Algorithmic aspects of the theory of Graph Minors

Ignasi Sau

LIRMM, Université de Montpellier, CNRS, France

ForWorC

UFC, Fortaleza, November 6-10, 2023

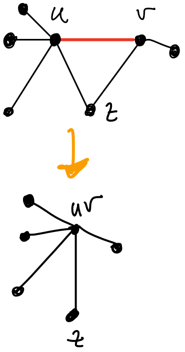
Outline of this mini-course

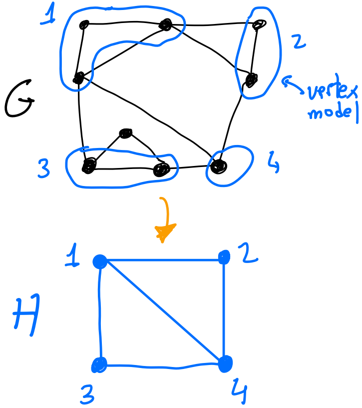
1. Introduction to graph minors
2. Bidimensionality
3. Irrelevant vertex technique

Next section is...

Graph minors

A graph H is a minor of a graph G, denoted by $H \leqslant_m G$, if H can be obtained by a subgraph of G by contracting edges.





Minor-closed graph classes

A graph class $\mathcal{C}$ is minor-closed (or closed under minors) if

$$G \in \mathcal{C} \;\; \Rightarrow \;\; H \in \mathcal{C} \text{ for every } H \leqslant_m G.$$

Minor-closed graph classes

A graph class $\mathcal{C}$ is minor-closed (or closed under minors) if

$$G \in \mathcal{C} \;\Rightarrow\; H \in \mathcal{C} \text{ for every } H \leqslant_m G.$$

Examples of minor-closed graph classes:

- Independent sets.

Minor-closed graph classes

A graph class $\mathcal{C}$ is minor-closed (or closed under minors) if

$$G \in \mathcal{C} \;\Rightarrow\; H \in \mathcal{C} \text{ for every } H \leqslant_m G.$$

Examples of minor-closed graph classes:

- Independent sets.
- Forests.

Minor-closed graph classes

A graph class $\mathcal{C}$ is minor-closed (or closed under minors) if

$$G \in \mathcal{C} \;\Rightarrow\; H \in \mathcal{C} \text{ for every } H \leqslant_m G.$$

Examples of minor-closed graph classes:

- Independent sets.
- Forests.
- Subgraphs of series-parallel graphs (why?).

Minor-closed graph classes

A graph class $\mathcal{C}$ is minor-closed (or closed under minors) if

$$G \in \mathcal{C} \;\Rightarrow\; H \in \mathcal{C} \text{ for every } H \leqslant_m G.$$

Examples of minor-closed graph classes:

- Independent sets.
- Forests.
- Subgraphs of series-parallel graphs (why?).
- Planar graphs (why?).

Minor-closed graph classes

A graph class $\mathcal{C}$ is minor-closed (or closed under minors) if

$$G \in \mathcal{C} \;\Rightarrow\; H \in \mathcal{C} \text{ for every } H \leqslant_m G.$$

Examples of minor-closed graph classes:

- Independent sets.
- Forests.
- Subgraphs of series-parallel graphs (why?).
- Planar graphs (why?).
- Graphs embeddable in a fixed surface.

Minor-closed graph classes

A graph class $\mathcal{C}$ is minor-closed (or closed under minors) if

$$G \in \mathcal{C} \;\Rightarrow\; H \in \mathcal{C} \text{ for every } H \leqslant_m G.$$

Examples of minor-closed graph classes:

- Independent sets.
- Forests.
- Subgraphs of series-parallel graphs (why?).
- Planar graphs (why?).
- Graphs embeddable in a fixed surface.
- Linklessly embeddable graphs.

Minor-closed graph classes

A graph class $\mathcal{C}$ is minor-closed (or closed under minors) if

$$G \in \mathcal{C} \;\Rightarrow\; H \in \mathcal{C} \text{ for every } H \leqslant_m G.$$

Examples of minor-closed graph classes:

- Independent sets.
- Forests.
- Subgraphs of series-parallel graphs (why?).
- Planar graphs (why?).
- Graphs embeddable in a fixed surface.
- Linklessly embeddable graphs.
- Knotlessly embeddable graphs.

Minor-closed graph classes

A graph class $\mathcal{C}$ is minor-closed (or closed under minors) if

$$G \in \mathcal{C} \;\Rightarrow\; H \in \mathcal{C} \text{ for every } H \leqslant_m G.$$

Examples of minor-closed graph classes:

- Independent sets.
- Forests.
- Subgraphs of series-parallel graphs (why?).
- Planar graphs (why?).
- Graphs embeddable in a fixed surface.
- Linklessly embeddable graphs.
- Knotlessly embeddable graphs.
- ...

Characterizing a graph class by excluded minors

Let $\mathcal{F}$ be a (possibly infinite) family of graphs. We define $\text{exc}(\mathcal{F})$ as the class of all graphs that do not contain any of the graphs in $\mathcal{F}$ as a minor.

Characterizing a graph class by excluded minors

Let $\mathcal{F}$ be a (possibly infinite) family of graphs. We define $\text{exc}(\mathcal{F})$ as the class of all graphs that do not contain any of the graphs in $\mathcal{F}$ as a minor.

Easy: for every family $\mathcal{F}$, the class $\text{exc}(\mathcal{F})$ is minor-closed (why?).

Characterizing a graph class by excluded minors

Let $\mathcal{F}$ be a (possibly infinite) family of graphs. We define $\text{exc}(\mathcal{F})$ as the class of all graphs that do not contain any of the graphs in $\mathcal{F}$ as a minor.

Easy: for every family $\mathcal{F}$, the class $\text{exc}(\mathcal{F})$ is minor-closed (why?).

We say that $\mathcal{F}$ characterizes $\text{exc}(\mathcal{F})$ by excluded minors.

Characterizing a graph class by excluded minors

Let $\mathcal{F}$ be a (possibly infinite) family of graphs. We define $\mathrm{exc}(\mathcal{F})$ as the class of all graphs that do not contain any of the graphs in $\mathcal{F}$ as a minor.

Easy: for every family $\mathcal{F}$, the class $\mathrm{exc}(\mathcal{F})$ is minor-closed (why?).

We say that $\mathcal{F}$ characterizes $\mathrm{exc}(\mathcal{F})$ by excluded minors.

Conversely, every minor-closed graph class $\mathcal{C}$ can be characterized by excluded minors:

> List all the graphs $\mathcal{F}_{\mathcal{C}} := \{G_1, G_2, \dots\}$ that do not belong to $\mathcal{C}$, and then $\mathcal{C} = \mathrm{exc}(\mathcal{F}_{\mathcal{C}})$.

Characterizing a graph class by excluded minors

Let $\mathcal{F}$ be a (possibly infinite) family of graphs. We define $\mathrm{exc}(\mathcal{F})$ as the class of all graphs that do not contain any of the graphs in $\mathcal{F}$ as a minor.

Easy: for every family $\mathcal{F}$, the class $\mathrm{exc}(\mathcal{F})$ is minor-closed (why?).

We say that $\mathcal{F}$ characterizes $\mathrm{exc}(\mathcal{F})$ by excluded minors.

Conversely, every minor-closed graph class $\mathcal{C}$ can be characterized by excluded minors:

> List all the graphs $\mathcal{F}_{\mathcal{C}} := \{G_1, G_2, \ldots\}$ that do not belong to $\mathcal{C}$, and then $\mathcal{C} = \mathrm{exc}(\mathcal{F}_{\mathcal{C}})$.

Note that, in general, this list $\mathcal{F}_{\mathcal{C}} = \{G_1, G_2, \ldots\}$ may be infinite.

Examples for some minor-closed classes

- If $\mathcal{C}$ = independent sets, then $\mathcal{C} =$

Examples for some minor-closed classes

- If $\mathcal{C}$ = independent sets, then $\mathcal{C}$ =exc(K_2).

Examples for some minor-closed classes

- If $\mathcal{C}$ = independent sets, then $\mathcal{C}$ =exc(K_2).
- If $\mathcal{C}$ = forests, then

Examples for some minor-closed classes

- If $\mathcal{C}$ = independent sets, then $\mathcal{C}$ =exc(K_2).
- If $\mathcal{C}$ = forests, then $\mathcal{C}$ = exc(K_3).

Examples for some minor-closed classes

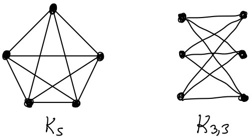
- If $\mathcal{C}$ = independent sets, then $\mathcal{C}$ =exc(K_2).
- If $\mathcal{C}$ = forests, then $\mathcal{C}$ = exc(K_3).
- If $\mathcal{C}$ = series-parallel graphs, then $\mathcal{C}$ = exc(K_4).

Examples for some minor-closed classes

- If $\mathcal{C}$ = independent sets, then $\mathcal{C}$ =exc(K_2).
- If $\mathcal{C}$ = forests, then $\mathcal{C}$ = exc(K_3).
- If $\mathcal{C}$ = series-parallel graphs, then $\mathcal{C}$ = exc(K_4).
- If $\mathcal{C}$ = outerplanar graphs, then $\mathcal{C}$ = exc$(K_4, K_{2,3})$.

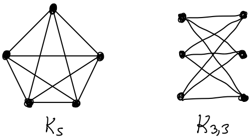
Examples for some minor-closed classes

- If $\mathcal{C}$ = independent sets, then $\mathcal{C}$ =exc(K_2).
- If $\mathcal{C}$ = forests, then $\mathcal{C}$ = exc(K_3).
- If $\mathcal{C}$ = series-parallel graphs, then $\mathcal{C}$ = exc(K_4).
- If $\mathcal{C}$ = outerplanar graphs, then $\mathcal{C}$ = exc$(K_4, K_{2,3})$.
- If $\mathcal{C}$ = planar graphs, then $\mathcal{C}$ = exc$(K_5, K_{3,3})$. [Kuratowski. 1930]



Examples for some minor-closed classes

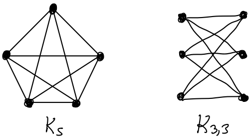
- If $\mathcal{C}$ = independent sets, then $\mathcal{C}$ =exc(K_2).
- If $\mathcal{C}$ = forests, then $\mathcal{C}$ = exc(K_3).
- If $\mathcal{C}$ = series-parallel graphs, then $\mathcal{C}$ = exc(K_4).
- If $\mathcal{C}$ = outerplanar graphs, then $\mathcal{C}$ = exc$(K_4, K_{2,3})$.
- If $\mathcal{C}$ = planar graphs, then $\mathcal{C}$ = exc$(K_5, K_{3,3})$. [Kuratowski. 1930]



- If $\mathcal{C}$ = graphs embeddable in the projective plane, then $|\mathcal{F}_{\mathcal{C}}| = 35$.

Examples for some minor-closed classes

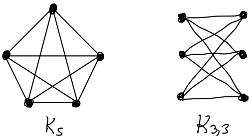
- If $\mathcal{C}$ = independent sets, then $\mathcal{C} = \mathrm{exc}(K_2)$.
- If $\mathcal{C}$ = forests, then $\mathcal{C} = \mathrm{exc}(K_3)$.
- If $\mathcal{C}$ = series-parallel graphs, then $\mathcal{C} = \mathrm{exc}(K_4)$.
- If $\mathcal{C}$ = outerplanar graphs, then $\mathcal{C} = \mathrm{exc}(K_4, K_{2,3})$.
- If $\mathcal{C}$ = planar graphs, then $\mathcal{C} = \mathrm{exc}(K_5, K_{3,3})$. [Kuratowski. 1930]



- If $\mathcal{C}$ = graphs embeddable in the projective plane, then $|\mathcal{F}_{\mathcal{C}}| = 35$.
- If $\mathcal{C}$ = graphs embeddable in a fixed non-orientable surface, then $\mathcal{F}_{\mathcal{C}}$ is finite. [Archdeacon, Huneke. 1989]

Examples for some minor-closed classes

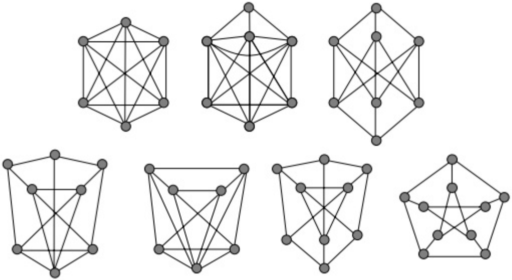
- If $\mathcal{C}$ = independent sets, then $\mathcal{C} = \text{exc}(K_2)$.
- If $\mathcal{C}$ = forests, then $\mathcal{C} = \text{exc}(K_3)$.
- If $\mathcal{C}$ = series-parallel graphs, then $\mathcal{C} = \text{exc}(K_4)$.
- If $\mathcal{C}$ = outerplanar graphs, then $\mathcal{C} = \text{exc}(K_4, K_{2,3})$.
- If $\mathcal{C}$ = planar graphs, then $\mathcal{C} = \text{exc}(K_5, K_{3,3})$. [Kuratowski. 1930]



- If $\mathcal{C}$ = graphs embeddable in the projective plane, then $|\mathcal{F}_\mathcal{C}| = 35$.
- If $\mathcal{C}$ = graphs embeddable in a fixed non-orientable surface, then $\mathcal{F}_\mathcal{C}$ is finite. [Archdeacon, Huneke. 1989]
- If $\mathcal{C}$ = graphs embeddable in a fixed orientable surface, then $\mathcal{F}_\mathcal{C}$ is finite. [Robertson, Seymour. 1990]

A last example

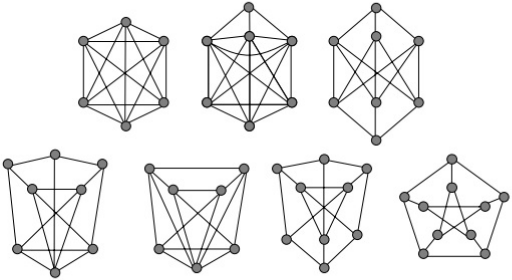
If $\mathcal{C}$ = linklessly embeddable graphs, then $\mathcal{F}_{\mathcal{C}} =$



[Robertson, Seymour. 1990]

A last example

If $\mathcal{C}$ = linklessly embeddable graphs, then $\mathcal{F}_{\mathcal{C}} =$



[Robertson, Seymour. 1990]

$\mathcal{F}_{\mathcal{C}}$ seems to get complicated... but always finite!

Wagner's conjecture

Conjecture (Wagner. 1970)

For every minor-closed graph class $\mathcal{C}$, there exists a finite set of graphs $\mathcal{F}_{\mathcal{C}}$ such that $\mathcal{C} = \mathrm{exc}(\mathcal{F}_{\mathcal{C}})$.

Wagner's conjecture... now Robertson-Seymour's theorem

Theorem (Robertson, Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, there exists a finite set of graphs $\mathcal{F}_{\mathcal{C}}$ such that $\mathcal{C} = \text{exc}(\mathcal{F}_{\mathcal{C}})$.

Reformulations

Theorem (Robertson, Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, there exists a finite set of graphs $\mathcal{F}_{\mathcal{C}}$ such that $\mathcal{C} = \mathrm{exc}(\mathcal{F}_{\mathcal{C}})$.

Reformulations

Theorem (Robertson, Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, there exists a finite set of graphs $\mathcal{F}_{\mathcal{C}}$ such that $\mathcal{C} = \text{exc}(\mathcal{F}_{\mathcal{C}})$.

Note that for every minor-closed graph class $\mathcal{C}$, the set of minor-minimal graphs not in $\mathcal{C}$ is unique (why?): it is denoted by $\text{obs}(\mathcal{C})$ (obstruction set).

Reformulations

Theorem (Robertson, Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, there exists a finite set of graphs $\mathcal{F}_{\mathcal{C}}$ such that $\mathcal{C} = \text{exc}(\mathcal{F}_{\mathcal{C}})$.

Note that for every minor-closed graph class $\mathcal{C}$, the set of minor-minimal graphs not in $\mathcal{C}$ is unique (why?): it is denoted by $\text{obs}(\mathcal{C})$ (obstruction set).

Equivalent: For every minor-closed graph class $\mathcal{C}$, $\text{obs}(\mathcal{C})$ is finite.

Reformulations

Theorem (Robertson, Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, there exists a finite set of graphs $\mathcal{F}_{\mathcal{C}}$ such that $\mathcal{C} = \text{exc}(\mathcal{F}_{\mathcal{C}})$.

Note that for every minor-closed graph class $\mathcal{C}$, the set of minor-minimal graphs not in $\mathcal{C}$ is unique (why?): it is denoted by $\text{obs}(\mathcal{C})$ (obstruction set).

Equivalent: For every minor-closed graph class $\mathcal{C}$, $\text{obs}(\mathcal{C})$ is finite.

Yet equivalent: Every infinite set $\{G_1, G_2, \ldots\}$ of finite graphs contains two graphs such that one is a minor of the other (there is no infinite antichain).

Well-quasi orders

A partially ordered set (poset) is a set P with a partial binary relation $\leq$:

1. Reflexive: $a \leq a$.
2. Antisymmetric: if $a \leq b$ and $b \leq a$, then $a = b$.
3. Transitive: if $a \leq b$ and $b \leq c$, then $a \leq c$.

Well-quasi orders

A partially ordered set (poset) is a set P with a partial binary relation $\leq$:

1. Reflexive: $a \leq a$.
2. Antisymmetric: if $a \leq b$ and $b \leq a$, then $a = b$.
3. Transitive: if $a \leq b$ and $b \leq c$, then $a \leq c$.

A poset $(P, \leq)$ is well-quasi-ordered (wqo) if every infinite sequence $(x_1, x_2, \ldots)$ has two elements x_i and x_j such that $i < j$ and $x_i \leq x_j$.

Well-quasi orders

A partially ordered set (poset) is a set P with a partial binary relation $\leq$:

1. Reflexive: $a \leq a$.
2. Antisymmetric: if $a \leq b$ and $b \leq a$, then $a = b$.
3. Transitive: if $a \leq b$ and $b \leq c$, then $a \leq c$.

A poset $(P, \leq)$ is well-quasi-ordered (wqo) if every infinite sequence $(x_1, x_2, \ldots)$ has two elements x_i and x_j such that $i < j$ and $x_i \leq x_j$.

Equivalent (why?): $(P, \leq)$ contains neither an infinite descending chain nor an infinite antichain (i.e., set of pairwise incomparable elements).

Well-quasi orders

A partially ordered set (poset) is a set P with a partial binary relation $\leq$:

1. Reflexive: $a \leq a$.
2. Antisymmetric: if $a \leq b$ and $b \leq a$, then $a = b$.
3. Transitive: if $a \leq b$ and $b \leq c$, then $a \leq c$.

A poset $(P, \leq)$ is well-quasi-ordered (wqo) if every infinite sequence $(x_1, x_2, \ldots)$ has two elements x_i and x_j such that $i < j$ and $x_i \leq x_j$.

Equivalent (why?): $(P, \leq)$ contains neither an infinite descending chain nor an infinite antichain (i.e., set of pairwise incomparable elements).

In the case of graph minors: there is no infinite descending chain (why?), so wqo $\Leftrightarrow$ no infinite antichain.

Well-quasi orders

A partially ordered set (poset) is a set P with a partial binary relation $\leq$:

1. Reflexive: $a \leq a$.
2. Antisymmetric: if $a \leq b$ and $b \leq a$, then $a = b$.
3. Transitive: if $a \leq b$ and $b \leq c$, then $a \leq c$.

A poset $(P, \leq)$ is well-quasi-ordered (wqo) if every infinite sequence $(x_1, x_2, \ldots)$ has two elements x_i and x_j such that $i < j$ and $x_i \leq x_j$.

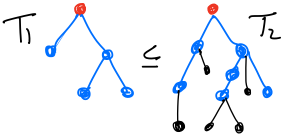
Equivalent (why?): $(P, \leq)$ contains neither an infinite descending chain nor an infinite antichain (i.e., set of pairwise incomparable elements).

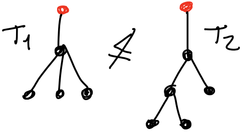
In the case of graph minors: there is no infinite descending chain (why?), so wqo $\Leftrightarrow$ no infinite antichain.

R&S theorem: Finite graphs are wqo with respect to the minor relation.

Illustrative example: rooted trees

Let T_1 and T_2 be two finite rooted trees.
Def: $T_1 \leq T_2$ if there is a subdivision of T_1 that occurs as a rooted subgraph of T_2 (the root of T_1 is not necessarily mapped to the root of T_2).

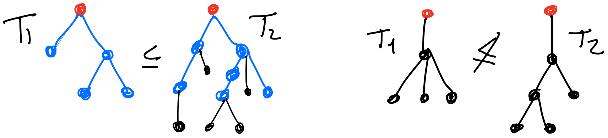




Illustrative example: rooted trees

Let T_1 and T_2 be two finite rooted trees.

Def: $T_1 \leq T_2$ if there is a subdivision of T_1 that occurs as a rooted subgraph of T_2 (the root of T_1 is not necessarily mapped to the root of T_2).



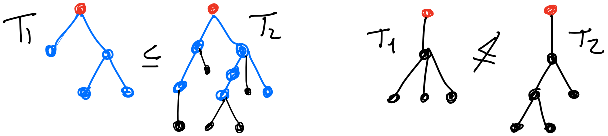
Conjecture (Vázsonyi. 1937)

Finite rooted trees are wqo with respect to the relation $\leq$.

Illustrative example: rooted trees

Let T_1 and T_2 be two finite rooted trees.

Def: $T_1 \leq T_2$ if there is a subdivision of T_1 that occurs as a rooted subgraph of T_2 (the root of T_1 is not necessarily mapped to the root of T_2).



Conjecture (Vázsonyi. 1937)

Finite rooted trees are wqo with respect to the relation $\leq$.

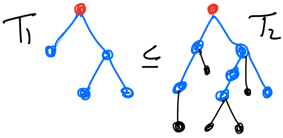
Proved independently by:

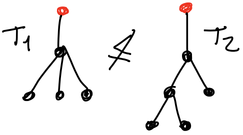
[Kruskal. 1960]

[Tarkowski. 1960]

Illustrative example: rooted trees

Let T_1 and T_2 be two finite rooted trees.
Def: $T_1 \leq T_2$ if there is a subdivision of T_1 that occurs as a rooted subgraph of T_2 (the root of T_1 is not necessarily mapped to the root of T_2).





Conjecture (Vázsonyi. 1937)

Finite rooted trees are wqo with respect to the relation $\leq$.

Proved independently by: [Kruskal. 1960]

[Tarkowski. 1960]

We will now see a simple proof by [Nash-Williams. 1963]

By contradiction, suppose that there is a bad infinite sequence:
$(T_1, T_2, \ldots)$ of rooted trees with no $i < j$ such that $T_i \leq T_j$.

By contradiction, suppose that there is a bad infinite sequence: $(T_1, T_2, \dots)$ of rooted trees with no $i < j$ such that $T_i \leq T_j$.

We choose the bad sequence in this particular way:

- Choose T_1 as a smallest tree that can start a bad sequence.

By contradiction, suppose that there is a bad infinite sequence: $(T_1, T_2, \ldots)$ of rooted trees with no $i < j$ such that $T_i \leq T_j$.

We choose the bad sequence in this particular way:

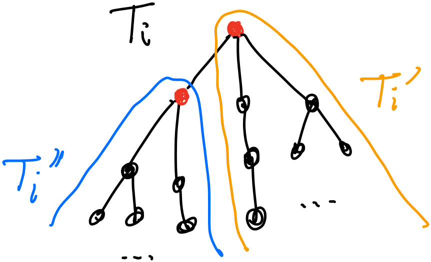
- Choose T_1 as a smallest tree that can start a bad sequence.
- For every $k > 1$, choose T_k as a smallest tree which occurs as the k-th element of a bad sequence starting with $(T_1, \ldots, T_{k-1})$.

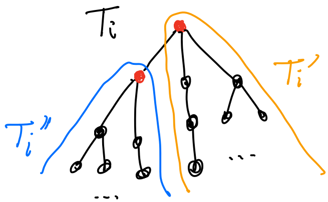
By contradiction, suppose that there is a bad infinite sequence: $(T_1, T_2, \ldots)$ of rooted trees with no $i < j$ such that $T_i \leq T_j$.

We choose the bad sequence in this particular way:

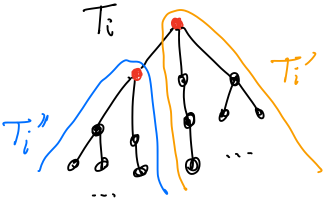
- Choose T_1 as a smallest tree that can start a bad sequence.
- For every $k > 1$, choose T_k as a smallest tree which occurs as the k-th element of a bad sequence starting with $(T_1, \ldots, T_{k-1})$.

For $k \geq 1$:
Let T_i' be the tree obtained from T_i by deleting any branch from the root.
Let T_i'' be the deleted branch (rooted at a child of the root of T_i).



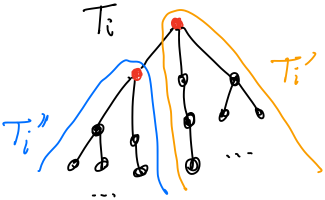


Claim: the sequence $(T_1', T_2', \ldots)$ cannot contain a bad subsequence.



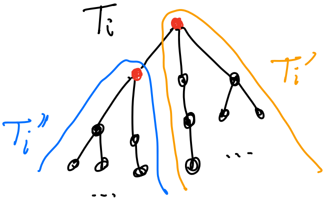
Claim: the sequence $(T_1', T_2', \dots)$ cannot contain a bad subsequence.

Proof: Suppose it does, and let $(T_{i_1}', T_{i_2}', \dots)$ be a bad subsequence.



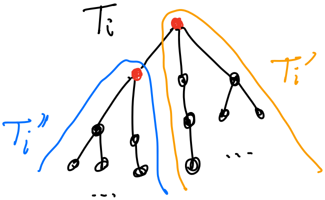
Claim: the sequence $(T_1', T_2', \ldots)$ cannot contain a bad subsequence.

Proof: Suppose it does, and let $(T_{i_1}', T_{i_2}', \ldots)$ be a bad subsequence. Then $(T_1, \ldots, T_{i_1-1}, T_{i_1}', T_{i_2}', \ldots)$ is bad



Claim: the sequence $(T_1', T_2', \ldots)$ cannot contain a bad subsequence.

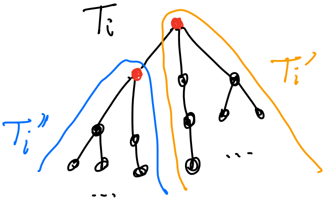
Proof: Suppose it does, and let $(T_{i_1}', T_{i_2}', \ldots)$ be a bad subsequence. Then $(T_1, \ldots, T_{i_1-1}, T_{i_1}', T_{i_2}', \ldots)$ is bad... but T_{i_1}' is smaller than T_{i_1}. $\square$



Claim: the sequence $(T_1', T_2', \ldots)$ cannot contain a bad subsequence.

Proof: Suppose it does, and let $(T_{i_1}', T_{i_2}', \ldots)$ be a bad subsequence. Then $(T_1, \ldots, T_{i_1 - 1}, T_{i_1}', T_{i_2}', \ldots)$ is bad... but T_{i_1}' is smaller than T_{i_1}. $\square$

It follows (why? hard! Uses Ramsey) that $(T_1', T_2', \ldots)$ contains an infinite increasing subsequence $T_{j_1}' \leq T_{j_2}' \leq \ldots$

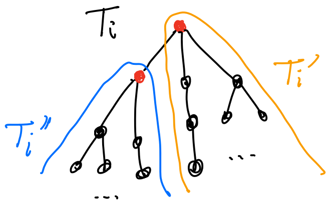


Claim: the sequence $(T_1', T_2', \ldots)$ cannot contain a bad subsequence.

Proof: Suppose it does, and let $(T_{i_1}', T_{i_2}', \ldots)$ be a bad subsequence. Then $(T_1, \ldots, T_{i_1-1}, T_{i_1}', T_{i_2}', \ldots)$ is bad... but T_{i_1}' is smaller than T_{i_1}. $\square$

It follows (why? hard! Uses Ramsey) that $(T_1', T_2', \ldots)$ contains an infinite increasing subsequence $T_{j_1}' \leq T_{j_2}' \leq \ldots$

Claim: the sequence $(T_{j_1}'', T_{j_2}'', \ldots)$ cannot be bad (why?).



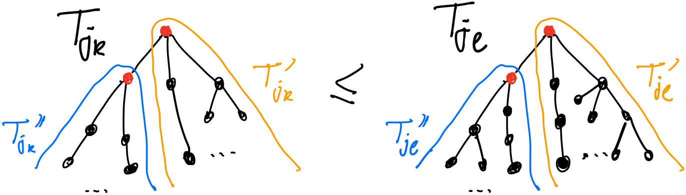
Claim: the sequence $(T'_1, T'_2, \ldots)$ cannot contain a bad subsequence.

Proof: Suppose it does, and let $(T'_{i_1}, T'_{i_2}, \ldots)$ be a bad subsequence. Then $(T_1, \ldots, T_{i_1-1}, T'_{i_1}, T'_{i_2}, \ldots)$ is bad... but T'_{i_1} is smaller than T_{i_1}. $\square$

It follows (why? hard! Uses Ramsey) that $(T'_1, T'_2, \ldots)$ contains an infinite increasing subsequence $T'_{j_1} \le T'_{j_2} \le \ldots$

Claim: the sequence $(T''_{j_1}, T''_{j_2}, \ldots)$ cannot be bad (why?).

There exist $k < \ell$ such that $T''_{j_k} \le T''_{j_\ell}$



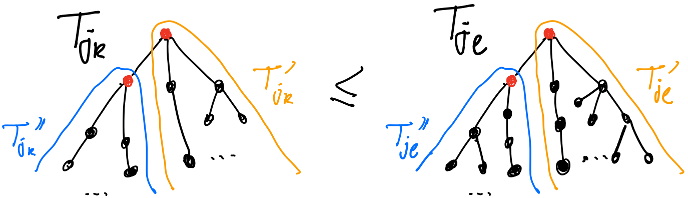
Claim: the sequence $(T'_1, T'_2, \ldots)$ cannot contain a bad subsequence.

Proof: Suppose it does, and let $(T'_{i_1}, T'_{i_2}, \ldots)$ be a bad subsequence. Then $(T_1, \ldots, T_{i_1-1}, T'_{i_1}, T'_{i_2}, \ldots)$ is bad... but T'_{i_1} is smaller than T_{i_1}. $\square$

It follows (why? hard! Uses Ramsey) that $(T'_1, T'_2, \ldots)$ contains an infinite increasing subsequence $T'_{j_1} \leq T'_{j_2} \leq \ldots$

Claim: the sequence $(T''_{j_1}, T''_{j_2}, \ldots)$ cannot be bad (why?).

There exist $k < \ell$ such that $T''_{j_k} \leq T''_{j_\ell}$



Claim: the sequence $(T'_1, T'_2, \ldots)$ cannot contain a bad subsequence.

Proof: Suppose it does, and let $(T'_{i_1}, T'_{i_2}, \ldots)$ be a bad subsequence. Then $(T_1, \ldots, T_{i_1-1}, T'_{i_1}, T'_{i_2}, \ldots)$ is bad... but T'_{i_1} is smaller than T_{i_1}. $\square$

It follows (why? hard! Uses Ramsey) that $(T'_1, T'_2, \ldots)$ contains an infinite increasing subsequence $T'_{j_1} \leq T'_{j_2} \leq \ldots$

Claim: the sequence $(T''_{j_1}, T''_{j_2}, \ldots)$ cannot be bad (why?).

There exist $k < \ell$ such that $T''_{j_k} \leq T''_{j_\ell} \;\Rightarrow\; T_{j_k} \leq T_{j_\ell}$, contradiction to bad!

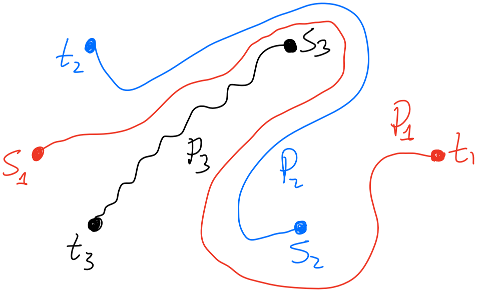
A notion strongly linked to graph minors

A notion strongly linked to graph minors

DISJOINT PATHS

Input: a graph G and $2k$ vertices $s_1, \ldots, s_k, t_1, \ldots, t_k$.

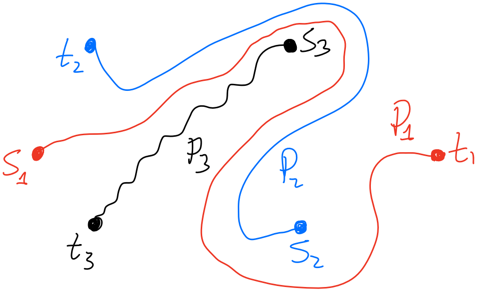
Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?



Disjoint Paths

Input: a graph G and $2k$ vertices $s_1, \dots, s_k, t_1, \dots, t_k$.

Question: does G contain k vertex-disjoint paths $P_1, \dots, P_k$ such that P_i connects s_i to t_i?



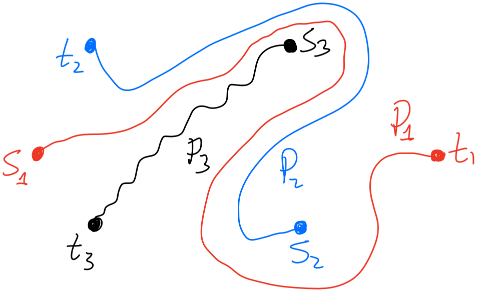
Much stronger than k vertex-disjoint paths from $s_1, \dots, s_k$ to $t_1, \dots, t_k$.

A notion strongly linked to graph minors

DISJOINT PATHS

Input: a graph G and $2k$ vertices $s_1, \ldots, s_k, t_1, \ldots, t_k$.

Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?



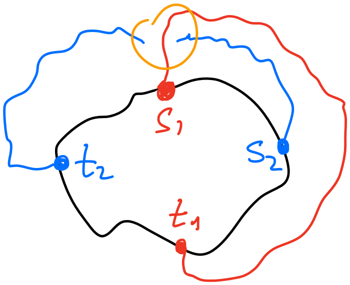
Much stronger than k vertex-disjoint paths from $s_1, \ldots, s_k$ to $t_1, \ldots, t_k$.

A graph G is k-linked if every instance of DISJOINT PATHS in G with k pairs is positive.

Topology appears naturally in linkages

Theorem (Thomassen and Seymour. 1980)

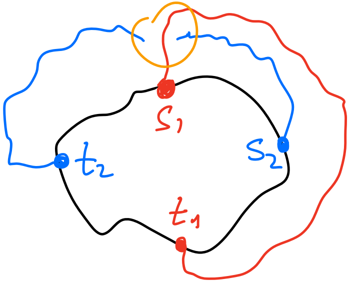
Let G be a 4-connected graph and $s_1, s_2, t_1, t_2 \in V(G)$. Then (s_1, s_2) and (t_1, t_2) are linked unless G is planar and s_1, s_2, t_1, t_2 are on the boundary of the same face, in this cyclic order.



Topology appears naturally in linkages

Theorem (Thomassen and Seymour. 1980)

Let G be a 4-connected graph and $s_1, s_2, t_1, t_2 \in V(G)$. Then (s_1, s_2) and (t_1, t_2) are linked unless G is planar and s_1, s_2, t_1, t_2 are on the boundary of the same face, in this cyclic order.



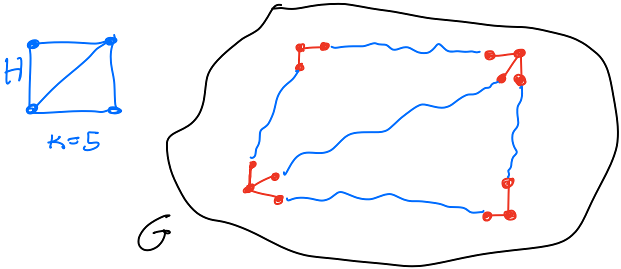
A combinatorial condition (linkage) is translated to a purely topological one (embedding).

Why linkages are useful for finding graph minors?

Let H be a graph with $|E(H)| = k$ and G be a k-linked graph.

Why linkages are useful for finding graph minors?

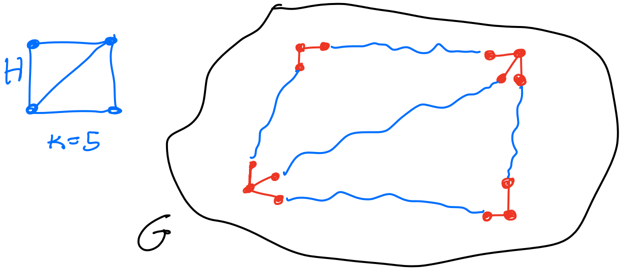
Let H be a graph with $|E(H)| = k$ and G be a k-linked graph.



Then we can easily find H as a minor in G!

Why linkages are useful for finding graph minors?

Let H be a graph with $|E(H)| = k$ and G be a k-linked graph.



Then we can easily find H as a minor in G!

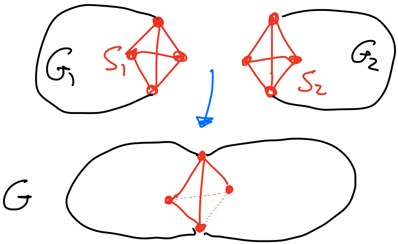
Idea: if the goal is to decide whether $H \leq_m G$, if G is k-linked, then "yes".
Otherwise, we may exploit a topological obstruction to k-linkedness...

Another crucial notion: treewidth

Let G_1 and G_2 be two graphs, and let $S_i \subseteq V(G_i)$ be a k-clique.

Another crucial notion: treewidth

Let G_1 and G_2 be two graphs, and let $S_i \subseteq V(G_i)$ be a k-clique.

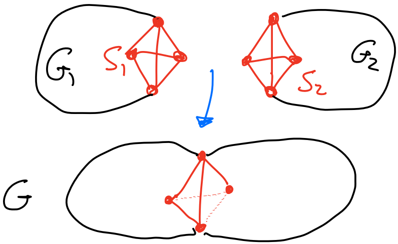


Let G be obtained by identifying S_1 with S_2 and deleting some (possibly none, possibly all) edges between the vertices in $S_1 = S_2$.

We say that G is a k-clique-sum of G_1 and G_2.

Another crucial notion: treewidth

Let G_1 and G_2 be two graphs, and let $S_i \subseteq V(G_i)$ be a k-clique.



Let G be obtained by identifying S_1 with S_2 and deleting some (possibly none, possibly all) edges between the vertices in $S_1 = S_2$.

We say that G is a k-clique-sum of G_1 and G_2.

We say that a graph G has treewidth at most k if it can be obtained by repeatedly taking a k-clique-sum with a graph on at most $k + 1$ vertices.

Structure of minor-free graphs

Let H be a fixed graph. Recall that $\text{exc}(H)$ is the class of all graphs that do not contain H as a minor.

Structure of minor-free graphs

Let H be a fixed graph. Recall that $\text{exc}(H)$ is the class of all graphs that do not contain H as a minor.

What is the typical structure of a graph $G \in \text{exc}(H)$?

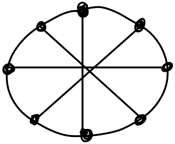
Structure of minor-free graphs

Let H be a fixed graph. Recall that $\mathrm{exc}(H)$ is the class of all graphs that do not contain H as a minor.

What is the typical structure of a graph $G \in \mathrm{exc}(H)$?

Theorem (Wagner. 1937)

A graph $G \in \mathrm{exc}(K_5)$ if and only if it can be obtained by 0-, 1-, 2- and 3-clique-sums from planar graphs and V_8.



V_8

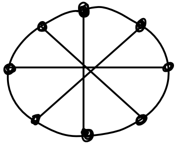
Structure of minor-free graphs

Let H be a fixed graph. Recall that $\mathrm{exc}(H)$ is the class of all graphs that do not contain H as a minor.

What is the typical structure of a graph $G \in \mathrm{exc}(H)$?

Theorem (Wagner. 1937)

A graph $G \in \mathrm{exc}(K_5)$ if and only if it can be obtained by 0-, 1-, 2- and 3-clique-sums from planar graphs and V_8.



V_8

Paradigm: we find "pieces" that exclude K_5 for topological reasons (planarity), add some exceptions (V_8), and then define rules (clique-sums) that preserve being K_5-minor-free.

An intermediate case: excluding a planar graph

Let H be a fixed planar graph.

What is the structure of a graph $G \in \mathrm{exc}(H)$?

An intermediate case: excluding a planar graph

Let H be a fixed planar graph.

What is the structure of a graph $G \in \text{exc}(H)$?

Theorem (Robertson, Seymour. 1986)

For every planar graph H there is an integer $t(H) > 0$ such that every graph in $\text{exc}(H)$ has treewidth at most $t(H)$.

An intermediate case: excluding a planar graph

Let H be a fixed planar graph.

What is the structure of a graph $G \in \mathrm{exc}(H)$?

Theorem (Robertson, Seymour. 1986)

For every planar graph H there is an integer $t(H) > 0$ such that every graph in $\mathrm{exc}(H)$ has treewidth at most $t(H)$.

Thus, every graph in $\mathrm{exc}(H)$ can be built by "gluing" bounded-sized graphs in a tree-like structure ($t(H)$-clique-sums).

An intermediate case: excluding a planar graph

Let H be a fixed planar graph.

What is the structure of a graph $G \in \mathrm{exc}(H)$?

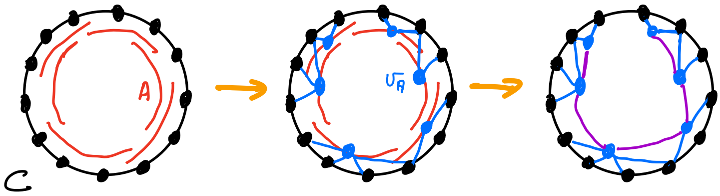
Theorem (Robertson, Seymour. 1986)

For every planar graph H there is an integer $t(H) > 0$ such that every graph in $\mathrm{exc}(H)$ has treewidth at most $t(H)$.

Thus, every graph in $\mathrm{exc}(H)$ can be built by "gluing" bounded-sized graphs in a tree-like structure ($t(H)$-clique-sums).

Note: this is an approximate characterization (i.e., not "iff").

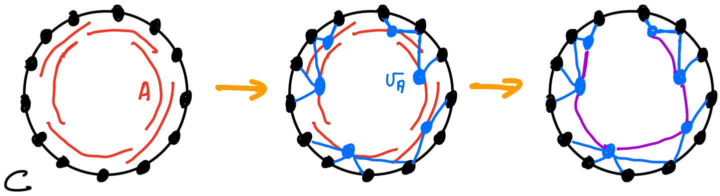
Vortices



Adding a vortex of depth h to a cycle C:

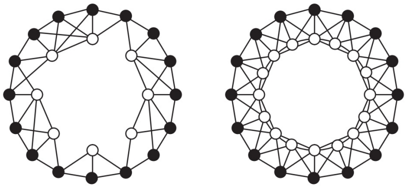
- Select arcs on C so that each vertex is contained in at most h arcs.
- For each arc A, create a vertex v_A.
- Connect v_A to some vertices on the arc A.
- connect any pair (v_A, v_B) for which A and B have a common vertex.

Vortices



Adding a vortex of depth h to a cycle C:

- Select arcs on C so that each vertex is contained in at most h arcs.
- For each arc A, create a vertex v_A.
- Connect v_A to some vertices on the arc A.
- connect any pair (v_A, v_B) for which A and B have a common vertex.



Structure theorem

Theorem (Robertson, Seymour. 1999)

For every graph H there is an integer $h > 0$ such that every graph in $\mathrm{exc}(H)$ can be (efficiently) constructed in the following way:

Structure theorem

Theorem (Robertson, Seymour. 1999)

For every graph H there is an integer $h > 0$ such that every graph in $\mathrm{exc}(H)$ can be (efficiently) constructed in the following way:

1. *Start with a graph G embedded in a connected closed surface Σ with genus at most h so that each face is homeomorphic with an open disc.*

Structure theorem

Theorem (Robertson, Seymour. 1999)

For every graph H there is an integer $h > 0$ such that every graph in $\mathrm{exc}(H)$ can be (efficiently) constructed in the following way:

1. *Start with a graph G embedded in a connected closed surface Σ with genus at most h so that each face is homeomorphic with an open disc.*
2. *Select at most h faces of G and add a vortex of depth at most h to each of them.*

Structure theorem

Theorem (Robertson, Seymour. 1999)

For every graph H there is an integer $h > 0$ such that every graph in $\mathrm{exc}(H)$ can be (efficiently) constructed in the following way:

1. *Start with a graph G embedded in a connected closed surface Σ with genus at most h so that each face is homeomorphic with an open disc.*
2. *Select at most h faces of G and add a vortex of depth at most h to each of them.*
3. *Create at most h new vertices (apices) and connect them to the other vertices arbitrarily.*

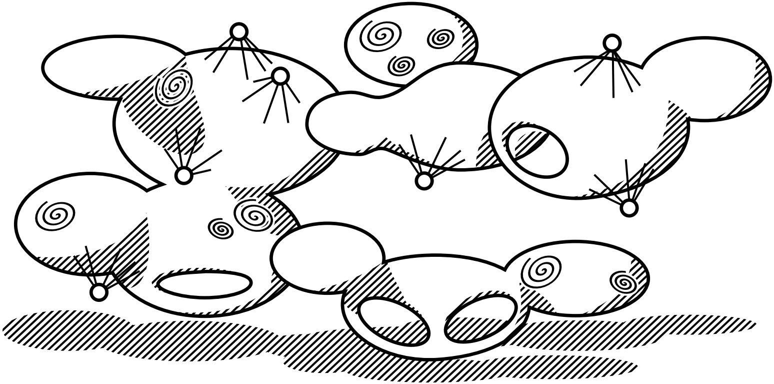
Structure theorem

Theorem (Robertson, Seymour. 1999)

For every graph H there is an integer $h > 0$ such that every graph in $\mathrm{exc}(H)$ can be (efficiently) constructed in the following way:

1. *Start with a graph G embedded in a connected closed surface Σ with genus at most h so that each face is homeomorphic with an open disc.*
2. *Select at most h faces of G and add a vortex of depth at most h to each of them.*
3. *Create at most h new vertices (apices) and connect them to the other vertices arbitrarily.*
4. *Repeatedly construct the h-clique-sum of the current graph with another graph constructed using steps 1-2-3 above.*

A visualization of an H-minor-free graph



[Figure by Felix Riedl]

Sketch of sketch of sketch of proof of Wagner's conjecture

Let's try to mimic the proof for rooted trees by Nash-Williams:

Sketch of sketch of sketch of proof of Wagner's conjecture

By contradiction, suppose that there is a bad infinite sequence: $(G_1, G_2, \ldots)$ of graphs with no $i < j$ such that $G_i \leq_{\mathrm{m}} G_j$.

Sketch of sketch of sketch of proof of Wagner's conjecture

By contradiction, suppose that there is a bad infinite sequence: $(G_1, G_2, \ldots)$ of graphs with no $i < j$ such that $G_i \leq_m G_j$.

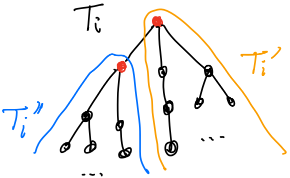
Again, choose $(G_1, G_2, \ldots)$ so that G_i is a minimal continuation.

Sketch of sketch of sketch of proof of Wagner's conjecture

By contradiction, suppose that there is a bad infinite sequence: $(G_1, G_2, \ldots)$ of graphs with no $i < j$ such that $G_i \leq_m G_j$.

Again, choose $(G_1, G_2, \ldots)$ so that G_i is a minimal continuation.

For trees, we decomposed each T_i into T_i' and T_i''... but now??

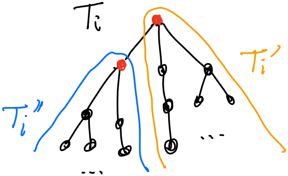


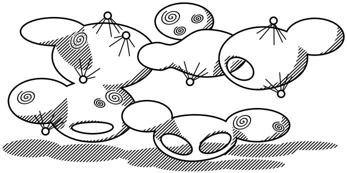
Sketch of sketch of sketch of proof of Wagner's conjecture

By contradiction, suppose that there is a bad infinite sequence: $(G_1, G_2, \ldots)$ of graphs with no $i < j$ such that $G_i \leq_{\mathrm{m}} G_j$.

Again, choose $(G_1, G_2, \ldots)$ so that G_i is a minimal continuation.

For trees, we decomposed each T_i into T_i' and T_i''... but now??





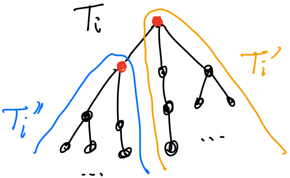
Every G_i with $i \geq 2$ is G_1-minor-free $\rightsquigarrow$ structure theorem of R&S!

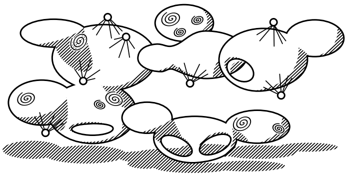
Sketch of sketch of sketch of proof of Wagner's conjecture

By contradiction, suppose that there is a bad infinite sequence:
$(G_1, G_2, \ldots)$ of graphs with no $i < j$ such that $G_i \leq_\mathrm{m} G_j$.

Again, choose $(G_1, G_2, \ldots)$ so that G_i is a minimal continuation.

For trees, we decomposed each T_i into T_i' and T_i''... but now??





Every G_i with $i \geq 2$ is G_1-minor-free $\rightsquigarrow$ structure theorem of R&S!

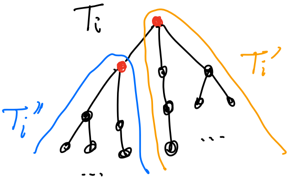
- If G_1 is planar, every G_i has bounded treewidth: similar to trees.

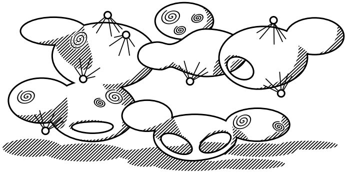
Sketch of sketch of sketch of proof of Wagner's conjecture

By contradiction, suppose that there is a bad infinite sequence: $(G_1, G_2, \ldots)$ of graphs with no $i < j$ such that $G_i \leq_{\mathrm{m}} G_j$.

Again, choose $(G_1, G_2, \ldots)$ so that G_i is a minimal continuation.

For trees, we decomposed each T_i into T_i' and T_i''... but now??





Every G_i with $i \geq 2$ is G_1-minor-free $\rightsquigarrow$ structure theorem of R&S!

- If G_1 is planar, every G_i has bounded treewidth: similar to trees.
- Otherwise, by the structure theorem: similar to "extended" surfaces (with apices and vortices), glued in a tree-like way.

Some algorithmic consequences

DISJOINT PATHS

Input: an n-vertex graph G and vertices $s_1, \ldots, s_k, t_1, \ldots, t_k$.

Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

Some algorithmic consequences

DISJOINT PATHS
Input: an n-vertex graph G and vertices $s_1, \ldots, s_k, t_1, \ldots, t_k$.
Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

Theorem (Robertson, Seymour. 1995)

The DISJOINT PATHS *problem can be solved in time* $f(k) \cdot n^3$.

Some algorithmic consequences

DISJOINT PATHS
Input: an n-vertex graph G and vertices $s_1, \ldots, s_k, t_1, \ldots, t_k$.
Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

Theorem (Robertson, Seymour. 1995)

The DISJOINT PATHS *problem can be solved in time* $f(k) \cdot n^3$.

Improved to $f(k) \cdot n^2$. [Kawarabayash, Kobayashi, Reed. 2012]

Some algorithmic consequences

DISJOINT PATHS
Input: an n-vertex graph G and vertices $s_1, \dots, s_k, t_1, \dots, t_k$.
Question: does G contain k vertex-disjoint paths $P_1, \dots, P_k$ such that P_i connects s_i to t_i?

Theorem (Robertson, Seymour. 1995)

The DISJOINT PATHS *problem can be solved in time* $f(k) \cdot n^3$.

Improved to $f(k) \cdot n^2$. [Kawarabayash, Kobayashi, Reed. 2012]

Corollary

For an n-vertex graph G and an h-vertex graph H, testing whether $H \leq_m G$ can be done in time $f(h) \cdot n^2$.

More algorithmic consequences

Corollary

For an n-vertex graph G and an h-vertex graph H, testing whether $H \leq_{\mathrm{m}} G$ can be done in time $f(h) \cdot n^2$.

More algorithmic consequences

Corollary

For an n-vertex graph G and an h-vertex graph H, testing whether $H \leq_{\mathrm{m}} G$ can be done in time $f(h) \cdot n^2$.

Recall:

Theorem (Robertson, Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, there exists a finite set of graphs $\mathcal{F}_{\mathcal{C}}$ such that $\mathcal{C} = \mathrm{exc}(\mathcal{F}_{\mathcal{C}})$.

More algorithmic consequences

Corollary

For an n-vertex graph G and an h-vertex graph H, testing whether $H \leq_{\mathrm{m}} G$ can be done in time $f(h) \cdot n^2$.

Recall:

Theorem (Robertson, Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, there exists a finite set of graphs $\mathcal{F}_{\mathcal{C}}$ such that $\mathcal{C} = \mathrm{exc}(\mathcal{F}_{\mathcal{C}})$.

Corollary

Every minor-closed property can be tested in quadratic time.

More algorithmic consequences

Corollary

For an n-vertex graph G and an h-vertex graph H, testing whether $H \leq_{\mathrm{m}} G$ can be done in time $f(h) \cdot n^2$.

Recall:

Theorem (Robertson, Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, there exists a finite set of graphs $\mathcal{F}_{\mathcal{C}}$ such that $\mathcal{C} = \mathrm{exc}(\mathcal{F}_{\mathcal{C}})$.

Corollary

Every minor-closed property can be tested in quadratic time.

Proof: check $H \leq_{\mathrm{m}} G$ for every graph H in the finite set $\mathcal{F}_{\mathcal{C}}$. $\square$

More algorithmic consequences

Corollary

For an n-vertex graph G and an h-vertex graph H, testing whether $H \leq_{\mathrm{m}} G$ can be done in time $f(h) \cdot n^2$.

Recall:

Theorem (Robertson, Seymour. 1983-2004)

For every minor-closed graph class $\mathcal{C}$, there exists a finite set of graphs $\mathcal{F}_{\mathcal{C}}$ such that $\mathcal{C} = \mathrm{exc}(\mathcal{F}_{\mathcal{C}})$.

Corollary

Every minor-closed property can be tested in quadratic time.

Proof: check $H \leq_{\mathrm{m}} G$ for every graph H in the finite set $\mathcal{F}_{\mathcal{C}}$. □

This says that there $\boxed{\text{exists}}$ an algorithm... no idea how to $\boxed{\text{construct}}$ it!!

A few words on other containment relations

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

A few words on other containment relations

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. MINOR TESTING is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation?

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leq 4$. [Brouwer and Veldman. 1987]

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leq 4$. [Brouwer and Veldman. 1987]
3. Nice structure?

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leq 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leq 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ **Topological minor:** $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree ≤ 2.

A few words on other containment relations

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leq 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ Topological minor: $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree ≤ 2.

1. Graphs are WQO w.r.t. the topological minor relation?

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leq 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ **Topological minor:** $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree ≤ 2.

1. Graphs are WQO w.r.t. the topological minor relation? NO! (why?)

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)
2. Contraction Minor Testing is FPT when param. by $|V(H)|$? NO! NP-hard already for $|V(H)| \leq 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ **Topological minor:** $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree ≤ 2.

1. Graphs are WQO w.r.t. the topological minor relation? NO! (why?)
2. Topological Minor Testing is FPT when param. by $|V(H)|$?

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leq 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ **Topological minor:** $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree ≤ 2.

1. Graphs are WQO w.r.t. the topological minor relation? NO! (why?)
2. Topological Minor Testing is FPT when param. by $|V(H)|$?
   YES! [Grohe, Kawarabayashi, Marx, Wollan. 2011]

A few words on other containment relations

★ Minor: $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ Contraction minor: $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leq 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ Topological minor: $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree ≤ 2.

1. Graphs are WQO w.r.t. the topological minor relation? NO! (why?)
2. Topological Minor Testing is FPT when param. by $|V(H)|$?
   YES! [Grohe, Kawarabayashi, Marx, Wollan. 2011]
3. Nice structure?

A few words on other containment relations

★ **Minor:** $H \preceq_m G$ if H can be obtained from a subgraph of G by contracting edges.

1. Graphs are WQO w.r.t. the minor relation.
2. Minor Testing is FPT when parameterized by $|V(H)|$.
3. H-minor-free graphs have a nice structure.

★ **Contraction minor:** $H \preceq_{cm} G$ if H can be obtained from G by contracting edges.

1. Graphs are WQO w.r.t. the contraction minor relation? NO! (why?)
2. Contraction Minor Testing is FPT when param. by $|V(H)|$?
   NO! NP-hard already for $|V(H)| \leq 4$. [Brouwer and Veldman. 1987]
3. Nice structure? Not really: They contain cliques, chordal graphs...

★ **Topological minor:** $H \preceq_{tp} G$ if H can be obtained from a subgraph of G by contracting edges with at least one endpoint of degree ≤ 2.

1. Graphs are WQO w.r.t. the topological minor relation? NO! (why?)
2. Topological Minor Testing is FPT when param. by $|V(H)|$?
   YES! [Grohe, Kawarabayashi, Marx, Wollan. 2011]
3. Nice structure? YES! [Grohe and Marx. 2012]

Structure of sparse graphs

H-topological-minor-free

$\cup$

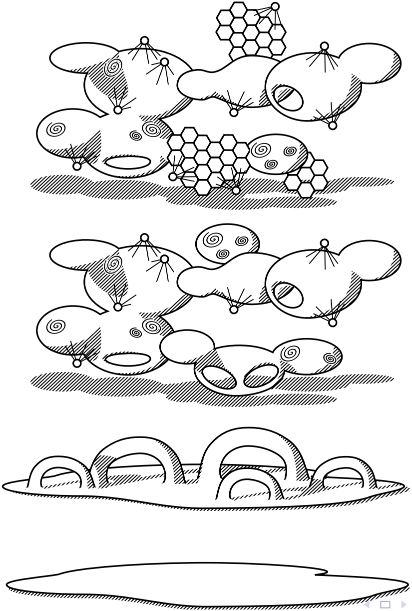
H-minor-free

$\cup$

bounded genus

$\cup$

planar



[Figure by Felix Riedl]

Next section is...

Next subsection is...

1. Introduction to graph minors
2. Bidimensionality
   - Preliminaries
   - Some ingredients and an illustrative example
   - Meta-algorithms
3. Irrelevant vertex technique

Parameterized complexity in 2 slides

A parameterized problem is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed, finite alphabet.

For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, k is called the parameter.

Parameterized complexity in 2 slides

A parameterized problem is a language $L \subseteq \Sigma^* \times \mathbb{N}$,
where Σ is a fixed, finite alphabet.

For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, k is called the parameter.

- k-Vertex Cover: Does a graph G contain a set $S \subseteq V(G)$, with $|S| \leq k$, containing at least an endpoint of every edge?
- k-Clique: Does a graph G contain a set $S \subseteq V(G)$, with $|S| \geq k$, of pairwise adjacent vertices?
- Vertex k-Coloring: Can the vertices of a graph be colored with $\leq k$ colors, so that any two adjacent vertices get different colors?

Parameterized complexity in 2 slides

A parameterized problem is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed, finite alphabet.

For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, k is called the parameter.

- k-Vertex Cover: Does a graph G contain a set $S \subseteq V(G)$, with $|S| \leq k$, containing at least an endpoint of every edge?
- k-Clique: Does a graph G contain a set $S \subseteq V(G)$, with $|S| \geq k$, of pairwise adjacent vertices?
- Vertex k-Coloring: Can the vertices of a graph be colored with $\leq k$ colors, so that any two adjacent vertices get different colors?

These three problems are NP-hard, but are they $\boxed{\text{equally}}$ hard?

They behave quite differently...

- k-Vertex Cover: Solvable in time $\mathcal{O}(2^k \cdot (m+n))$
- k-Clique: Solvable in time $\mathcal{O}(k^2 \cdot n^k)$
- Vertex k-Coloring: NP-hard for fixed $k = 3$.

They behave quite differently...

- k-Vertex Cover: Solvable in time $\mathcal{O}(2^k \cdot (m+n)) = f(k) \cdot n^{\mathcal{O}(1)}$.

- k-Clique: Solvable in time $\mathcal{O}(k^2 \cdot n^k) = f(k) \cdot n^{g(k)}$.

- Vertex k-Coloring: NP-hard for fixed $k = 3$.

They behave quite differently...

- k-VERTEX COVER: Solvable in time $\mathcal{O}(2^k \cdot (m+n)) = f(k) \cdot n^{\mathcal{O}(1)}$.

  $\boxed{\text{The problem is FPT}}$ (fixed-parameter tractable)

- k-CLIQUE: Solvable in time $\mathcal{O}(k^2 \cdot n^k) = f(k) \cdot n^{g(k)}$.

- VERTEX k-COLORING: NP-hard for fixed $k = 3$.

They behave quite differently...

- k-Vertex Cover: Solvable in time $\mathcal{O}(2^k \cdot (m+n)) = f(k) \cdot n^{\mathcal{O}(1)}$.

  $\boxed{\text{The problem is FPT}}$ (fixed-parameter tractable)

- k-Clique: Solvable in time $\mathcal{O}(k^2 \cdot n^k) = f(k) \cdot n^{g(k)}$.

  $\boxed{\text{The problem is XP}}$ (slice-wise polynomial)

- Vertex k-Coloring: NP-hard for fixed $k = 3$.

They behave quite differently...

- k-Vertex Cover: Solvable in time $\mathcal{O}(2^k \cdot (m+n)) = f(k) \cdot n^{\mathcal{O}(1)}$.

  $\boxed{\text{The problem is FPT}}$ (fixed-parameter tractable)

- k-Clique: Solvable in time $\mathcal{O}(k^2 \cdot n^k) = f(k) \cdot n^{g(k)}$.

  $\boxed{\text{The problem is XP}}$ (slice-wise polynomial)

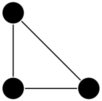
- Vertex k-Coloring: NP-hard for fixed $k = 3$.

  $\boxed{\text{The problem is para-NP-hard}}$

skip

Treewidth via k-trees

Example of a 2-tree:

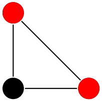


[Figure by Julien Baste]

For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

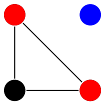
For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Example of a 2-tree:



[Figure by Julien Baste]

Example of a 2-tree:

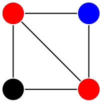


[Figure by Julien Baste]

For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:



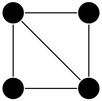
[Figure by Julien Baste]

For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

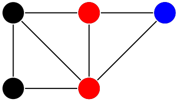
For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.



[Figure by Julien Baste]

Treewidth via k-trees

Example of a 2-tree:

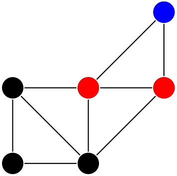


[Figure by Julien Baste]

For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

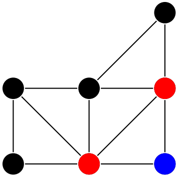


[Figure by Julien Baste]

For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

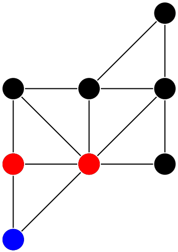


[Figure by Julien Baste]

For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

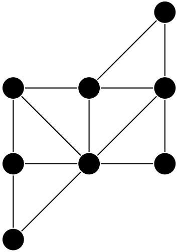


[Figure by Julien Baste]

For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:

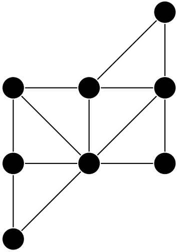


[Figure by Julien Baste]

For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

Treewidth via k-trees

Example of a 2-tree:



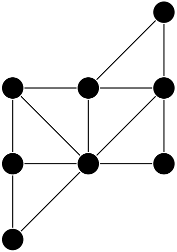
[Figure by Julien Baste]

For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

A partial k-tree is a subgraph of a k-tree.

Treewidth via k-trees

Example of a 2-tree:



[Figure by Julien Baste]

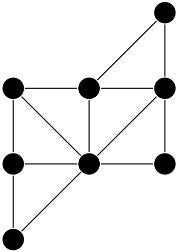
For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

A partial k-tree is a subgraph of a k-tree.

Treewidth of a graph G, denoted $\mathrm{tw}(G)$: smallest integer k such that G is a partial k-tree.

Treewidth via k-trees

Example of a 2-tree:



[Figure by Julien Baste]

For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

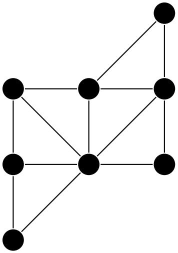
A partial k-tree is a subgraph of a k-tree.

Treewidth of a graph G, denoted $\mathrm{tw}(G)$: smallest integer k such that G is a partial k-tree.

Invariant that measures the topological resemblance of a graph to a forest.

Treewidth via k-trees

Example of a 2-tree:



[Figure by Julien Baste]

For $k \geq 1$, a k-tree is a graph that can be built starting from a $(k+1)$-clique and then iteratively adding a vertex connected to a k-clique.

A partial k-tree is a subgraph of a k-tree.

Treewidth of a graph G, denoted $\text{tw}(G)$: smallest integer k such that G is a partial k-tree.

Invariant that measures the topological resemblance of a graph to a forest.

Construction suggests the notion of tree decomposition: small separators.

Dynamic programming on tree decompositions

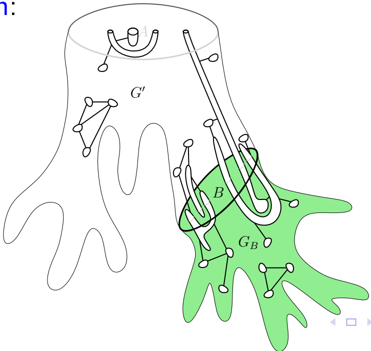
- Typically, FPT algorithms parameterized by treewidth are based on dynamic programming (DP) over a tree decomposition.

Dynamic programming on tree decompositions

- Typically, FPT algorithms parameterized by treewidth are based on dynamic programming (DP) over a tree decomposition.
- Starting from the leaves of the tree decomposition, a set of appropriately defined partial solutions is computed recursively until the root, where a global solution is obtained.

Dynamic programming on tree decompositions

- Typically, FPT algorithms parameterized by treewidth are based on dynamic programming (DP) over a tree decomposition.
- Starting from the leaves of the tree decomposition, a set of appropriately defined partial solutions is computed recursively until the root, where a global solution is obtained.
- The way that these partial solutions are defined depends on each particular problem:



[Figure by Valentin Garnero]

Monadic second order logic of graphs

We represent a graph $G = (V, E)$ with a structure $\mathcal{G} = (U, \text{vertex}, \text{edge}, I)$, where

Monadic second order logic of graphs

We represent a graph $G = (V, E)$ with a structure $\mathcal{G} = (U, \text{vertex}, \text{edge}, I)$, where

- $U = V \cup E$ is the universe.

Monadic second order logic of graphs

We represent a graph $G = (V, E)$ with a structure $\mathcal{G} = (U, \text{vertex}, \text{edge}, I)$, where

- $U = V \cup E$ is the universe.
- "vertex" and "edge" are unary relations that allow to distinguish vertices and edges.

Monadic second order logic of graphs

We represent a graph $G = (V, E)$ with a structure $\mathcal{G} = (U, \text{vertex}, \text{edge}, I)$, where

- $U = V \cup E$ is the universe.
- "vertex" and "edge" are unary relations that allow to distinguish vertices and edges.
- $I = \{(v, e) \mid v \in V, e \in E, v \in e\}$ is the incidence relation.

We represent a graph $G = (V, E)$ with a structure $\mathcal{G} = (U, \text{vertex}, \text{edge}, I)$, where

- $U = V \cup E$ is the universe.
- "vertex" and "edge" are unary relations that allow to distinguish vertices and edges.
- $I = \{(v, e) \mid v \in V, e \in E, v \in e\}$ is the incidence relation.

An MSO formula is built using the following:

Monadic second order logic of graphs

We represent a graph $G = (V, E)$ with a structure $\mathcal{G} = (U, \mathrm{vertex}, \mathrm{edge}, I)$, where

- $U = V \cup E$ is the universe.
- "vertex" and "edge" are unary relations that allow to distinguish vertices and edges.
- $I = \{(v, e) \mid v \in V, e \in E, v \in e\}$ is the incidence relation.

An MSO formula is built using the following:

- Logical connectors $\vee$, $\wedge$, $\Rightarrow$, $\neg$, $=$, $\neq$.

Monadic second order logic of graphs

We represent a graph $G = (V, E)$ with a structure $\mathcal{G} = (U, \text{vertex}, \text{edge}, I)$, where

- $U = V \cup E$ is the universe.
- "vertex" and "edge" are unary relations that allow to distinguish vertices and edges.
- $I = \{(v, e) \mid v \in V, e \in E, v \in e\}$ is the incidence relation.

An MSO formula is built using the following:

- Logical connectors $\vee$, $\wedge$, $\Rightarrow$, $\neg$, $=$, $\neq$.
- Predicates $\text{adj}(u, v)$ and $\text{inc}(e, v)$.

Monadic second order logic of graphs

We represent a graph $G = (V, E)$ with a structure $\mathcal{G} = (U, \text{vertex}, \text{edge}, I)$, where

- $U = V \cup E$ is the universe.
- "vertex" and "edge" are unary relations that allow to distinguish vertices and edges.
- $I = \{(v, e) \mid v \in V, e \in E, v \in e\}$ is the incidence relation.

An MSO formula is built using the following:

- Logical connectors $\vee$, $\wedge$, $\Rightarrow$, $\neg$, $=$, $\neq$.
- Predicates $\text{adj}(u, v)$ and $\text{inc}(e, v)$.
- Relations $\in$, $\subseteq$ on vertex/edge sets.

Monadic second order logic of graphs

We represent a graph $G = (V, E)$ with a structure $\mathcal{G} = (U, \text{vertex}, \text{edge}, I)$, where

- $U = V \cup E$ is the universe.
- "vertex" and "edge" are unary relations that allow to distinguish vertices and edges.
- $I = \{(v, e) \mid v \in V, e \in E, v \in e\}$ is the incidence relation.

An MSO formula is built using the following:

- Logical connectors $\vee$, $\wedge$, $\Rightarrow$, $\neg$, $=$, $\neq$.
- Predicates $\text{adj}(u, v)$ and $\text{inc}(e, v)$.
- Relations $\in$, $\subseteq$ on vertex/edge sets.
- Quantifiers $\exists$, $\forall$ on vertex/edge variables or vertex/edge **sets**.

($\text{MSO}_1/\text{MSO}_2$)

Monadic second order logic of graphs: examples

$\boxed{\text{Example 1}}$ Expressing that $\{u, v\} \in E(G)$: $\exists e \in E, \mathrm{inc}(u, e) \wedge \mathrm{inc}(v, e)$.

Monadic second order logic of graphs: examples

$\boxed{\text{Example 1}}$ Expressing that $\{u, v\} \in E(G)$: $\exists e \in E, \mathrm{inc}(u, e) \wedge \mathrm{inc}(v, e)$.

$\boxed{\text{Example 2}}$ Expressing that a set $S \subseteq V(G)$ is a dominating set.

$\texttt{DomSet}(S): \quad \forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G).$

Monadic second order logic of graphs: examples

$\boxed{\text{Example 1}}$ Expressing that $\{u, v\} \in E(G)$: $\quad \exists e \in E, \mathrm{inc}(u, e) \wedge \mathrm{inc}(v, e)$.

$\boxed{\text{Example 2}}$ Expressing that a set $S \subseteq V(G)$ is a dominating set.

$\texttt{DomSet}(S): \quad \forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G)$.

$\boxed{\text{Example 3}}$ Expressing that a graph $G = (V, E)$ is connected.

Monadic second order logic of graphs: examples

Example 1 Expressing that $\{u, v\} \in E(G)$: $\exists e \in E, \mathrm{inc}(u, e) \wedge \mathrm{inc}(v, e)$.

Example 2 Expressing that a set $S \subseteq V(G)$ is a dominating set.

$\mathtt{DomSet}(S): \quad \forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G)$.

Example 3 Expressing that a graph $G = (V, E)$ is connected.

- For every bipartition de V, there is a transversal edge:

Monadic second order logic of graphs: examples

$\boxed{\text{Example 1}}$ Expressing that $\{u, v\} \in E(G)$: $\quad \exists e \in E, \mathrm{inc}(u, e) \wedge \mathrm{inc}(v, e)$.

$\boxed{\text{Example 2}}$ Expressing that a set $S \subseteq V(G)$ is a dominating set.

$\texttt{DomSet}(S): \quad \forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G)$.

$\boxed{\text{Example 3}}$ Expressing that a graph $G = (V, E)$ is connected.

- For every bipartition de V, there is a transversal edge:

Expressing that two sets V_1, V_2 define a bipartition of V:

$\forall v \in V, (v \in V_1 \vee v \in V_2) \wedge (v \in V_1 \Rightarrow v \notin V_2) \wedge (v \in V_2 \Rightarrow v \notin V_1)$.

Monadic second order logic of graphs: examples

Example 1 Expressing that $\{u, v\} \in E(G)$: $\exists e \in E, \mathrm{inc}(u, e) \wedge \mathrm{inc}(v, e)$.

Example 2 Expressing that a set $S \subseteq V(G)$ is a dominating set.

$\texttt{DomSet}(S)$: $\forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G)$.

Example 3 Expressing that a graph $G = (V, E)$ is connected.

- For every bipartition de V, there is a transversal edge:

Expressing that two sets V_1, V_2 define a bipartition of V:

$\forall v \in V, (v \in V_1 \vee v \in V_2) \wedge (v \in V_1 \Rightarrow v \notin V_2) \wedge (v \in V_2 \Rightarrow v \notin V_1)$.

Connected: $\forall$ bipartition V_1, V_2, $\exists v_1 \in V_1, \exists v_2 \in V_2, \{v_1, v_2\} \in E(G)$.

Monadic second order logic of graphs: examples

$\boxed{\text{Example 1}}$ Expressing that $\{u, v\} \in E(G)$: $\exists e \in E, \text{inc}(u, e) \wedge \text{inc}(v, e)$.

$\boxed{\text{Example 2}}$ Expressing that a set $S \subseteq V(G)$ is a dominating set.

$\texttt{DomSet}(S)$: $\forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G)$.

$\boxed{\text{Example 3}}$ Expressing that a graph $G = (V, E)$ is connected.

- For every bipartition de V, there is a transversal edge:

Expressing that two sets V_1, V_2 define a bipartition of V:

$\forall v \in V, (v \in V_1 \vee v \in V_2) \wedge (v \in V_1 \Rightarrow v \notin V_2) \wedge (v \in V_2 \Rightarrow v \notin V_1)$.

Connected: $\forall$ bipartition V_1, V_2, $\exists v_1 \in V_1, \exists v_2 \in V_2, \{v_1, v_2\} \in E(G)$.

Other properties that can be expressed in MSO_2:

- a set being a vertex cover, independent set. (why?)

Monadic second order logic of graphs: examples

Example 1 Expressing that $\{u, v\} \in E(G)$: $\exists e \in E, \mathrm{inc}(u, e) \wedge \mathrm{inc}(v, e)$.

Example 2 Expressing that a set $S \subseteq V(G)$ is a dominating set.

$\mathtt{DomSet}(S)$: $\forall v \in V(G) \setminus S, \exists u \in S : \{u, v\} \in E(G)$.

Example 3 Expressing that a graph $G = (V, E)$ is connected.

- For every bipartition de V, there is a transversal edge:

Expressing that two sets V_1, V_2 define a bipartition of V:

$\forall v \in V, (v \in V_1 \vee v \in V_2) \wedge (v \in V_1 \Rightarrow v \notin V_2) \wedge (v \in V_2 \Rightarrow v \notin V_1)$.

Connected: $\forall$ bipartition V_1, V_2, $\exists v_1 \in V_1, \exists v_2 \in V_2, \{v_1, v_2\} \in E(G)$.

Other properties that can be expressed in MSO_2:

- a set being a vertex cover, independent set. (why?)
- a graph being k-colorable (for fixed k), having a Hamiltonian cycle.

Theorem (Courcelle. 1990)

Every problem expressible in MSO_2 *can be solved in time* $f(\mathrm{tw}) \cdot n$ *on graphs on* n *vertices and treewidth at most* tw.

Theorem (Courcelle. 1990)

Every problem expressible in MSO_2 *can be solved in time* $f(\mathrm{tw}) \cdot n$ *on graphs on* n *vertices and treewidth at most* tw.

The function $f(\mathrm{tw})$ depends on the structure of the MSO_2 formula.

Theorem (Courcelle. 1990)

Every problem expressible in MSO_2 *can be solved in time* $f(\mathrm{tw}) \cdot n$ *on graphs on* n *vertices and treewidth at most* tw.

The function $f(\mathrm{tw})$ depends on the structure of the MSO_2 formula.

Within the same running time, one can also optimize the size of a vertex/edge set satisfying an MSO_2 formula.

Theorem (Courcelle. 1990)

Every problem expressible in MSO_2 *can be solved in time* $f(\mathsf{tw}) \cdot n$ *on graphs on* n *vertices and treewidth at most* tw*.*

The function $f(\mathsf{tw})$ depends on the structure of the MSO_2 formula.

Within the same running time, one can also optimize the size of a vertex/edge set satisfying an MSO_2 formula.

Examples: Vertex Cover, Dominating Set, Hamiltonian Cycle, Clique, Independent Set, k-Coloring for fixed k, ...

Theorem (Courcelle. 1990)

Every problem expressible in MSO_2 *can be solved in time* $f(\mathrm{tw}) \cdot n$ *on graphs on* n *vertices and treewidth at most* tw.

The function $f(\mathrm{tw})$ depends on the structure of the MSO_2 formula.

Within the same running time, one can also optimize the size of a vertex/edge set satisfying an MSO_2 formula.

Examples: VERTEX COVER, DOMINATING SET, HAMILTONIAN CYCLE, CLIQUE, INDEPENDENT SET, k-COLORING for fixed k, ...

In parameterized complexity: FPT parameterized by treewidth.

Small parenthesis: only good news?

Theorem (Courcelle. 1990)

Every problem expressible in MSO_2 *can be solved in time* $f(\mathrm{tw}) \cdot n$ *on graphs on* n *vertices and treewidth at most* tw.

In parameterized complexity: FPT parameterized by treewidth.

Small parenthesis: only good news?

Theorem (Courcelle. 1990)

Every problem expressible in MSO_2 *can be solved in time* $f(\mathrm{tw}) \cdot n$ *on graphs on* n *vertices and treewidth at most* tw.

In parameterized complexity: FPT parameterized by treewidth.

1. Are all "natural" graph problems FPT parameterized by treewidth?

Small parenthesis: only good news?

Theorem (Courcelle. 1990)

Every problem expressible in MSO_2 *can be solved in time* $f(\mathrm{tw}) \cdot n$ *on graphs on* n *vertices and treewidth at most* tw.

In parameterized complexity: FPT parameterized by treewidth.

1. Are all "natural" graph problems FPT parameterized by treewidth?

   The vast majority, but not all of them:

   - List Coloring is W[1]-hard parameterized by treewidth.

     [Fellows, Fomin, Lokshtanov, Rosamond, Saurabh, Szeider, Thomassen. 2007]

Small parenthesis: only good news?

Theorem (Courcelle. 1990)

Every problem expressible in MSO_2 can be solved in time $f(\mathsf{tw}) \cdot n$ on graphs on n vertices and treewidth at most tw.

In parameterized complexity: FPT parameterized by treewidth.

1. Are all “natural” graph problems FPT parameterized by treewidth?

   The vast majority, but not all of them:

   - List Coloring is W[1]-hard parameterized by treewidth.

     [Fellows, Fomin, Lokshtanov, Rosamond, Saurabh, Szeider, Thomassen. 2007]

   - Some problems are even NP-hard on graphs of constant treewidth: Steiner Forest ($\mathsf{tw} = 3$), Bandwidth ($\mathsf{tw} = 1$).

Small parenthesis: only good news?

Theorem (Courcelle. 1990)

Every problem expressible in MSO_2 *can be solved in time* $f(\mathsf{tw}) \cdot n$ *on graphs on* n *vertices and treewidth at most* tw.

In parameterized complexity: FPT parameterized by treewidth.

1. Are all "natural" graph problems FPT parameterized by treewidth?

   The vast majority, but not all of them:

   - List Coloring is W[1]-hard parameterized by treewidth.
     [Fellows, Fomin, Lokshtanov, Rosamond, Saurabh, Szeider, Thomassen. 2007]
   - Some problems are even NP-hard on graphs of constant treewidth: Steiner Forest ($\mathsf{tw} = 3$), Bandwidth ($\mathsf{tw} = 1$).

2. Most natural problems (Vertex Cover, Dominating Set, ...) do not admit polynomial kernels parameterized by treewidth.

Next subsection is...

A few representative problems

VERTEX COVER
Input: A graph $G = (V, E)$ and a positive integer k.
Parameter: k.
Question: Does there exist a subset $C \subseteq V$ of size at most k such that $G[V \setminus C]$ is an independent set?

A few representative problems

Vertex Cover
Input: A graph $G = (V, E)$ and a positive integer k.
Parameter: k.
Question: Does there exist a subset $C \subseteq V$ of size at most k such that $G[V \setminus C]$ is an independent set?

Long Path
Input: A graph $G = (V, E)$ and a positive integer k.
Parameter: k.
Question: Does there exist a path P in G of length at least k?

A few representative problems (II)

Feedback Vertex Set
Input: A graph $G = (V, E)$ and a positive integer k.
Parameter: k.
Question: Does there exist a subset $F \subseteq V$ of size at most k such that for $G[V \setminus F]$ is a forest?

A few representative problems (II)

FEEDBACK VERTEX SET
Input: A graph $G = (V, E)$ and a positive integer k.
Parameter: k.
Question: Does there exist a subset $F \subseteq V$ of size at most k such that for $G[V \setminus F]$ is a forest?

DOMINATING SET
Input: A graph $G = (V, E)$ and a positive integers k.
Parameter: k.
Question: Does there exist a subset $D \subseteq V$ of size at most k such that for all $v \in V$, $N[v] \cap D \neq \emptyset$?

Minor-closed parameters

- A graph class $\mathcal{G}$ is *minor (contraction)-closed* if any minor (contraction) of a graph in $\mathcal{G}$ is also in $\mathcal{G}$.

Minor-closed parameters

- A graph class $\mathcal{G}$ is *minor (contraction)-closed* if any minor (contraction) of a graph in $\mathcal{G}$ is also in $\mathcal{G}$.
- A parameter P is any function mapping graphs to nonnegative integers.

Minor-closed parameters

- A graph class $\mathcal{G}$ is *minor (contraction)-closed* if any minor (contraction) of a graph in $\mathcal{G}$ is also in $\mathcal{G}$.
- A parameter P is any function mapping graphs to nonnegative integers.
- The *parameterized problem associated with* P asks, for some fixed k, whether for a given graph G, $P(G) \leq k$ (for minimization) or $P(G) \geq k$ (for maximization problem).

Minor-closed parameters

- A graph class $\mathcal{G}$ is *minor (contraction)-closed* if any minor (contraction) of a graph in $\mathcal{G}$ is also in $\mathcal{G}$.
- A parameter P is any function mapping graphs to nonnegative integers.
- The *parameterized problem associated with* P asks, for some fixed k, whether for a given graph G, $P(G) \leq k$ (for minimization) or $P(G) \geq k$ (for maximization problem).
- We say that a parameter P is *closed under taking of minors/contractions* (or, briefly, *minor/contraction-closed*) if for every graph H, $H \preceq_m G$ / $H \preceq_{cm} G$ implies that $P(H) \leq P(G)$.

Examples of minor/contraction closed parameters

- Minor-closed parameters:

  VERTEX COVER, FEEDBACK VERTEX SET, LONG PATH, TREEWIDTH, ... (why?)

Examples of minor/contraction closed parameters

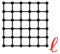
- Minor-closed parameters:

  VERTEX COVER, FEEDBACK VERTEX SET, LONG PATH, TREEWIDTH, ... (why?)

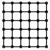
- Contraction-closed parameters:

  DOMINATING SET, CONNECTED VERTEX COVER, r-DOMINATING SET, ... (why?)

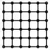
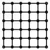
Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid: 

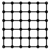
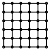
Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_\ell$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.

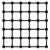
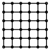
Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid: $_\ell$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor-closed, if $_\ell \preceq_m G$, then $\mathrm{tw}(G) \geq \mathrm{tw}(H_{\ell,\ell}) = \ell$.

Grid Exclusion Theorem

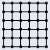
- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_\ell$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor-closed, if  $_\ell \preceq_m G$, then $\mathrm{tw}(G) \geq \mathrm{tw}(H_{\ell,\ell}) = \ell$. $\boxed{\text{Does the reverse implication hold?}}$

Grid Exclusion Theorem

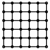
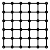
- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid: $_{\ell}$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor-closed, if $_{\ell} \preceq_m G$, then $\mathrm{tw}(G) \geq \mathrm{tw}(H_{\ell,\ell}) = \ell$.

  Does the reverse implication hold?

Theorem (Robertson and Seymour. 1986)

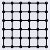
For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geq c(\ell)$ contains $_{\ell}$ *as a minor.*

Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_\ell$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor-closed, if  $_\ell \preceq_m G$, then $\mathrm{tw}(G) \geq \mathrm{tw}(H_{\ell,\ell}) = \ell$.

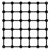
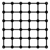
  Does the reverse implication hold?

Theorem (Robertson and Seymour. 1986)

For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geq c(\ell)$ contains  $_\ell$ *as a minor.*

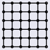
- Smallest possible function $c(\ell)$?

Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid: $_\ell$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor-closed, if $_\ell \preceq_m G$, then $\mathrm{tw}(G) \geq \mathrm{tw}(H_{\ell,\ell}) = \ell$.

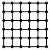
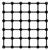
Does the reverse implication hold?

Theorem (Robertson and Seymour. 1986)

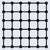
For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geq c(\ell)$ contains $_\ell$ *as a minor.*

- Smallest possible function $c(\ell)$? $\Omega(\ell^2 \log \ell) \leq c(\ell) \leq 20^{2^{\ell^5}}$

Grid Exclusion Theorem

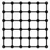
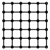
- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid: $_\ell$ We have $\text{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor-closed, if $_\ell \preceq_m G$, then $\text{tw}(G) \geq \text{tw}(H_{\ell,\ell}) = \ell$. Does the reverse implication hold?

Theorem (Robertson and Seymour. 1986)

For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geq c(\ell)$ contains $_\ell$ *as a minor.*

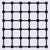
- Smallest possible function $c(\ell)$? $\Omega(\ell^2 \log \ell) \leq c(\ell) \leq 20^{2^{\ell^5}}$
- Some improvement: $c(\ell) = 2^{O(\ell \log \ell)}$. [Leaf and Seymour. 2012]

Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_\ell$ We have $\mathrm{tw}(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor-closed, if  $_\ell \preceq_m G$, then $\mathrm{tw}(G) \geq \mathrm{tw}(H_{\ell,\ell}) = \ell$.

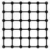
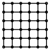
Does the reverse implication hold?

Theorem (Robertson and Seymour. 1986)

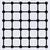
For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geq c(\ell)$ contains  $_\ell$ *as a minor.*

- Smallest possible function $c(\ell)$? $\Omega(\ell^2 \log \ell) \leq c(\ell) \leq 20^{2^{\ell^5}}$
- Some improvement: $c(\ell) = 2^{O(\ell \log \ell)}$. [Leaf and Seymour. 2012]
- Recent breakthrough: $c(\ell) = \mathrm{poly}(\ell)$. [Chekuri and Chuzhoy. 2013]

Grid Exclusion Theorem

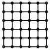
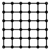
- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid: $_\ell$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor-closed, if $_\ell \preceq_m G$, then $\mathrm{tw}(G) \geq \mathrm{tw}(H_{\ell,\ell}) = \ell$. $\boxed{\text{Does the reverse implication hold?}}$

Theorem (Robertson and Seymour. 1986)

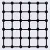
For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geq c(\ell)$ contains $_\ell$ *as a minor.*

- Smallest possible function $c(\ell)$? $\Omega(\ell^2 \log \ell) \leq c(\ell) \leq 20^{2\ell^5}$
- Some improvement: $c(\ell) = 2^{O(\ell \log \ell)}$. [Leaf and Seymour. 2012]
- Recent breakthrough: $c(\ell) = \mathrm{poly}(\ell)$. [Chekuri and Chuzhoy. 2013]
  $c(\ell) = O(\ell^9 \mathrm{polylog} \ell)$. [Chuzhoy and Tan. 2021]

Grid Exclusion Theorem

- Let $H_{\ell,\ell}$ be the $(\ell \times \ell)$-grid:  $_\ell$ We have $\mathrm{tw}\,(H_{\ell,\ell}) = \ell$.
- As TREEWIDTH is minor-closed, if  $_\ell \preceq_m G$, then $\mathrm{tw}(G) \geq \mathrm{tw}(H_{\ell,\ell}) = \ell$. $\boxed{\text{Does the reverse implication hold?}}$

Theorem (Robertson and Seymour. 1986)

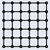
For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geq c(\ell)$ contains  $_\ell$ *as a minor.*

- Smallest possible function $c(\ell)$? $\Omega(\ell^2 \log \ell) \leq c(\ell) \leq 20^{2^{\ell^5}}$
- Some improvement: $c(\ell) = 2^{O(\ell \log \ell)}$. [Leaf and Seymour. 2012]
- Recent breakthrough: $c(\ell) = \mathrm{poly}(\ell)$. [Chekuri and Chuzhoy. 2013]
  $c(\ell) = O(\ell^9 \mathrm{polylog} \ell)$. [Chuzhoy and Tan. 2021]

$\boxed{\text{Important message}}$ grid-minors are the certificate of large treewidth.

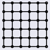
Grid Exclusion Theorems on sparse graphs

Theorem (Robertson, Seymour, Thomas. 1994)

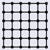
Every planar graph of treewidth $\geq 6 \cdot \ell$ *contains*  $_{\ell}$ *as a minor.*

Grid Exclusion Theorems on sparse graphs

Theorem (Robertson, Seymour, Thomas. 1994)

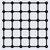
Every planar graph of treewidth $\geq 6 \cdot \ell$ *contains*  $_\ell$ *as a minor.*

Theorem (Demaine, Fomin, Hajiaghayi, Thilikos. 2005)

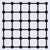
For every fixed g, *there is a constant* c_g *such that every graph of genus* g *and of treewidth* $\geq c_g \cdot \ell$ *contains*  $_\ell$ *as a minor.*

Grid Exclusion Theorems on sparse graphs

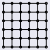
Theorem (Robertson, Seymour, Thomas. 1994)

Every planar graph of treewidth $\geq 6 \cdot \ell$ contains $_\ell$ *as a minor.*

Theorem (Demaine, Fomin, Hajiaghayi, Thilikos. 2005)

For every fixed g, there is a constant c_g such that every graph of genus g and of treewidth $\geq c_g \cdot \ell$ contains $_\ell$ *as a minor.*

Theorem (Demaine and Hajiaghayi. 2008)

For every fixed graph H, there is a constant c_H such that every H-minor-free graph of treewidth $\geq c_H \cdot \ell$ contains $_\ell$ *as a minor.*

Best constant in the above theorem is by [Kawarabayashi and Kobayashi. 2012]

Grid Exclusion Theorems on sparse graphs

Theorem (Robertson, Seymour, Thomas. 1994)

Every planar graph of treewidth $\geq 6 \cdot \ell$ contains $\boxplus_\ell$ as a minor.

Theorem (Demaine, Fomin, Hajiaghayi, Thilikos. 2005)

For every fixed g, there is a constant c_g such that every graph of genus g and of treewidth $\geq c_g \cdot \ell$ contains $\boxplus_\ell$ as a minor.

Theorem (Demaine and Hajiaghayi. 2008)

For every fixed graph H, there is a constant c_H such that every H-minor-free graph of treewidth $\geq c_H \cdot \ell$ contains $\boxplus_\ell$ as a minor.

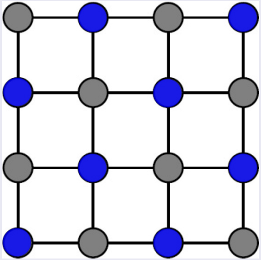
Best constant in the above theorem is by [Kawarabayashi and Kobayashi. 2012]

In sparse graphs: linear dependency between treewidth and grid-minors

How to use Grid Theorems algorithmically?

Example: FPT algorithm for Planar Vertex Cover

A vertex cover of a graph G is a set of vertices C such that every edge of G has at least one endpoint in C. Min size: $\mathbf{vc}(G)$.



Example: FPT algorithm for Planar Vertex Cover

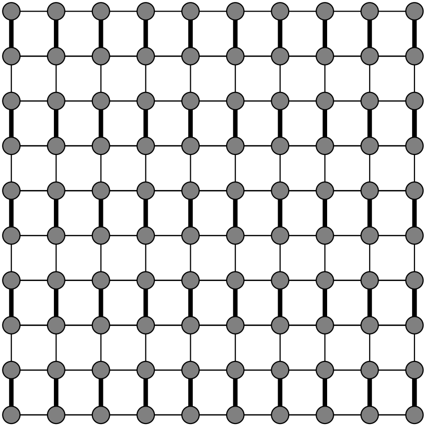
INPUT: Planar graph G on n vertices, and an integer k.

OUTPUT: Either a vertex cover of G of size $\leq k$, or a proof that G has no such a vertex cover.

RUNNING TIME: $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

$\boxed{\text{Objective}}$ subexponential FPT algorithm for Planar Vertex Cover.

Example: FPT algorithm for Planar Vertex Cover



$\mathbf{vc}(H_{\ell,\ell}) \geq \frac{\ell^2}{2}$

Example: FPT algorithm for Planar Vertex Cover

Let G be a planar graph of treewidth $\geq 6 \cdot \ell$

Example: FPT algorithm for Planar Vertex Cover

Let G be a planar graph of treewidth $\geq 6 \cdot \ell$ $\Longrightarrow$ G contains the $(\ell \times \ell)$-grid $H_{\ell,\ell}$ as a minor

Example: FPT algorithm for Planar Vertex Cover

Let G be a planar graph of treewidth $\geq 6 \cdot \ell$ $\Longrightarrow$ G contains the $(\ell \times \ell)$-grid $H_{\ell,\ell}$ as a minor

- The size of any vertex cover of $H_{\ell,\ell}$ is at least $\ell^2/2$.
- Recall that Vertex Cover is a minor-closed parameter.
- Since $H_{\ell,\ell} \preceq_m G$, it holds that $\mathbf{vc}(G) \geq \mathbf{vc}(H_{\ell,\ell}) \geq \ell^2/2$.

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\mathrm{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geq \ell^2/2$.

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\mathrm{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geq \ell^2/2$.

WIN/WIN approach:

- If $k < \ell^2/2$, we can safely answer "NO".

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\text{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geq \ell^2/2$.

WIN/WIN approach:

- If $k < \ell^2/2$, we can safely answer "NO".
- If $k \geq \ell^2/2$, then $\text{tw}(G) = O(\ell) = O(\sqrt{k})$,

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\mathrm{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geq \ell^2/2$.

WIN/WIN approach:

- If $k < \ell^2/2$, we can safely answer "NO".
- If $k \geq \ell^2/2$, then $\mathrm{tw}(G) = O(\ell) = O(\sqrt{k})$, and we can solve the problem by standard DP in time $2^{O(\mathrm{tw}(G))} \cdot n^{O(1)}$

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\text{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geq \ell^2/2$.

WIN/WIN approach:

- If $k < \ell^2/2$, we can safely answer "NO".
- If $k \geq \ell^2/2$, then $\text{tw}(G) = O(\ell) = O(\sqrt{k})$, and we can solve the problem by standard DP in time $2^{O(\text{tw}(G))} \cdot n^{O(1)} = 2^{O(\sqrt{k})} \cdot n^{O(1)}$.

We are already very close to an algorithm...

Recall:

- k is the parameter of the problem.
- We have that $\mathrm{tw}(G) = 6 \cdot \ell$ and ℓ is the size of a grid-minor of G.
- Therefore, $\mathbf{vc}(G) \geq \ell^2/2$.

WIN/WIN approach:

- If $k < \ell^2/2$, we can safely answer "NO".
- If $k \geq \ell^2/2$, then $\mathrm{tw}(G) = O(\ell) = O(\sqrt{k})$, and we can solve the problem by standard DP in time $2^{O(\mathrm{tw}(G))} \cdot n^{O(1)} = 2^{O(\sqrt{k})} \cdot n^{O(1)}$.

This gives a subexponential FPT algorithm!

Was Vertex Cover really just an example...?

What is so special in Vertex Cover?

Where did we use planarity?

Was Vertex Cover really just an example...?

What is so special in Vertex Cover?

★ Nothing special! It is just a minor bidimensional parameter:

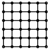
minor-closed $\quad + \quad \mathbf{vc}(\boxplus_{k}) = \Omega(k^2)$.

Where did we use planarity?

Was VERTEX COVER really just an example...?

What is so special in VERTEX COVER?

★ Nothing special! It is just a minor bidimensional parameter:

minor-closed $+$ **vc**($_k$) $= \Omega(k^2)$.

Where did we use planarity?

★ Only the linear Grid Exclusion Theorem!

Arguments go through up to H-minor-free graphs.

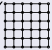
Next subsection is...

Minor Bidimensionality:

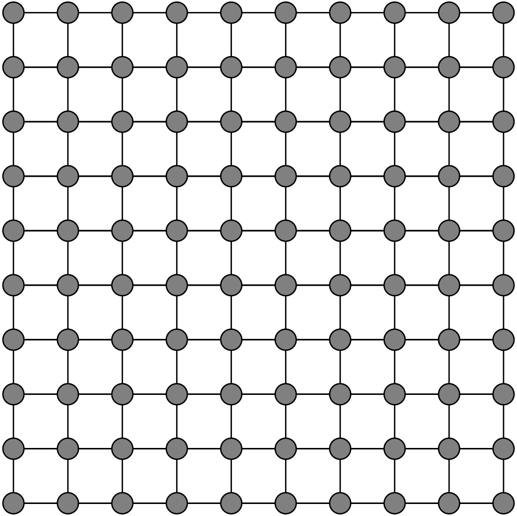
[Demaine, Fomin, Hajiaghayi, Thilikos. 2005]

Definition

A parameter **p** is *minor bidimensional* if

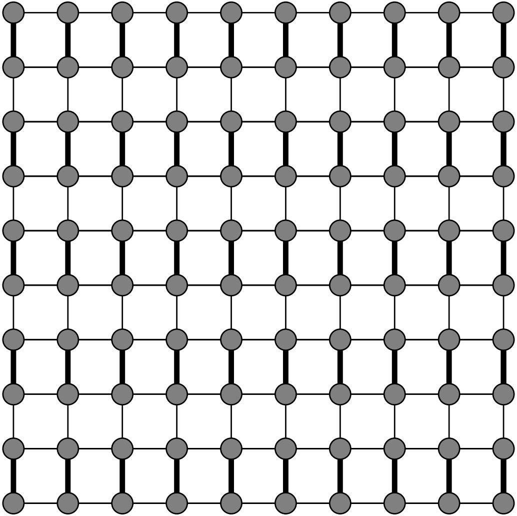
1. **p** is closed under taking of minors (minor-closed), and
2. **p** $\Bigg($  $_{k}\Bigg) = \Omega(k^2)$.

Vertex Cover of a Grid



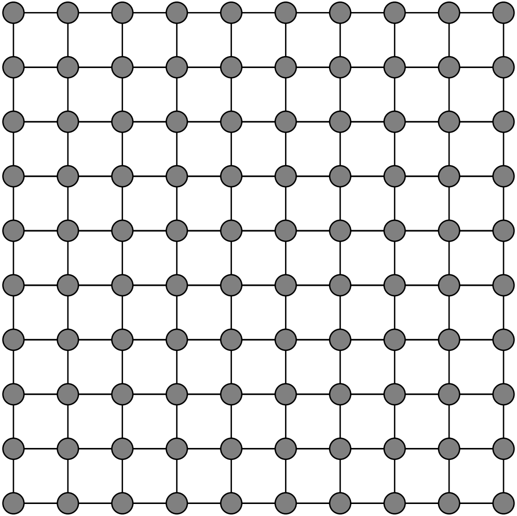
$H_{\ell,\ell}$ for $\ell = 10$

Vertex Cover of a Grid

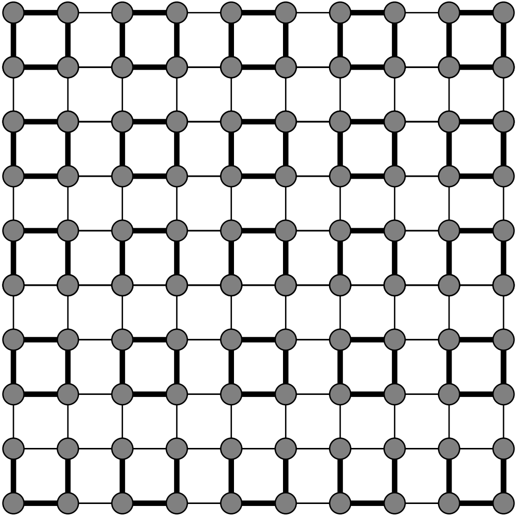


$\mathbf{vc}(H_{\ell,\ell}) \geq \ell^2/2$

Feedback Vertex Set of a Grid



Feedback Vertex Set of a Grid



$\mathbf{fvs}(H_{\ell,\ell}) \geq \ell^2/4$

How to obtain subexponential algorithms for BP?

- First we must restrict ourselves to special graph classes, like planar or *H*-minor-free graphs.

How to obtain subexponential algorithms for BP?

- First we must restrict ourselves to special graph classes, like planar or H-minor-free graphs.
  - Show that if the graph has large treewidth $(> c\sqrt{k})$ then it has a $(\sqrt{k} \times \sqrt{k})$-grid as a minor, and hence the answer to the problem is YES (or NO) immediately.

How to obtain subexponential algorithms for BP?

- First we must restrict ourselves to special graph classes, like planar or H-minor-free graphs.
  - Show that if the graph has large treewidth $(> c\sqrt{k})$ then it has a $(\sqrt{k} \times \sqrt{k})$-grid as a minor, and hence the answer to the problem is YES (or NO) immediately.
  - Otherwise, the treewidth is bounded by $c\sqrt{k}$, and hence we can use a dynamic programming (DP) algorithm on graphs of bounded treewidth.

How to obtain subexponential algorithms for BP?

- First we must restrict ourselves to special graph classes, like planar or H-minor-free graphs.
  - Show that if the graph has large treewidth $(> c\sqrt{k})$ then it has a $(\sqrt{k} \times \sqrt{k})$-grid as a minor, and hence the answer to the problem is YES (or NO) immediately.
  - Otherwise, the treewidth is bounded by $c\sqrt{k}$, and hence we can use a dynamic programming (DP) algorithm on graphs of bounded treewidth.
- If we have a DP algorithm for bounded treewidth running in time c^t or t^t, then it implies $2^{O(\sqrt{k})}$ or $2^{O(\sqrt{k}\log k)}$ algorithm.

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

1. Compute (or approximate) $\mathbf{tw}(G)$.

2. If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then safely answer NO (or YES).

3. Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

$\boxed{1}$ Compute (or approximate) $\mathbf{tw}(G)$.
We can use a fast FPT algorithm or a constant-factor approx.

$\boxed{2}$ If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then safely answer NO (or YES).

$\boxed{3}$ Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

1. Compute (or approximate) $\mathbf{tw}(G)$.
   We can use a fast FPT algorithm or a constant-factor approx.
2. If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then safely answer NO (or YES).
   This follows because of the linear Grid Exclusion Theorems.
3. Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

$\boxed{1}$ Compute (or approximate) $\mathbf{tw}(G)$.
We can use a fast FPT algorithm or a constant-factor approx.

$\boxed{2}$ If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then safely answer NO (or YES).
This follows because of the linear Grid Exclusion Theorems.

$\boxed{3}$ Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.
Doing DP in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$ is a whole area of research:

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ be a minor bidimensional graph parameter computable in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding "$\mathbf{p}(G) = k$" can be done in time $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

1. Compute (or approximate) $\mathbf{tw}(G)$.
   We can use a fast FPT algorithm or a constant-factor approx.
2. If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then safely answer NO (or YES).
   This follows because of the linear Grid Exclusion Theorems.
3. Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.
   Doing DP in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$ is a whole area of research:
   - Exploiting Catalan structures on sparse graphs. [Dorn et al. 2005-2008] [Rué, S., Thilikos. 2010]

Piecing everything together

Theorem

Let G be an H-minor-free graph, and let $\mathbf{p}$ *be a minor bidimensional graph parameter computable in time* $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$.
Then deciding " $\mathbf{p}(G) = k$ *" can be done in time* $2^{O(\sqrt{k})} \cdot n^{O(1)}$.

$\boxed{1}$ Compute (or approximate) $\mathbf{tw}(G)$.
We can use a fast FPT algorithm or a constant-factor approx.

$\boxed{2}$ If $\mathbf{tw}(G) = \Omega(\sqrt{k})$, then safely answer NO (or YES).
This follows because of the linear Grid Exclusion Theorems.

$\boxed{3}$ Otherwise $\mathbf{tw}(G) = O(\sqrt{k})$, and we solve the problem by DP.
Doing DP in time $2^{O(\mathbf{tw}(G))} \cdot n^{O(1)}$ is a whole area of research:

- Exploiting Catalan structures on sparse graphs. [Dorn et al. 2005-2008] [Rué, S., Thilikos. 2010]
- Randomized algorithms using Cut&Count. [Cygan et al. 2011]
- Deterministic algorithms based on matrix rank. [Boadlaender et al. 2012]
- Deterministic algorithms based on matroids. [Fomin et al. 2013]

Further applications of Bidimensionality

1. Bidimensionality + DP $\Rightarrow$ $\boxed{\text{Subexponential FPT algorithms}}$

   [Demaine, Fomin, Hajiaghayi, Thilikos. 2004-2005]
   [Fomin, Golovach, Thilikos. 2009]

Further applications of Bidimensionality

1. Bidimensionality + DP $\Rightarrow$ $\boxed{\text{Subexponential FPT algorithms}}$

   [Demaine, Fomin, Hajiaghayi, Thilikos. 2004-2005]
   [Fomin, Golovach, Thilikos. 2009]

2. Bidimensionality + separation properties $\Rightarrow$ $\boxed{\text{(E)PTAS}}$

   [Demaine and Hajiaghayi. 2005]
   [Fomin, Lokshtanov, Raman, Saurabh. 2011]

Further applications of Bidimensionality

1. Bidimensionality + DP $\Rightarrow$ $\boxed{\text{Subexponential FPT algorithms}}$

   [Demaine, Fomin, Hajiaghayi, Thilikos. 2004-2005]
   [Fomin, Golovach, Thilikos. 2009]

2. Bidimensionality + separation properties $\Rightarrow$ $\boxed{\text{(E)PTAS}}$

   [Demaine and Hajiaghayi. 2005]
   [Fomin, Lokshtanov, Raman, Saurabh. 2011]

3. Bidimensionality + separation properties $\Rightarrow$ $\boxed{\text{Kernelization}}$

   [Fomin, Lokshtanov, Saurabh, Thilikos. 2009-2010]

Further applications of Bidimensionality

1. Bidimensionality + DP $\Rightarrow$ $\boxed{\text{Subexponential FPT algorithms}}$

   [Demaine, Fomin, Hajiaghayi, Thilikos. 2004-2005]
   [Fomin, Golovach, Thilikos. 2009]

2. Bidimensionality + separation properties $\Rightarrow$ $\boxed{\text{(E)PTAS}}$

   [Demaine and Hajiaghayi. 2005]
   [Fomin, Lokshtanov, Raman, Saurabh. 2011]

3. Bidimensionality + separation properties $\Rightarrow$ $\boxed{\text{Kernelization}}$

   [Fomin, Lokshtanov, Saurabh, Thilikos. 2009-2010]

4. Bidimensionality + new Grid Theorems $\Rightarrow$ $\boxed{\text{Geometric graphs}}$

   [Fomin, Lokshtanov, Saurabh. 2012]
   [Grigoriev, Koutsonas, Thilikos. 2013]

Next section is...

Basic principle of the irrelevant vertex technique

This technique was invented in [Robertson and Seymour. 1995]

Basic principle of the irrelevant vertex technique

This technique was invented in [Robertson and Seymour. 1995]

DISJOINT PATHS
Input: a graph G and k pairs of vertices $T = \{s_1, \ldots, s_k, t_1, \ldots, t_k\}$.
Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

Basic principle of the irrelevant vertex technique

This technique was invented in [Robertson and Seymour. 1995]

Disjoint Paths

Input: a graph G and k pairs of vertices $T = \{s_1, \ldots, s_k, t_1, \ldots, t_k\}$.

Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

Strategy:

1. If $\mathrm{tw}(G) > f(k)$, find an irrelevant vertex:

   A vertex $v \in V(G)$ such that (G, T, k) and $(G \setminus v, T, k)$ are equivalent instances.

Basic principle of the irrelevant vertex technique

This technique was invented in [Robertson and Seymour. 1995]

Disjoint Paths
Input: a graph G and k pairs of vertices $T = \{s_1, \ldots, s_k, t_1, \ldots, t_k\}$.
Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

Strategy:

1. If $\text{tw}(G) > f(k)$, find an irrelevant vertex:

   A vertex $v \in V(G)$ such that (G, T, k) and $(G \setminus v, T, k)$ are equivalent instances.

2. Otherwise, if $\text{tw}(G) \leq f(k)$, solve the problem using dynamic programming (by Courcelle).

How to find an irrelevant vertex when the treewidth is large?

How to find an irrelevant vertex when the treewidth is large?

By using the Grid Exclusion Theorem!

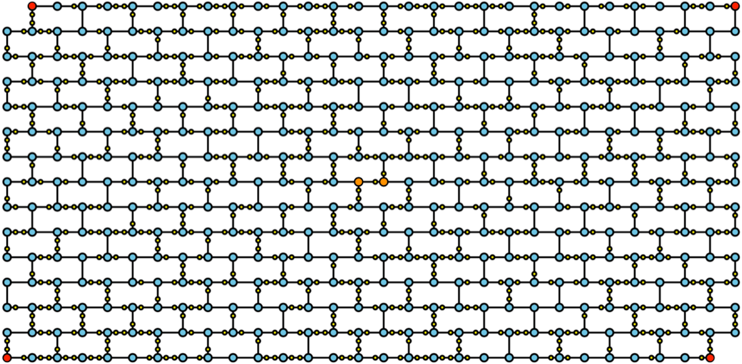
How to find an irrelevant vertex when the treewidth is large?

By using the $\boxed{\text{Wall}}$ Exclusion Theorem!

How to find an irrelevant vertex when the treewidth is large?

Theorem (Robertson and Seymour. 1986)

For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geq c(\ell)$ contains an ℓ-wall as a minor.

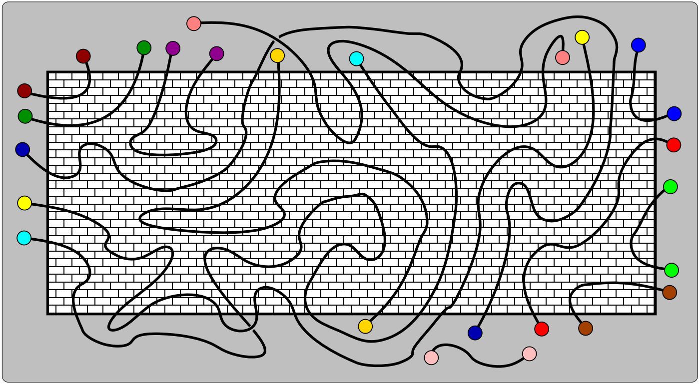


[Figure by Dimitrios M. Thilikos]

How to find an irrelevant vertex when the treewidth is large?

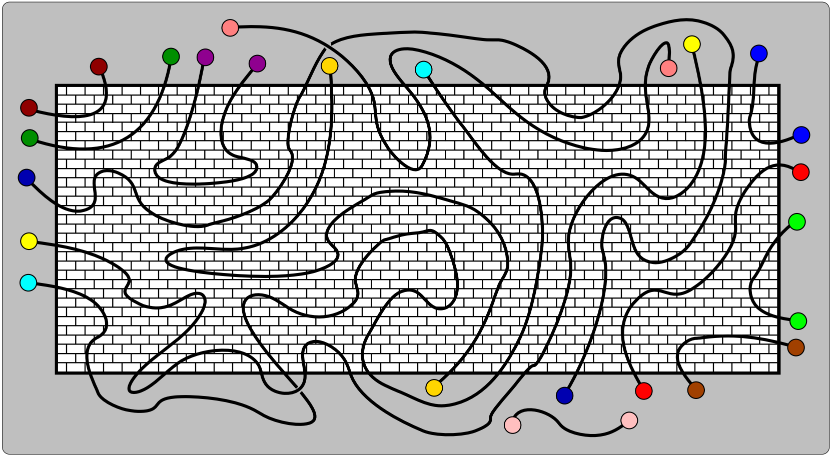
Theorem (Robertson and Seymour. 1986)

For every integer $\ell > 0$, there is an integer $c(\ell)$ such that every graph of treewidth $\geq c(\ell)$ contains an ℓ-wall as a minor.

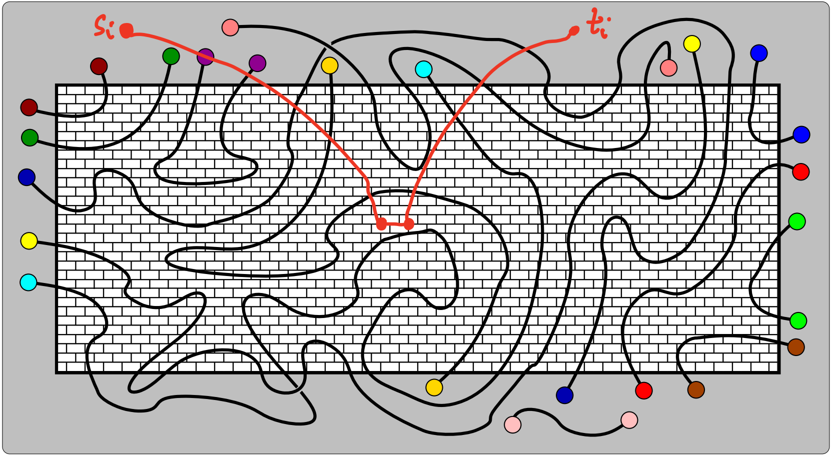


[Figure by Dimitrios M. Thilikos]

Goal: declare one of the central vertices of the wall irrelevant.



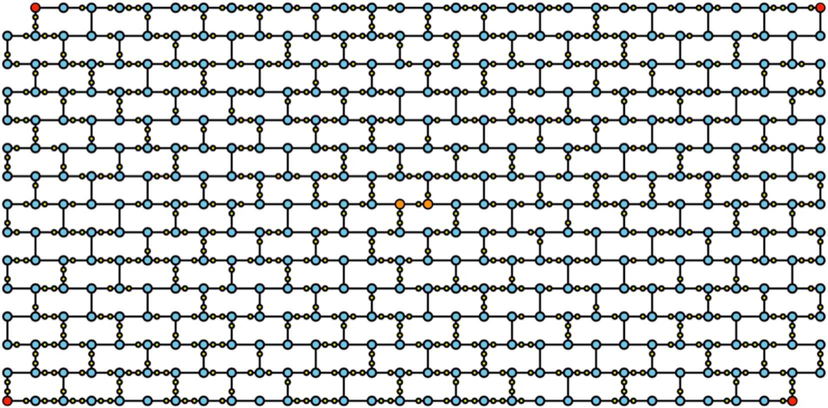
Goal: declare one of the central vertices of the wall irrelevant.



This is only possible if the wall is insulated from the exterior!

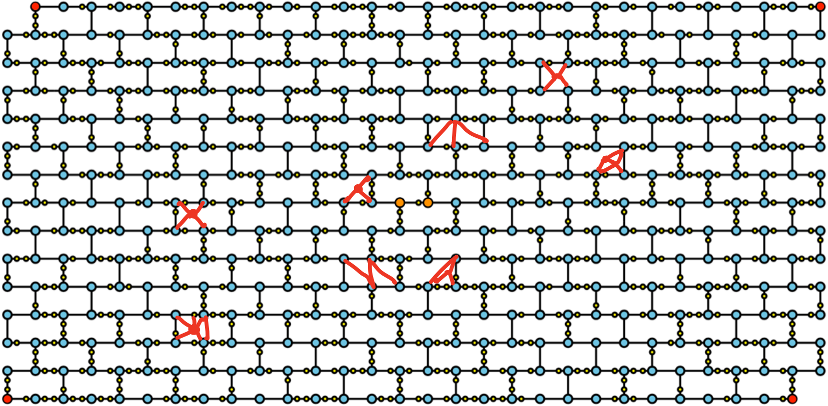
Flat walls

Goal: enrich the notion of wall so that we can insulate it from the exterior.



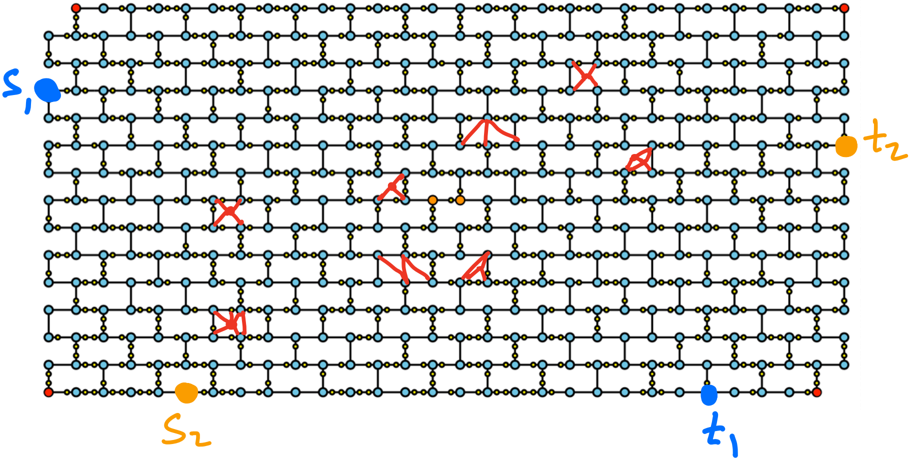
Flat walls

We need to allow some extra edges in the interior of the wall.



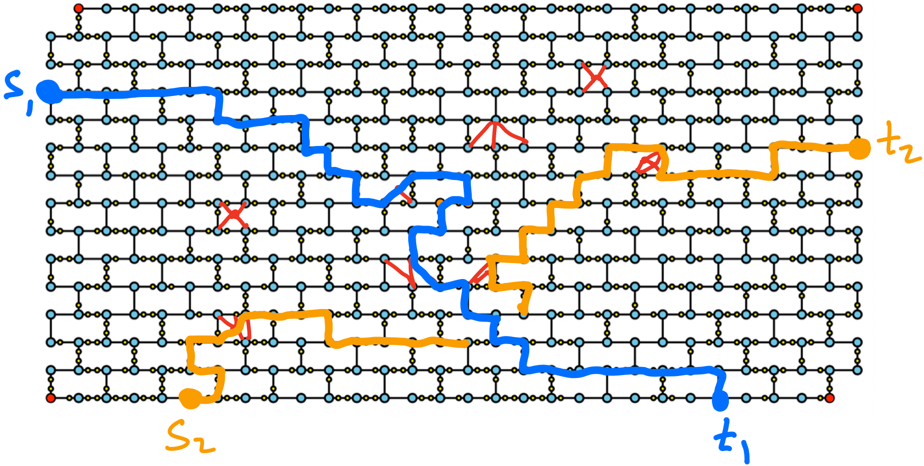
Flat walls

We impose a topological property that defines the "flatness" of the wall.



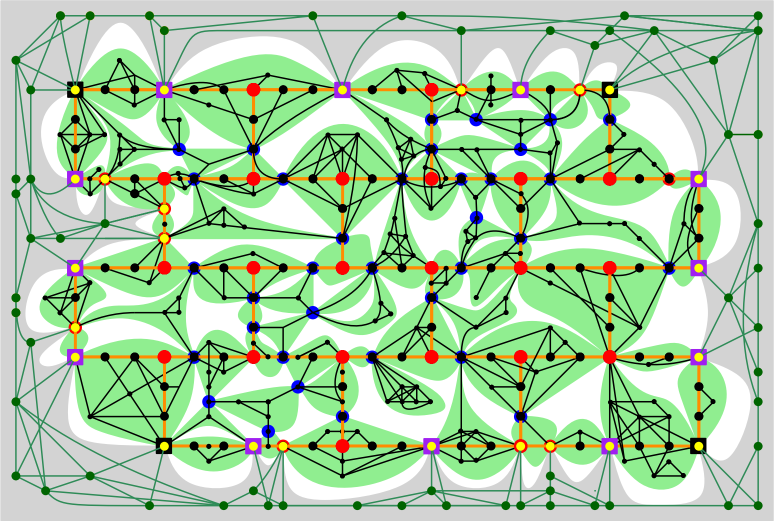
Flat walls

There are no crossing paths $s_1 - t_1$ and $s_2 - t_2$ from/to the perimeter.

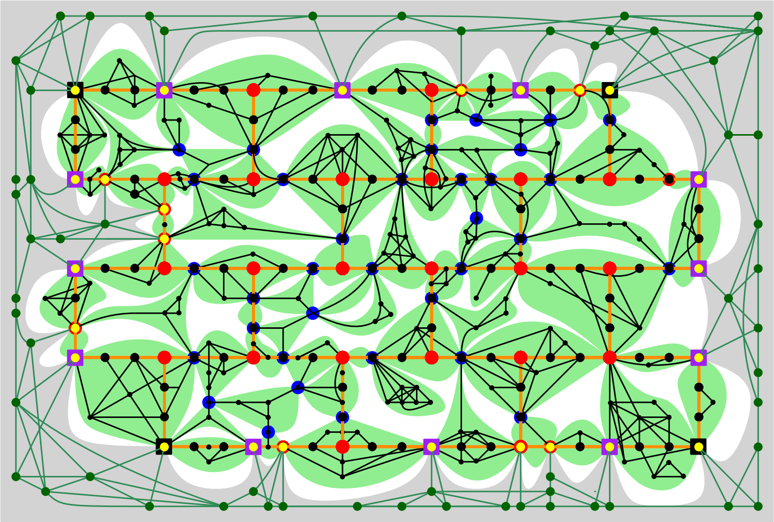


Flat walls

A real flat wall can be quite wild... [Figure by Dimitrios M. Thilikos]

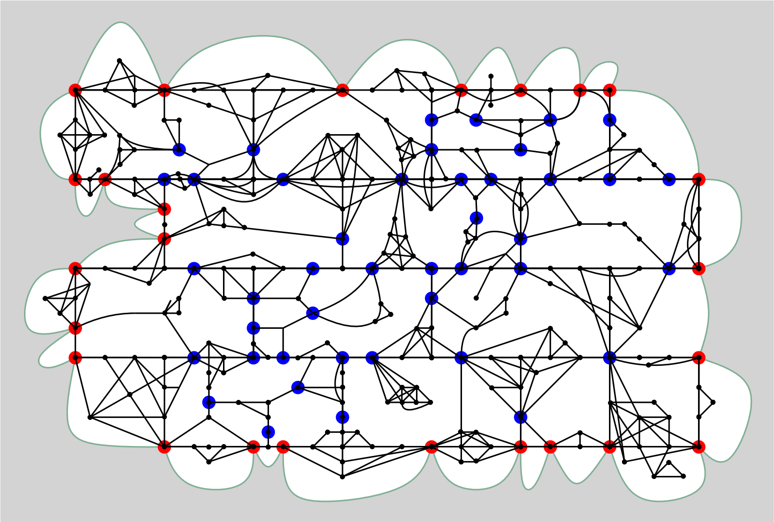


Flat walls: a bit more formal



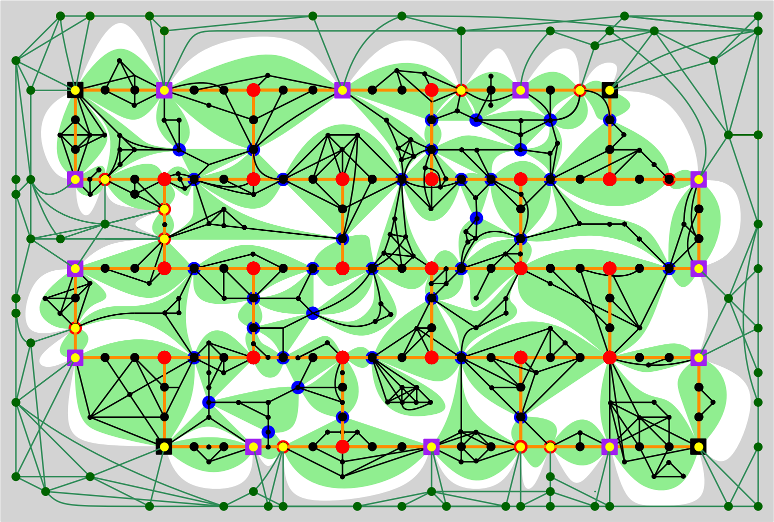
[Figures by Dimitrios M. Thilikos]

Flat walls: a bit more formal



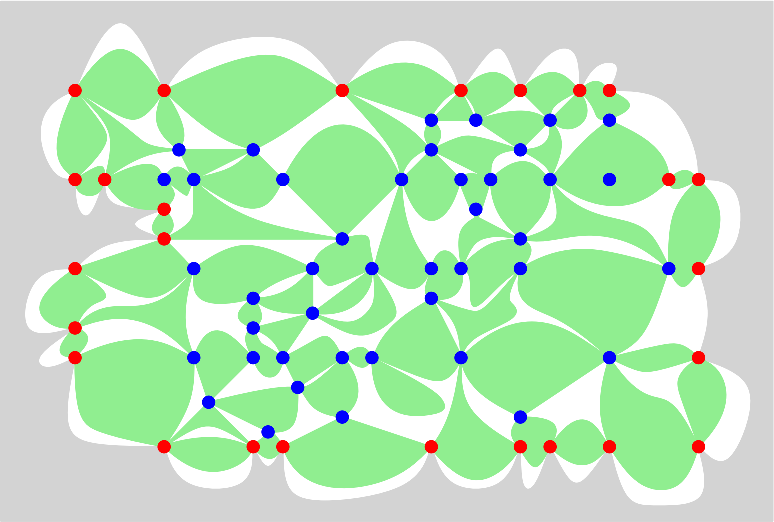
[Figures by Dimitrios M. Thilikos]

Flat walls: a bit more formal



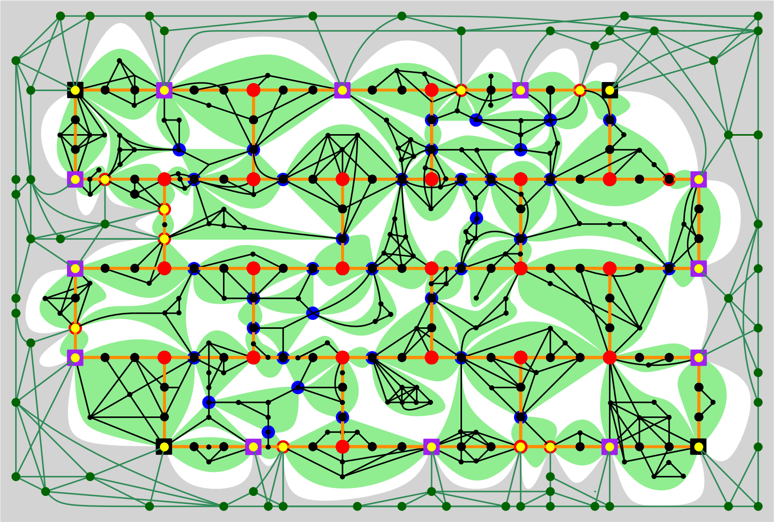
[Figures by Dimitrios M. Thilikos]

Flat walls: a bit more formal



[Figures by Dimitrios M. Thilikos]

Flat walls: a bit more formal



[Figures by Dimitrios M. Thilikos]

The Weak Structure Graph Minors Theorem

Theorem (Robertson and Seymour. 1995)

There exist recursive functions $f_1 : \mathbb{N}^2 \to \mathbb{N}$ *and* $f_2 : \mathbb{N} \to \mathbb{N}$, *such that for every graph* G *and every* $q, r \in \mathbb{N}$, *one of the following holds:*

The Weak Structure Graph Minors Theorem

Theorem (Robertson and Seymour. 1995)

There exist recursive functions $f_1 : \mathbb{N}^2 \to \mathbb{N}$ *and* $f_2 : \mathbb{N} \to \mathbb{N}$, *such that for every graph* G *and every* $q, r \in \mathbb{N}$, *one of the following holds:*

1. K_q *is a minor of* G.

The Weak Structure Graph Minors Theorem

Theorem (Robertson and Seymour. 1995)

There exist recursive functions $f_1 : \mathbb{N}^2 \to \mathbb{N}$ and $f_2 : \mathbb{N} \to \mathbb{N}$, such that for every graph G and every $q, r \in \mathbb{N}$, one of the following holds:

1. *K_q is a minor of G.*
2. *The treewidth of G is at most $f_1(q, r)$.*

The Weak Structure Graph Minors Theorem

Theorem (Robertson and Seymour. 1995)

There exist recursive functions $f_1 : \mathbb{N}^2 \to \mathbb{N}$ and $f_2 : \mathbb{N} \to \mathbb{N}$, such that for every graph G and every $q, r \in \mathbb{N}$, one of the following holds:

1. K_q *is a minor of* G.
2. *The treewidth of* G *is at most* $f_1(q, r)$.
3. *There exists* $A \subseteq V(G)$ *(apices) with* $|A| \leq f_2(q)$ *such that* $G \setminus A$ *contains as a subgraph a flat wall* W *of height* r.

The Weak Structure Graph Minors Theorem

Theorem (Robertson and Seymour. 1995)

There exist recursive functions $f_1 : \mathbb{N}^2 \to \mathbb{N}$ and $f_2 : \mathbb{N} \to \mathbb{N}$, such that for every graph G and every $q, r \in \mathbb{N}$, one of the following holds:

1. K_q *is a minor of* G.
2. *The treewidth of* G *is at most* $f_1(q, r)$.
3. *There exists* $A \subseteq V(G)$ *(apices) with* $|A| \leq f_2(q)$ *such that* $G \setminus A$ *contains as a subgraph a flat wall* W *of height* r.

There are many different variants and optimizations of this theorem...

[Chuzhoy. 2015]

[Kawarabayashi, Thomas, Wollan. 2018]

[S., Stamoulis, Thilikos. 2021]

The Weak Structure Graph Minors Theorem

Theorem (Robertson and Seymour. 1995)

There exist recursive functions $f_1 : \mathbb{N}^2 \to \mathbb{N}$ and $f_2 : \mathbb{N} \to \mathbb{N}$, such that for every graph G and every $q, r \in \mathbb{N}$, one of the following holds:

1. *K_q is a minor of G.*
2. *The treewidth of G is at most $f_1(q, r)$.*
3. *There exists $A \subseteq V(G)$ (apices) with $|A| \leq f_2(q)$ such that $G \setminus A$ contains as a subgraph a flat wall W of height r.*

There are many different variants and optimizations of this theorem...

[Chuzhoy. 2015]

[Kawarabayashi, Thomas, Wollan. 2018]

[S., Stamoulis, Thilikos. 2021]

Important: possible to find one of the outputs in time $f(q, r) \cdot |V(G)|$.

Back to the DISJOINT PATHS problem

DISJOINT PATHS
Input: a graph G and k pairs of vertices $T = \{s_1, \ldots, s_k, t_1, \ldots, t_k\}$.
Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

Back to the DISJOINT PATHS problem

DISJOINT PATHS
Input: a graph G and k pairs of vertices $T = \{s_1, \ldots, s_k, t_1, \ldots, t_k\}$.
Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

By the Weak Structure Theorem:

- If $\mathrm{tw}(G) \leq f(k)$: solve using dynamic programming.

Back to the DISJOINT PATHS problem

DISJOINT PATHS
Input: a graph G and k pairs of vertices $T = \{s_1, \ldots, s_k, t_1, \ldots, t_k\}$.
Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

By the Weak Structure Theorem:

- If $\mathrm{tw}(G) \leq f(k)$: solve using dynamic programming.
- If G contains a $K_{g(k)}$-minor: "easy" to find an irrelevant vertex.

Back to the Disjoint Paths problem

Disjoint Paths

Input: a graph G and k pairs of vertices $T = \{s_1, \ldots, s_k, t_1, \ldots, t_k\}$.

Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

By the Weak Structure Theorem:

- If $\text{tw}(G) \leq f(k)$: solve using dynamic programming.
- If G contains a $K_{g(k)}$-minor: "easy" to find an irrelevant vertex.
- If G contains a "small" apex set A and a flat wall W in $G \setminus A$ of size at least $h(k)$: declare the central vertex of the flat wall irrelevant.

Back to the DISJOINT PATHS problem

DISJOINT PATHS

Input: a graph G and k pairs of vertices $T = \{s_1, \ldots, s_k, t_1, \ldots, t_k\}$.

Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

By the Weak Structure Theorem:

- If $\text{tw}(G) \leq f(k)$: solve using dynamic programming.
- If G contains a $K_{g(k)}$-minor: “easy” to find an irrelevant vertex.
- If G contains a “small” apex set A and a flat wall W in $G \setminus A$ of size at least $h(k)$: declare the central vertex of the flat wall irrelevant.

The irrelevant vertex technique has been applied to many problems...

Back to the DISJOINT PATHS problem

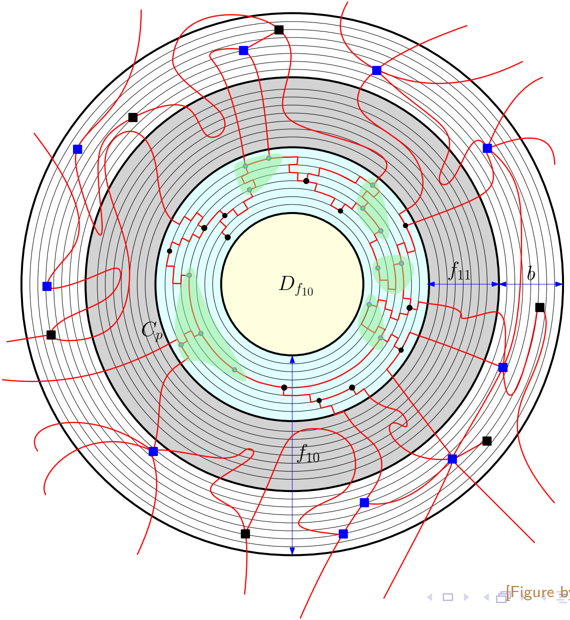
DISJOINT PATHS
Input: a graph G and k pairs of vertices $T = \{s_1, \ldots, s_k, t_1, \ldots, t_k\}$.
Question: does G contain k vertex-disjoint paths $P_1, \ldots, P_k$ such that P_i connects s_i to t_i?

By the Weak Structure Theorem:

- If $\mathrm{tw}(G) \leq f(k)$: solve using dynamic programming.
- If G contains a $K_{g(k)}$-minor: "easy" to find an irrelevant vertex.
- If G contains a "small" apex set A and a flat wall W in $G \setminus A$ of size at least $h(k)$: declare the central vertex of the flat wall irrelevant.

The irrelevant vertex technique has been applied to many problems... usually with a lot of technical pain.

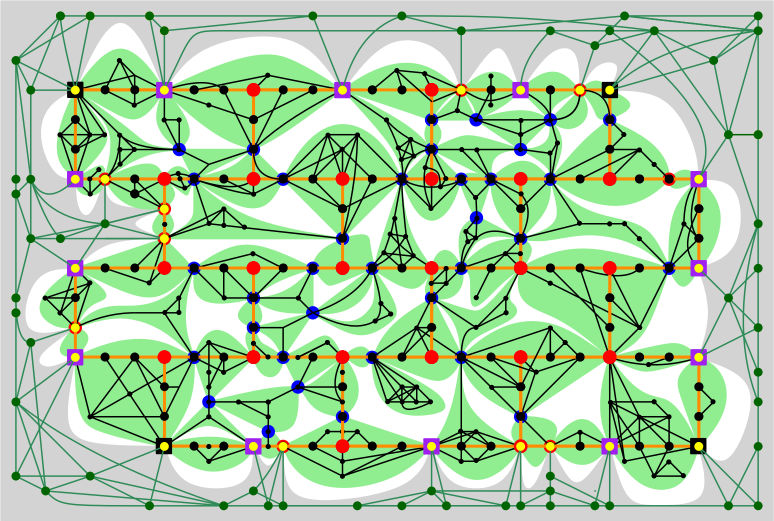
Rerouting inside a big flat wall...



[Figure by Dimitrios M. Thilikos]

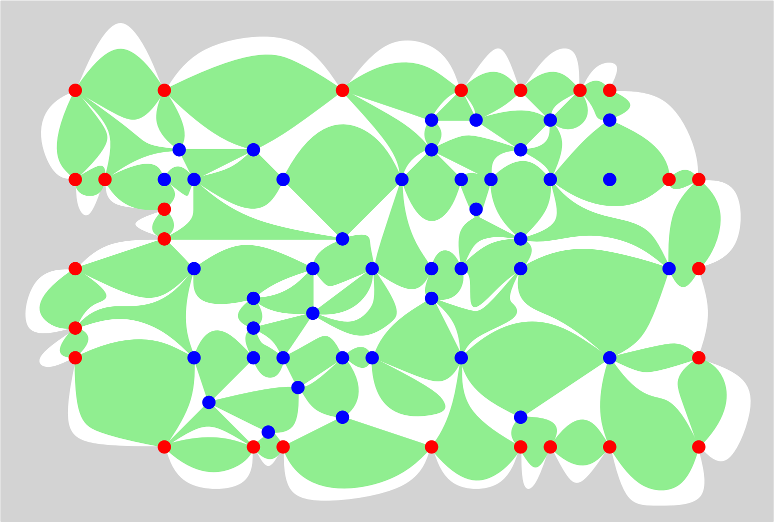
Crucial notion: homogeneity

In order to declare a vertex irrelevant for some problem, usually we need to consider a homogenous flat wall, which we proceed to define.



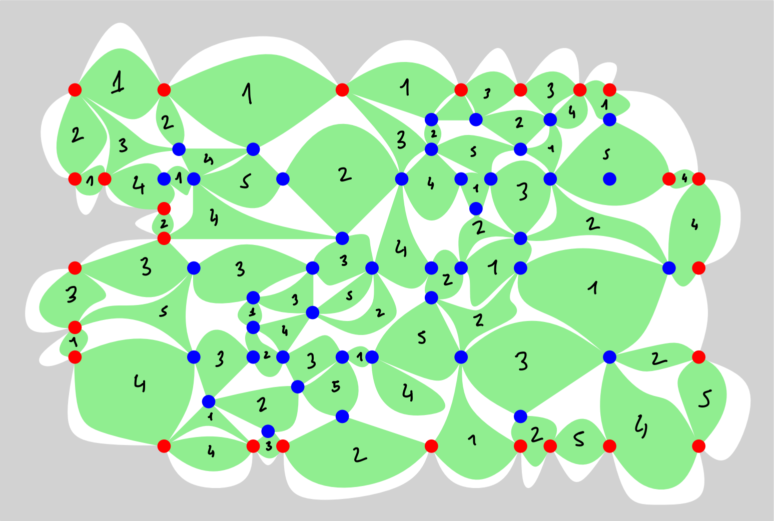
Crucial notion: homogeneity

We consider a flap-coloring encoding the relevant information of our favorite problem inside each flap (similar to tables of DP).



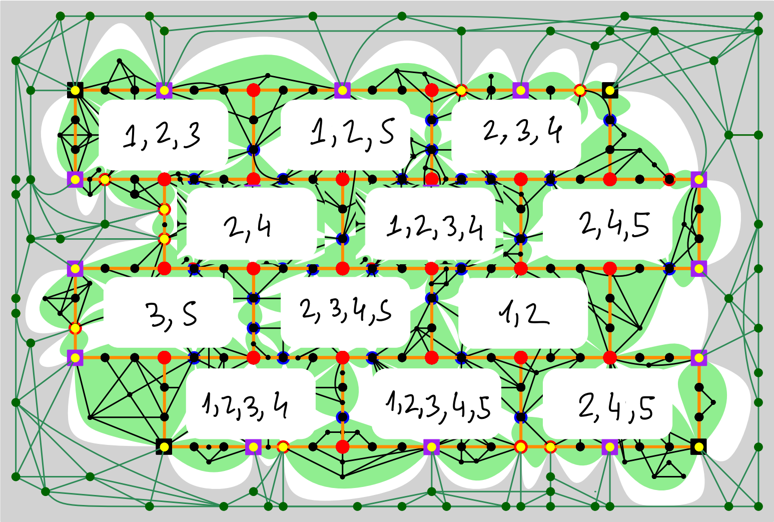
Crucial notion: homogeneity

We consider a flap-coloring encoding the relevant information of our favorite problem inside each flap (similar to tables of DP).



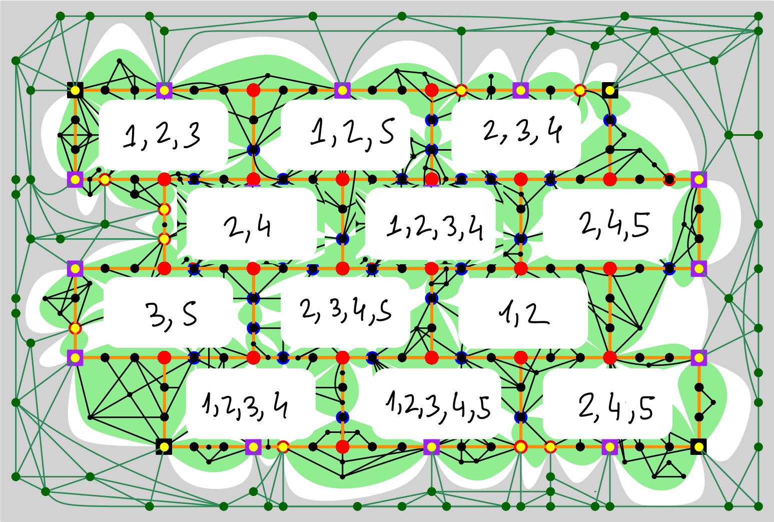
Crucial notion: homogeneity

For every brick of the wall, we define its palette as the colors appearing in the flaps it contains.



Crucial notion: homogeneity

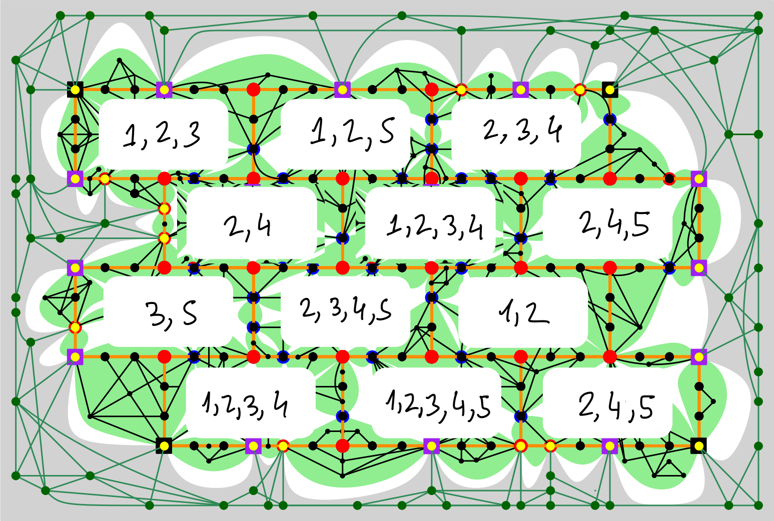
A flat wall is homogenous if every (internal) brick has the same palette.
Fact: every brick of a homogenous flat wall has the same "behavior".



Crucial notion: homogeneity

Price of homogeneity to obtain a homogenous flat r-wall (zooming):

If we have c colors, we need to start with a flat r^c-wall. (why?)



Gràcies!