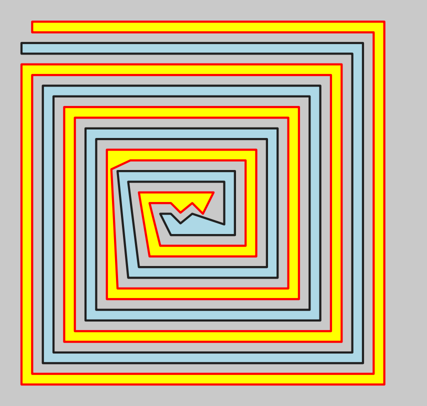
For disjoint polygons



- m disjoint polygons. n: total complexity of polygons.
- Want:
  - Decomposition into $O(m)$ cells.
  - Cell def. by $O(1)$ polygons.
- No hope with vertical decomposition.
- Need a more topological approach.

20: Vertical decomposition for polygons

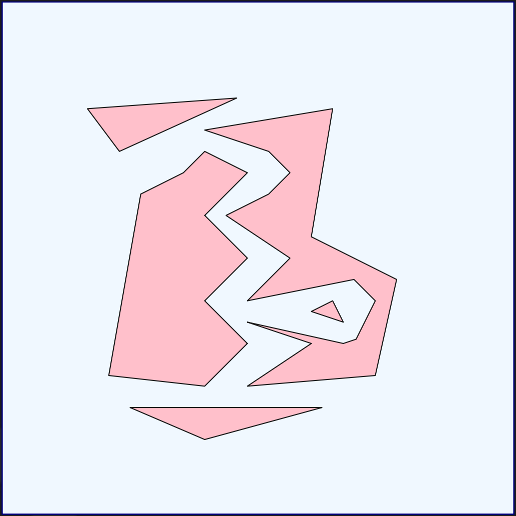
For disjoint polygons



1. m disjoint polygons. n: total complexity of polygons.
2. Want:
   1. Decomposition into $O(m)$ cells.
   2. Cell def. by $O(1)$ polygons.
3. No hope with vertical decomposition.
4. Need a more topological approach.

21: Corridor decomposition

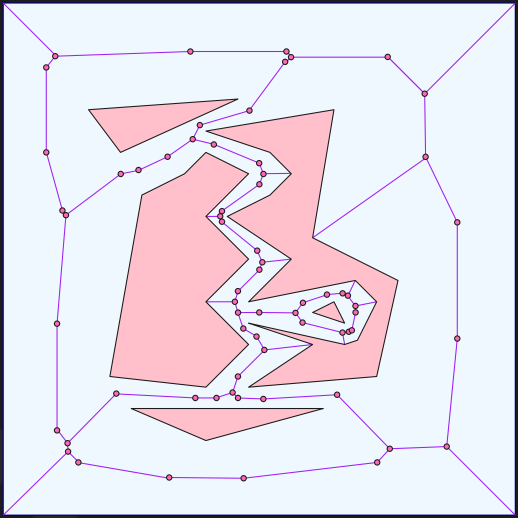
Breaking into cells for polygons



- L_∞ medial axis. Deformation retract.
- Critical squares.
- Tendrils...
- ...remove them
- deg 3 Vertices...
- ...critical squares
- ...add spokes.
- ...broke comp. into **corridors**.
  - Def. by 4 polygons.
  - $O(m)$ corridors. m: # polygons.

21: Corridor decomposition

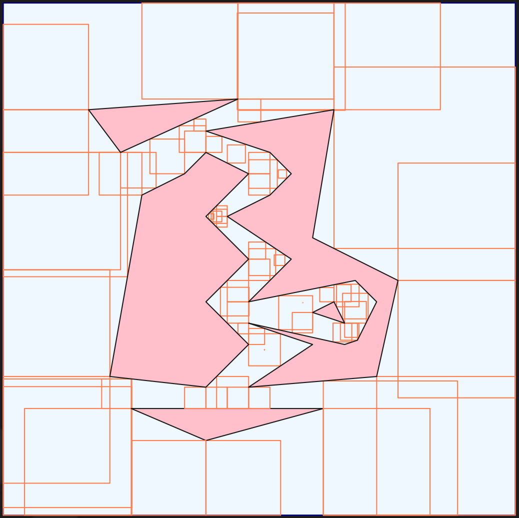
Breaking into cells for polygons



- L_∞ medial axis. Deformation retract.
- Critical squares.
- Tendrils...
- ...remove them
- deg 3 Vertices...
- ...critical squares
- ...add spokes.
- ...broke comp. into **corridors**.
  - Def. by 4 polygons.
  - $O(m)$ corridors. m: # polygons.

21: Corridor decomposition

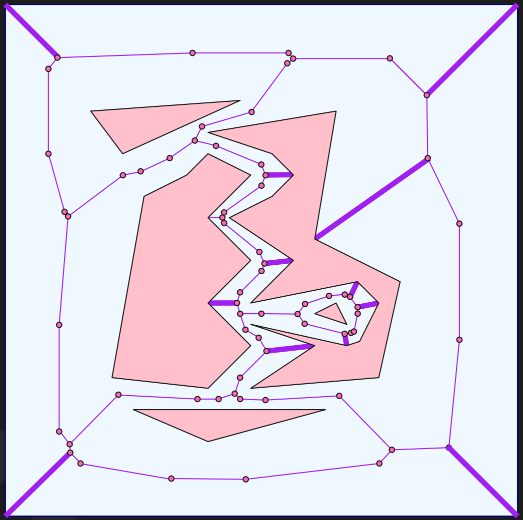
Breaking into cells for polygons



- L_∞ medial axis. Deformation retract.
- Critical squares.
- Tendrils...
- ...remove them
- deg 3 Vertices...
- ...critical squares
- ...add spokes.
- ...broke comp. into **corridors**.
  - Def. by 4 polygons.
  - $O(m)$ corridors. m: # polygons.

21: Corridor decomposition

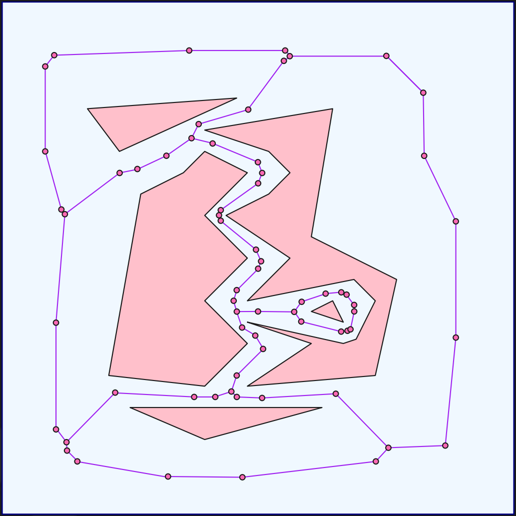
Breaking into cells for polygons



1. L_∞ medial axis. Deformation retract.
2. Critical squares.
3. Tendrils...
4. ...remove them
5. deg 3 Vertices...
6. ...critical squares
7. ...add spokes.
8. ...broke comp. into **corridors**.
   1. Def. by 4 polygons.
   2. $O(m)$ corridors. m: # polygons.

21: Corridor decomposition

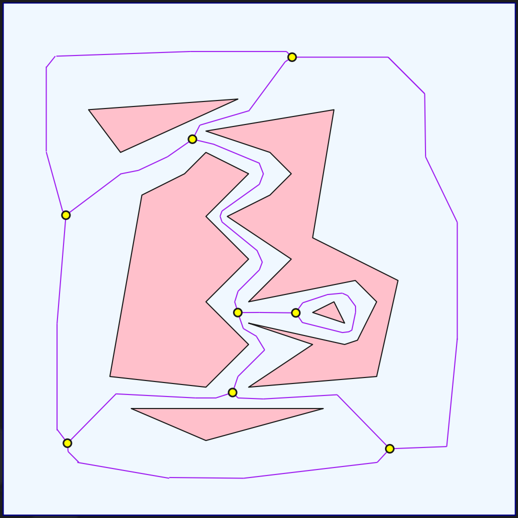
Breaking into cells for polygons



- L_∞ medial axis. Deformation retract.
- Critical squares.
- Tendrils...
- ...remove them
- deg 3 Vertices...
- ...critical squares
- ...add spokes.
- ...broke comp. into **corridors**.
  - Def. by 4 polygons.
  - $O(m)$ corridors. m: # polygons.

21: Corridor decomposition

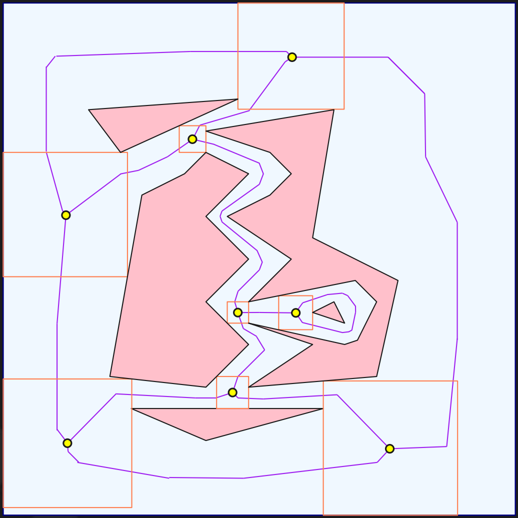
Breaking into cells for polygons



- L_∞ medial axis. Deformation retract.
- Critical squares.
- Tendrils...
- ...remove them
- deg 3 Vertices...
- ...critical squares
- ...add spokes.
- ...broke comp. into **corridors**.
  - Def. by 4 polygons.
  - $O(m)$ corridors. m: # polygons.

21: Corridor decomposition

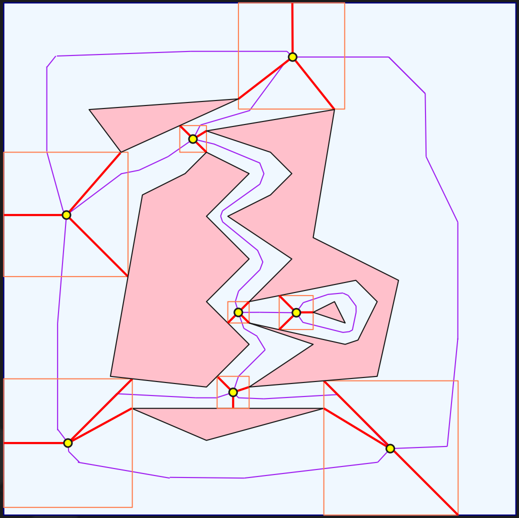
Breaking into cells for polygons



1. L_∞ medial axis. Deformation retract.
2. Critical squares.
3. Tendrils...
4. ...remove them
5. deg 3 Vertices...
6. ...critical squares
7. ...add spokes.
8. ...broke comp. into **corridors**.
   - Def. by 4 polygons.
   - $O(m)$ corridors. m: # polygons.

21: Corridor decomposition

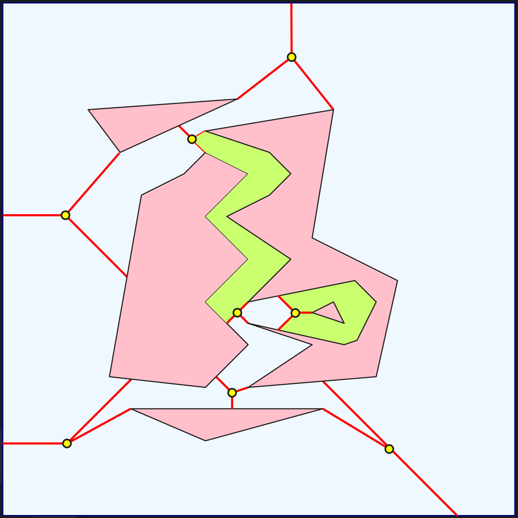
Breaking into cells for polygons



1. L_∞ medial axis. Deformation retract.
2. Critical squares.
3. Tendrils...
4. ...remove them
5. deg 3 Vertices...
6. ...critical squares
7. ...add spokes.
8. ...broke comp. into **corridors**.
   - Def. by 4 polygons.
   - $O(m)$ corridors. m: # polygons.

21: Corridor decomposition

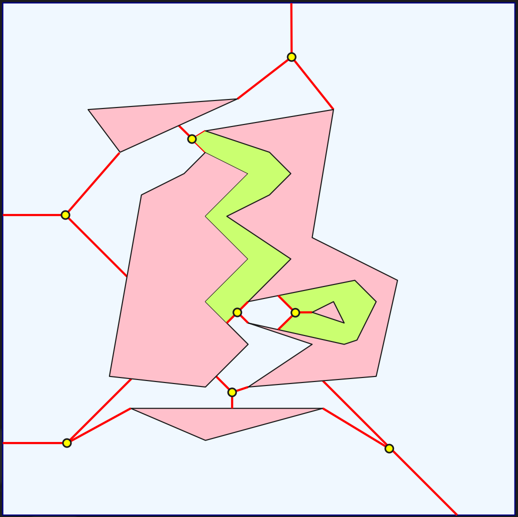
Breaking into cells for polygons



1. L_∞ medial axis. Deformation retract.
2. Critical squares.
3. Tendrils...
4. ...remove them
5. deg 3 Vertices...
6. ...critical squares
7. ...add spokes.
8. ...broke comp. into **corridors**.
   - Def. by 4 polygons.
   - $O(m)$ corridors. m: # polygons.

21: Corridor decomposition

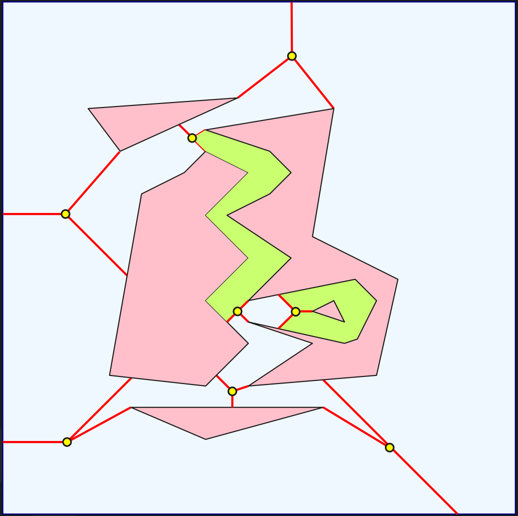
Breaking into cells for polygons



1. L_∞ medial axis. Deformation retract.
2. Critical squares.
3. Tendrils...
4. ...remove them
5. deg 3 Vertices...
6. ...critical squares
7. ...add spokes.
8. ...broke comp. into **corridors**.
   1. Def. by 4 polygons.
   2. $O(m)$ corridors. m: # polygons.

21: Corridor decomposition

Breaking into cells for polygons

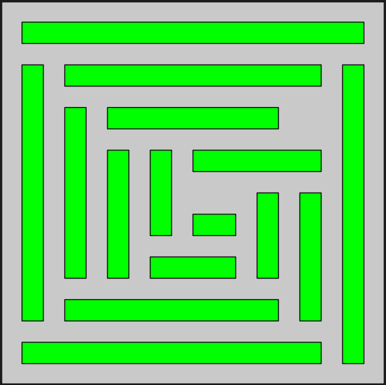


1. L_∞ medial axis. Deformation retract.
2. Critical squares.
3. Tendrils...
4. ...remove them
5. deg 3 Vertices...
6. ...critical squares
7. ...add spokes.
8. ...broke comp. into **corridors**.
   1. Def. by 4 polygons.
   2. $O(m)$ corridors. m: # polygons.

Deep in the technicalities zone

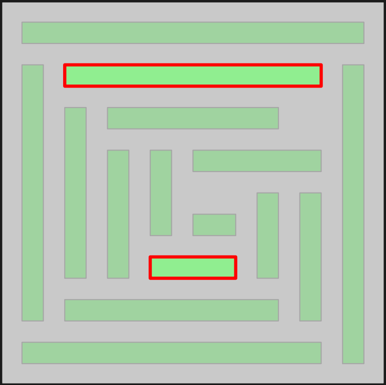
II: Cuttings

22: Cuttings



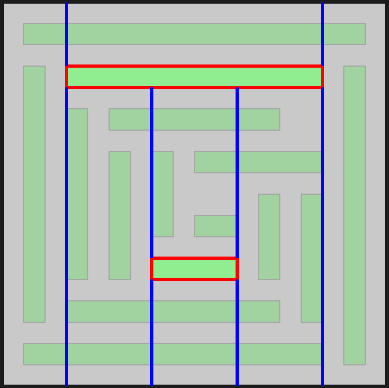
- $\mathcal{O}$: Set of n disjoint axis-parallel rectangles.
  r: Parameter
- **$1/r$-cutting**: Partition into cells, s.t. each cell intersects $\leq n/r$ rects $\mathcal{O}$.
- **[Chazelle and Friedman, 1990]**
  $1/r$-cutting with $O(r)$ cells.

22: Cuttings



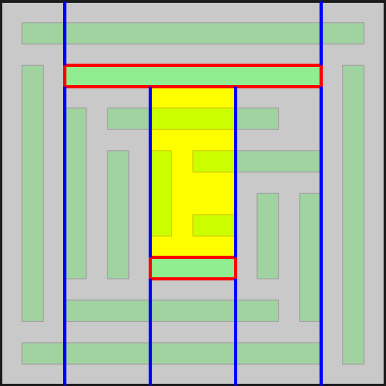
1. $\mathcal{O}$: Set of n disjoint axis-parallel rectangles.
   r: Parameter
2. **$1/r$-cutting**: Partition into cells, s.t. each cell intersects $\leq n/r$ rects $\mathcal{O}$.
3. **[Chazelle and Friedman, 1990]**
   $1/r$-cutting with $O(r)$ cells.

22: Cuttings



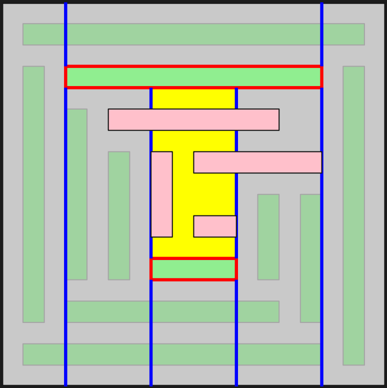
- $\mathcal{O}$: Set of n disjoint axis-parallel rectangles.
  r: Parameter
- **$1/r$-cutting**: Partition into cells, s.t. each cell intersects $\leq n/r$ rects $\mathcal{O}$.
- **[Chazelle and Friedman, 1990]**
  $1/r$-cutting with $O(r)$ cells.

22: Cuttings



- $\mathcal{O}$: Set of n disjoint axis-parallel rectangles.
  r: Parameter
- $1/r$-**cutting**: Partition into cells, s.t. each cell intersects $\leq n/r$ rects $\mathcal{O}$.
- **[Chazelle and Friedman, 1990]**
  $1/r$-cutting with $O(r)$ cells.

22: Cuttings



1. $\mathcal{O}$: Set of n disjoint axis-parallel rectangles. r: Parameter
2. $1/r$-**cutting**: Partition into cells, s.t. each cell intersects $\leq n/r$ rects $\mathcal{O}$.
3. **[Chazelle and Friedman, 1990]** $1/r$-cutting with $O(r)$ cells.

23: Weak cuttings for polygons

Lemma

$\mathcal{P}$: disjoint weighted polygons
W: total weight W
$\implies$ *compute $1/r$-cutting with $O(r \log r)$ corridors.*

- Proof is extension of previous proof.
- Might be slightly simpler.

23: Weak cuttings for polygons

Lemma

$\mathcal{P}$: *disjoint weighted polygons*
W: *total weight* W
$\implies$ *compute* $1/r$*-cutting with* $O(r \log r)$ *corridors.*

- Proof is extension of previous proof.
- Might be slightly simpler.

23: Weak cuttings for polygons

Lemma

$\mathcal{P}$: *disjoint weighted polygons*
W: *total weight* W
$\implies$ *compute* $1/r$*-cutting with* $O(r \log r)$ *corridors.*

1. Proof is extension of previous proof.
2. Might be slightly simpler.

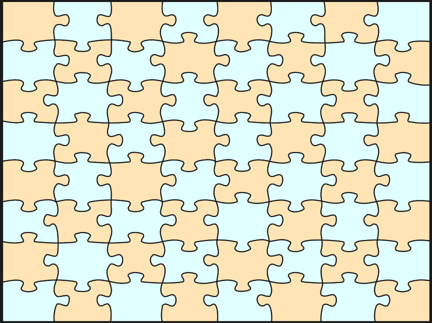
Part III

The algorithm

24: Recursing on optimal solution

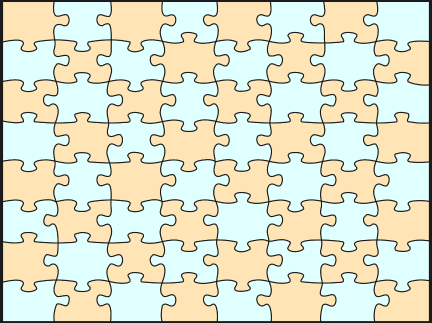
- $\mathcal{O}$: Opt sol $m = |\mathcal{O}|$, r.
- $1/r$-cutting, with $\widetilde{O}(r)$ cells.
- Tiles form a planar graph.
- tile intersects $\leq \frac{m}{r}$ polys $\in \mathcal{O}$.
- Cycle separator $\widetilde{O}(\sqrt{r})$ edges.
- Separator intersects $\widetilde{O}((m/r)\sqrt{r}) = \widetilde{O}(m/\sqrt{r})$.
- Recurse: In : Out

24: Recursing on optimal solution



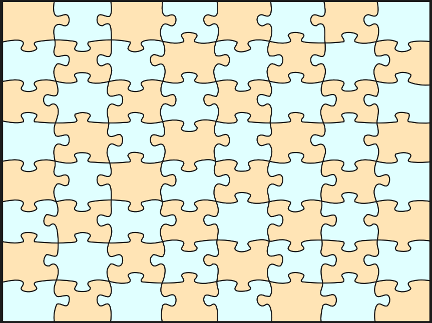
- $\mathcal{O}$: Opt sol $m = |\mathcal{O}|$, r.
- $1/r$-cutting, with $\widetilde{O}(r)$ cells.
- Tiles form a planar graph.
- tile intersects $\leq \frac{m}{r}$ polys $\in \mathcal{O}$.
- Cycle separator $\widetilde{O}(\sqrt{r})$ edges.
- Separator intersects $\widetilde{O}((m/r)\sqrt{r}) = \widetilde{O}(m/\sqrt{r})$.
- Recurse: In : Out

24: Recursing on optimal solution



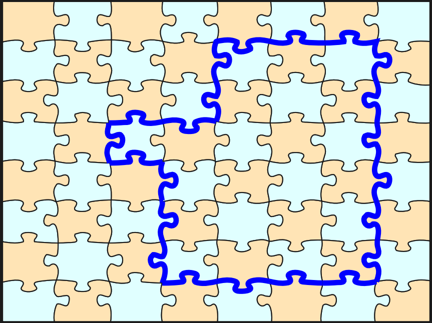
1. $\mathcal{O}$: Opt sol $m = |\mathcal{O}|$, r.
2. $1/r$-cutting, with $\widetilde{O}(r)$ cells.
3. Tiles form a planar graph.
4. tile intersects $\leq \frac{m}{r}$ polys $\in \mathcal{O}$.
5. Cycle separator $\widetilde{O}(\sqrt{r})$ edges.
6. Separator intersects $\widetilde{O}((m/r)\sqrt{r}) = \widetilde{O}(m/\sqrt{r})$.
7. Recurse: In : Out

24: Recursing on optimal solution



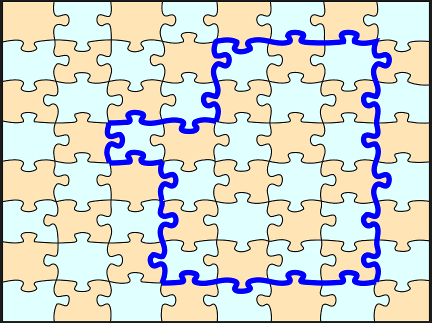
1. $\mathcal{O}$: Opt sol $m = |\mathcal{O}|$, r.
2. $1/r$-cutting, with $\widetilde{O}(r)$ cells.
3. Tiles form a planar graph.
4. tile intersects $\leq \frac{m}{r}$ polys $\in \mathcal{O}$.
5. Cycle separator $\widetilde{O}(\sqrt{r})$ edges.
6. Separator intersects $\widetilde{O}((m/r)\sqrt{r}) = \widetilde{O}(m/\sqrt{r})$.
7. Recurse: In : Out

24: Recursing on optimal solution



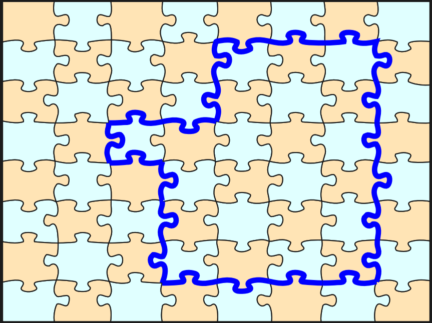
1. $\mathcal{O}$: Opt sol $m = |\mathcal{O}|$, r.
2. $1/r$-cutting, with $\widetilde{O}(r)$ cells.
3. Tiles form a planar graph.
4. tile intersects $\leq \frac{m}{r}$ polys $\in \mathcal{O}$.
5. Cycle separator $\widetilde{O}(\sqrt{r})$ edges.
6. Separator intersects $\widetilde{O}((m/r)\sqrt{r}) = \widetilde{O}(m/\sqrt{r})$.
7. Recurse: In : Out

24: Recursing on optimal solution



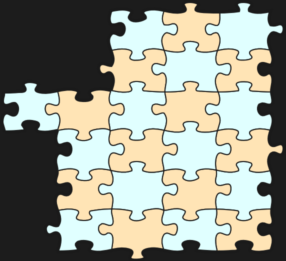
1. $\mathcal{O}$: Opt sol $m = |\mathcal{O}|$, r.
2. $1/r$-cutting, with $\widetilde{O}(r)$ cells.
3. Tiles form a planar graph.
4. tile intersects $\leq \frac{m}{r}$ polys $\in \mathcal{O}$.
5. Cycle separator $\widetilde{O}(\sqrt{r})$ edges.
6. Separator intersects $\widetilde{O}((m/r)\sqrt{r}) = \widetilde{O}(m/\sqrt{r})$.
7. Recurse: In : Out

24: Recursing on optimal solution



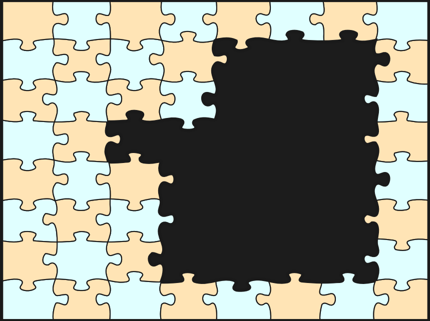
1. $\mathcal{O}$: Opt sol $m = |\mathcal{O}|$, r.
2. $1/r$-cutting, with $\widetilde{O}(r)$ cells.
3. Tiles form a planar graph.
4. tile intersects $\leq \frac{m}{r}$ polys $\in \mathcal{O}$.
5. Cycle separator $\widetilde{O}(\sqrt{r})$ edges.
6. Separator intersects $\widetilde{O}((m/r)\sqrt{r}) = \widetilde{O}(m/\sqrt{r})$.
7. Recurse: In : Out

24: Recursing on optimal solution



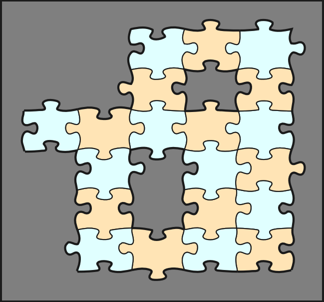
1. $\mathcal{O}$: Opt sol $m = |\mathcal{O}|$, r.
2. $1/r$-cutting, with $\widetilde{O}(r)$ cells.
3. Tiles form a planar graph.
4. tile intersects $\leq \frac{m}{r}$ polys $\in \mathcal{O}$.
5. Cycle separator $\widetilde{O}(\sqrt{r})$ edges.
6. Separator intersects $\widetilde{O}((m/r)\sqrt{r}) = \widetilde{O}(m/\sqrt{r})$.
7. Recurse: In : Out

24: Recursing on optimal solution



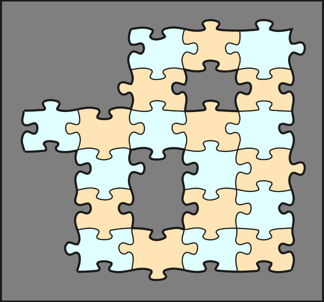
1. $\mathcal{O}$: Opt sol $m = |\mathcal{O}|$, r.
2. $1/r$-cutting, with $\widetilde{O}(r)$ cells.
3. Tiles form a planar graph.
4. tile intersects $\leq \frac{m}{r}$ polys $\in \mathcal{O}$.
5. Cycle separator $\widetilde{O}(\sqrt{r})$ edges.
6. Separator intersects $\widetilde{O}((m/r)\sqrt{r}) = \widetilde{O}(m/\sqrt{r})$.
7. Recurse: In : Out

25: How a subproblem looks like



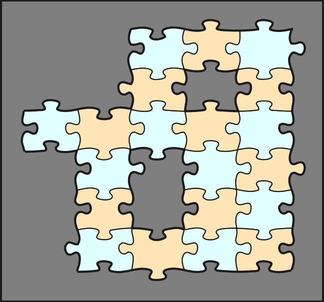
1. $\mathcal{O}$: Opt sol $|\mathcal{O}| \leq m$.
2. Subproblem defined by $O(\log m)$ cycles.
3. Each cycle has $\widetilde{O}(\sqrt{r})$ "corridor" edges.

26: Guessing the separator



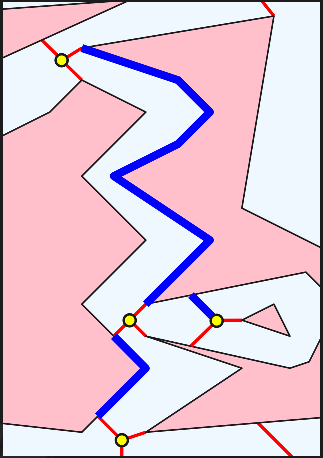
- $\mathcal{O}$: Optimal solution
  r: Param (specified shortly).
- $m = |\mathcal{P}|$.
- $m^{O(1)}$ possible edges. (cuttings+corridors decomposition)
- Separator has $\widetilde{O}(\sqrt{r})$ edges.
- $m^{\widetilde{O}(\sqrt{r})}$ different subproblems in the root.
- Depth of recursion $O(\log m)$.
- $m^{\widetilde{O}(\sqrt{r}\log m)}$ different subproblems overall.

26: Guessing the separator



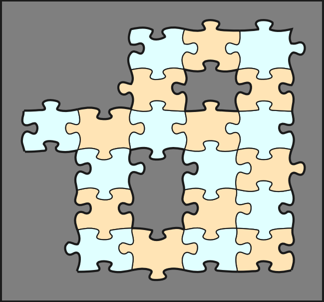
1. $\mathcal{O}$: Optimal solution
   r: Param (specified shortly).
2. $m = |\mathcal{P}|$.
3. $m^{O(1)}$ possible edges. (cuttings+corridors decomposition)
4. Separator has $\widetilde{O}(\sqrt{r})$ edges.
5. $m^{\widetilde{O}(\sqrt{r})}$ different subproblems in the root.
6. Depth of recursion $O(\log m)$.
7. $m^{\widetilde{O}(\sqrt{r}\log m)}$ different subproblems overall.

26: Guessing the separator



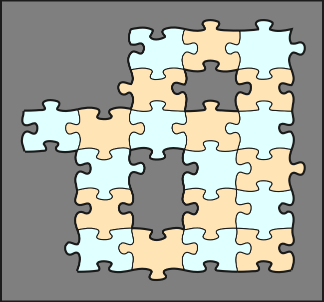
- $\mathcal{O}$: Optimal solution
  r: Param (specified shortly).
- $m = |\mathcal{P}|$.
- $m^{O(1)}$ possible edges. (cuttings+corridors decomposition)
- Separator has $\widetilde{O}(\sqrt{r})$ edges.
- $m^{\widetilde{O}(\sqrt{r})}$ different subproblems in the root.
- Depth of recursion $O(\log m)$.
- $m^{\widetilde{O}(\sqrt{r}\log m)}$ different subproblems overall.

26: Guessing the separator



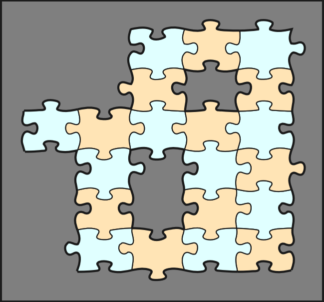
- $\mathcal{O}$: Optimal solution
  r: Param (specified shortly).
- $m = |\mathcal{P}|$.
- $m^{O(1)}$ possible edges. (cuttings+corridors decomposition)
- Separator has $\widetilde{O}(\sqrt{r})$ edges.
- $m^{\widetilde{O}(\sqrt{r})}$ different subproblems in the root.
- Depth of recursion $O(\log m)$.
- $m^{\widetilde{O}(\sqrt{r} \log m)}$ different subproblems overall.

26: Guessing the separator



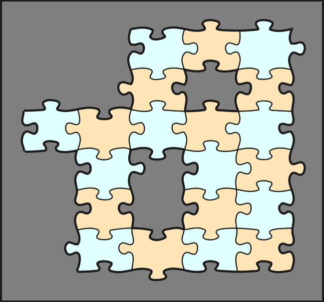
1. $\mathcal{O}$: Optimal solution
   r: Param (specified shortly).
2. $m = |\mathcal{P}|$.
3. $m^{O(1)}$ possible edges. (cuttings+corridors decomposition)
4. Separator has $\widetilde{O}(\sqrt{r})$ edges.
5. $m^{\widetilde{O}(\sqrt{r})}$ different subproblems in the root.
6. Depth of recursion $O(\log m)$.
7. $m^{\widetilde{O}(\sqrt{r}\log m)}$ different subproblems overall.

26: Guessing the separator



1. $\mathcal{O}$: Optimal solution
   r: Param (specified shortly).
2. $m = |\mathcal{P}|$.
3. $m^{O(1)}$ possible edges. (cuttings+corridors decomposition)
4. Separator has $\widetilde{O}(\sqrt{r})$ edges.
5. $m^{\widetilde{O}(\sqrt{r})}$ different subproblems in the root.
6. Depth of recursion $O(\log m)$.
7. $m^{\widetilde{O}(\sqrt{r}\log m)}$ different subproblems overall.

26: Guessing the separator



1. $\mathcal{O}$: Optimal solution
   r: Param (specified shortly).
2. $m = |\mathcal{P}|$.
3. $m^{O(1)}$ possible edges. (cuttings+corridors decomposition)
4. Separator has $\widetilde{O}(\sqrt{r})$ edges.
5. $m^{\widetilde{O}(\sqrt{r})}$ different subproblems in the root.
6. Depth of recursion $O(\log m)$.
7. $m^{\widetilde{O}(\sqrt{r}\log m)}$ different subproblems overall.

27: What we lose in the recursion

1. $|\mathcal{O}| \leq m$.
2. Every level...
   Lose $c\sqrt{\frac{\log r}{r}}$ fraction opt sol.
3. $\leq \log m$: depth recursion.
4. Total loss: $c\sqrt{\frac{\log r}{r}} \log m \leq \varepsilon$.
5. $r = O\left(\frac{\log^3 m}{\varepsilon^2}\right) \implies (1-\varepsilon)$-approximation.
6. $m = |\mathcal{P}|$.
7. Running time $m^{\widetilde{O}(\sqrt{r} \log m)}$.
   $\implies$ Running time $m^{O\left(\left(\frac{\log m}{\varepsilon}\right)^3\right)}$ **QPTAS**.

27: What we lose in the recursion

1. $|\mathcal{O}| \leq m$.
2. Every level...
   Lose $c\sqrt{\frac{\log r}{r}}$ fraction opt sol.
3. $\leq \log m$: depth recursion.
4. Total loss: $c\sqrt{\frac{\log r}{r}} \log m \leq \varepsilon$.
5. $r = O\left(\frac{\log^3 m}{\varepsilon^2}\right) \implies (1-\varepsilon)$-approximation.
6. $m = |\mathcal{P}|$.
7. Running time $m^{\widetilde{O}(\sqrt{r}\log m)}$.
   $\implies$ Running time $m^{O\left(\left(\frac{\log m}{\varepsilon}\right)^3\right)}$ **QPTAS**.

27: What we lose in the recursion

1. $|\mathcal{O}| \leq m$.
2. Every level...
   Lose $c\sqrt{\frac{\log r}{r}}$ fraction opt sol.
3. $\leq \log m$: depth recursion.
4. Total loss: $c\sqrt{\frac{\log r}{r}} \log m \leq \varepsilon$.
5. $r = O\left(\frac{\log^3 m}{\varepsilon^2}\right) \implies (1-\varepsilon)$-approximation.
6. $m = |\mathcal{P}|$.
7. Running time $m^{\tilde{O}(\sqrt{r}\log m)}$.
   $\implies$ Running time $m^{O\left(\left(\frac{\log m}{\varepsilon}\right)^3\right)}$ **QPTAS**.

27: What we lose in the recursion

1. $|\mathcal{O}| \leq m$.
2. Every level...
   Lose $c\sqrt{\frac{\log r}{r}}$ fraction opt sol.
3. $\leq \log m$: depth recursion.
4. Total loss: $c\sqrt{\frac{\log r}{r}} \log m \leq \varepsilon$.
5. $r = O\left(\frac{\log^3 m}{\varepsilon^2}\right) \implies (1-\varepsilon)$-approximation.
6. $m = |\mathcal{P}|$.
7. Running time $m^{\widetilde{O}(\sqrt{r}\log m)}$.
   $\implies$ Running time $m^{O\left(\left(\frac{\log m}{\varepsilon}\right)^3\right)}$ **QPTAS**.

27: What we lose in the recursion

1. $|\mathcal{O}| \leq m$.
2. Every level...
   Lose $c\sqrt{\frac{\log r}{r}}$ fraction opt sol.
3. $\leq \log m$: depth recursion.
4. Total loss: $c\sqrt{\frac{\log r}{r}} \log m \leq \varepsilon$.
5. $r = O\left(\frac{\log^3 m}{\varepsilon^2}\right) \implies (1-\varepsilon)$-approximation.
6. $m = |\mathcal{P}|$.
7. Running time $m^{\tilde{O}(\sqrt{r}\log m)}$.
   $\implies$ Running time $m^{O\left(\left(\frac{\log m}{\varepsilon}\right)^3\right)}$ **QPTAS**.

27: What we lose in the recursion

1. $|\mathcal{O}| \leq m$.
2. Every level...
   Lose $c\sqrt{\frac{\log r}{r}}$ fraction opt sol.
3. $\leq \log m$: depth recursion.
4. Total loss: $c\sqrt{\frac{\log r}{r}} \log m \leq \varepsilon$.
5. $r = O\left(\frac{\log^3 m}{\varepsilon^2}\right) \implies (1-\varepsilon)$-approximation.
6. $m = |\mathcal{P}|$.
7. Running time $m^{\widetilde{O}(\sqrt{r}\log m)}$.

$\implies$ Running time $m^{O\left(\left(\frac{\log m}{\varepsilon}\right)^3\right)}$ **QPTAS**.

27: What we lose in the recursion

1. $|\mathcal{O}| \leq m$.
2. Every level...
   Lose $c\sqrt{\frac{\log r}{r}}$ fraction opt sol.
3. $\leq \log m$: depth recursion.
4. Total loss: $c\sqrt{\frac{\log r}{r}} \log m \leq \varepsilon$.
5. $r = O\left(\frac{\log^3 m}{\varepsilon^2}\right) \implies (1-\varepsilon)$-approximation.
6. $m = |\mathcal{P}|$.
7. Running time $m^{\widetilde{O}(\sqrt{r}\log m)}$.
   $\implies$ Running time $m^{O\left(\left(\frac{\log m}{\varepsilon}\right)^3\right)}$ **QPTAS**.

Part IV

Conclusions

28: Conclusions

1. Framework of [Adamaszek and Wiese, 2013, 2014].
   Insight: Using planar separator on cuttings.
2. In this paper:
   1. Extended framework...
   2. Showed **QPTAS** for independent set of polygons of arbitrary complexity.
   3. Showed techniques works for computing sparse subsets.
3. Open questions for future research:
   1. **PTAS**?
   2. Better running time.

A. Adamaszek and A. Wiese. Approximation schemes for maximum weight independent set of rectangles. In *Proc. 54th Annu. IEEE Sympos. Found. Comput. Sci.* (FOCS), pages 400–409, 2013.

A. Adamaszek and A. Wiese. A QPTAS for maximum weight indepdent set of polygons with polylogarithmic many vertices. In *Proc. 25th ACM-SIAM Sympos. Discrete Algs.* (SODA), pages 400–409, 2014.

P. K. Agarwal and N. H. Mustafa. Independent set of intersection graphs of convex objects in 2D. *Comput. Geom. Theory Appl.*, 34(2):83–95, 2006.

P. Chalermsook and J. Chuzhoy. Maximum independent set of rectangles. In *Proc. 20th ACM-SIAM Sympos. Discrete Algs.* (SODA), pages 892–901, 2009.

T. M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms*, 46(2): 178–189, 2003.

T. M. Chan and S. Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete Comput. Geom.*, 48:373–392, 2012.

B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10(3): 229–249, 1990.

T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM J. Comput.*, 34(6):1302–1323, 2005.

J. Fox and J. Pach. Separator theorems and turn-type results for planar intersection graphs. *Adv. Math.*, 219:1070–1080, 2008.

J. Fox and J. Pach. Computing the independence number of intersection graphs. In *Proc. 22nd ACM-SIAM Sympos. Discrete Algs.* (SODA), pages 1161–1165, 2011.

S. Har-Peled. A simple proof of the existence of a planar separator. *CoRR*, abs/1105.0103, 2013.

J. Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, pages 627–636, 1996.

P. Koebe. Kontaktprobleme der konformen Abbildung. *Ber. Verh. Sächs. Akademie der Wissenschaften Leipzig, Math.-Phys. Klasse*, 88:141–164, 1936.

R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. In *Proc. GI Conf. Theoret. Comput. Sci.*, 1977.

G. L. Miller. Finding small simple cycle separators for 2-connected planar graphs. *J. Comput. Sys. Sci.*, 32(3): 265–279, 1986.

G. L. Miller, S. H. Teng, W. P. Thurston, and S. A. Vavasis. Separators for sphere-packings and nearest neighbor graphs. *J. Assoc. Comput. Mach.*, 44(1):1–29, 1997.