On the complexity of embedding in graph products

Therese Biedl [1] David Eppstein Torsten Ueckerdt

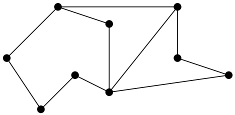
[1]University of Waterloo, Canada. biedl@uwaterloo.ca

August 2, 2023

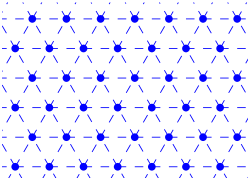
With thanks to the Workshop on Graph Product Structure Theory (BIRS21w5235) at the Banff International Research Station, Nov. 21-26, 2021.

Graph embedding

Given: A graph G



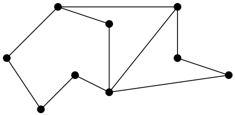
Given: A *host graph* H



Want: Can G be embedded in H?
$\iff$ Is G a subgraph of H?
$\iff$ $G \subseteq H$?

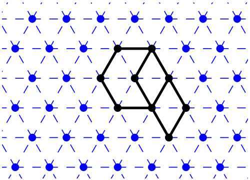
Graph embedding

Given: A graph G



Possibly restricted to be a tree or planar or ....

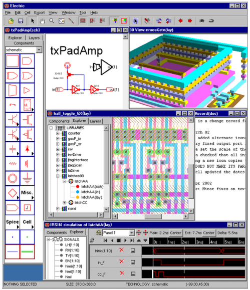
Given: A *host graph H*



Often with structure, e.g. grid or $T \boxtimes P$ (defined below).
Often infinite (but G will always be finite).

Want: Can G be embedded in H?
$\iff$ Is G a subgraph of H?
$\iff$ $G \subseteq H$?

Motivation/applications

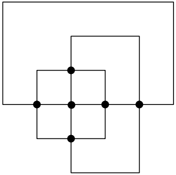


©wikimedia

VLSI design (1970s):

- create a computer chip
- one step: how to route connections horizontally and vertically

⇔ how to embed graph in grid



Orthogonal graph drawing (1990s):

- similar to above, but focus on beauty rather than area
- grid embedding ⇔ orth. drawing with edge-lengths 1

Motivation/applications

Graph theory: Extract properties of G via embedding in host-graph.

Theorem (Graph Product Structure (DJMMUW20))

Every planar graph G can be embedded in $H \boxtimes P_\infty$ for some planar graph H of treewidth ≤ 8.

(P_∞: infinite path. *Treewidth*, $\boxtimes$: see below.)

Motivation/applications

Graph theory: Extract properties of G via embedding in host-graph.

Theorem (Graph Product Structure (DJMMUW20))

Every planar graph G can be embedded in $H \boxtimes P_\infty$ for some planar graph H of treewidth ≤ 8.

(P_∞: infinite path. *Treewidth*, $\boxtimes$: see below.)

- Lots of implications: queue layouts, non-repetetive colourings, adjacency labellings, . . .
- Lots of generalizations: k-planar graphs, squares of planar graphs, . . .
- Embedding can be computed efficiently
- One can improve on '≤ 8'

Row treewidth and row pathwidth

Theorem (Graph Product Structure (DJMMUW20,UWY21))

Every planar graph G can be embedded in $H \boxtimes P_\infty$ for some graph H of treewidth ≤ 6.

- Define *row-treewidth*(G): Smallest k s.t. $G \subseteq H \boxtimes P_\infty$ for some graph H of treewidth k.
- [UWY21]: row-treewidth$(G) \leq 6$ for all planar graphs G.
- [DJM+20]: row-treewidth$(G) \geq 3$ for some planar graph G.

Q1: Which number in $\{3, 4, 5, 6\}$ is the right number here?

(Lovely question, but not in this talk)

Row treewidth and row pathwidth

Theorem (Graph Product Structure (DJMMUW20,UWY21))

Every planar graph G can be embedded in $H \boxtimes P_\infty$ for some graph H of treewidth ≤ 6.

- Define *row-treewidth*(G): Smallest k s.t. $G \subseteq H \boxtimes P_\infty$ for some graph H of treewidth k.
- [UWY21]: row-treewidth$(G) \leq 6$ for all planar graphs G.
- [DJM+20]: row-treewidth$(G) \geq 3$ for some planar graph G.

Q1: Which number in $\{3, 4, 5, 6\}$ is the right number here?

(Lovely question, but not in this talk)

Q2: What is the complexity of computing row-treewidth(G)?

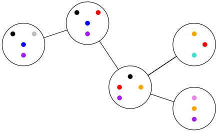
Some definitions

Goal: What is the complexity of testing whether $G \subseteq H \boxtimes P_\infty$ for some graph H of treewidth/pathwidth k?

Some definitions

Goal: What is the complexity of testing whether $G \subseteq H \boxtimes P_\infty$ for some graph H of treewidth/pathwidth k?

- Treewidth: That parameter with bags arranged in a tree.

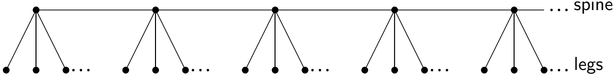


(We only need: treewidth 1 $\Leftrightarrow$ subgraph of tree)

- Pathwidth: That parameter with bags arranged in a path.

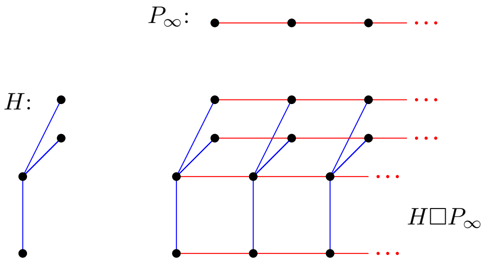
(We only need: pathwidth 1 $\Leftrightarrow$ subgraph of C_∞)

caterpillar C_∞:



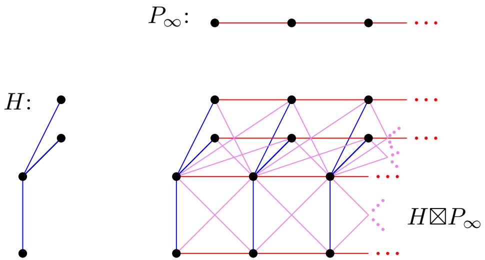
Products of graphs

- *Cartesian product* $H\Box P_\infty$:
  - $P_\infty = \langle p_1, p_2, \ldots \rangle$ (infinite path).
  - $v \in V(H) \longrightarrow \langle v \times p_1, v \times p_2, \ldots \rangle$ (*extension* of v)
  - *horizontal* edges: $(v \times p_i, v \times p_{i+1})$ for $i \geq 1$
  - *vertical* edges: $(v, w) \in E(H) \rightarrow (v \times p_i, w \times p_i)$ for $i \geq 1$



Products of graphs

- *Cartesian product* $H \square P_\infty$:
  - $P_\infty = \langle p_1, p_2, \ldots \rangle$ (infinite path).
  - $v \in V(H) \longrightarrow \langle v \times p_1, v \times p_2, \ldots \rangle$ (*extension* of v)
  - *horizontal* edges: $(v \times p_i, v \times p_{i+1})$ for $i \geq 1$
  - *vertical* edges: $(v, w) \in E(H) \rightarrow (v \times p_i, w \times p_i)$ for $i \geq 1$

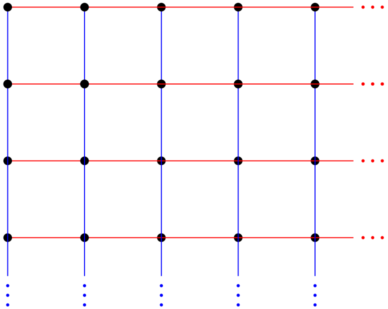


- *Strong product* $H \boxtimes P_\infty$: Cartesian product plus
  - *diagonal* edges: $(v, w) \in E(H) \rightarrow (v \times p_i, w \times p_{i+1})$ for $i \geq 1$

Examples

(We will almost only study these host-graphs.)

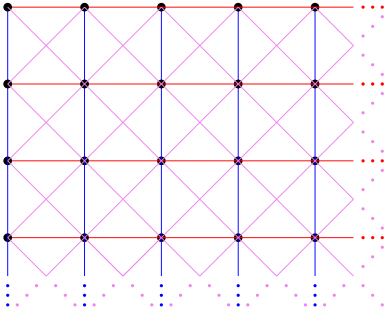
- $P_\infty \square P_\infty$ = rectangular grid



Examples

(We will almost only study these host-graphs.)

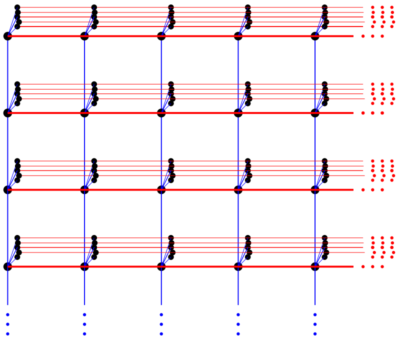
- $P_\infty \square P_\infty$ = rectangular grid
- $P_\infty \boxtimes P_\infty$ = king's graph



Examples

(We will almost only study these host-graphs.)

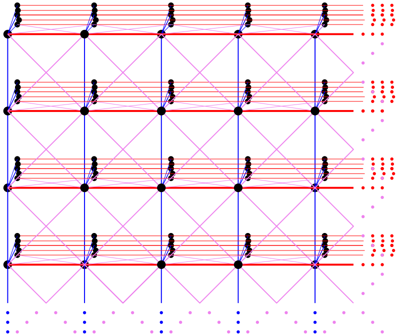
- $P_\infty \Box P_\infty$ = rectangular grid
- $P_\infty \boxtimes P_\infty$ = king's graph
- $C_\infty \Box P_\infty \approx$ grid with stuff at rows



Examples

(We will almost only study these host-graphs.)

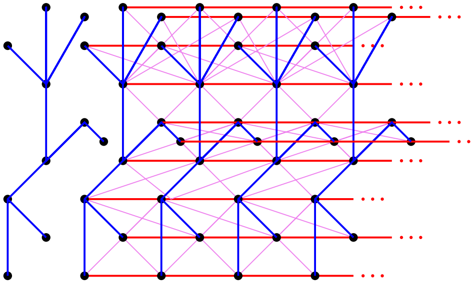
- $P_\infty \square P_\infty$ = rectangular grid
- $P_\infty \boxtimes P_\infty$ = king's graph
- $C_\infty \square P_\infty \approx$ grid with stuff at rows
- $C_\infty \boxtimes P_\infty \approx$ grid with stuff at rows



Examples

(We will almost only study these host-graphs.)

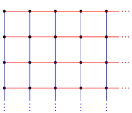
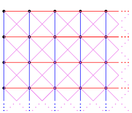
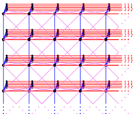
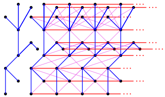
- $P_\infty \Box P_\infty$ = rectangular grid
- $P_\infty \boxtimes P_\infty$ = king's graph
- $C_\infty \Box P_\infty \approx$ grid with stuff at rows
- $C_\infty \boxtimes P_\infty \approx$ grid with stuff at rows
- $T \Box P_\infty$ and $T \boxtimes P$: hard to visualize



Problems

Given a graph G:

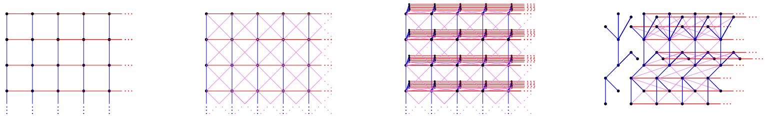
1. GridEmbedding: Is G subgraph of $P_\infty \square P_\infty$?
2. KingGraphEmbedding: Is G subgraph of $P_\infty \boxtimes P_\infty$?
3. RowPathWidth1: Does G have row-pathwidth 1?
   (Same as: Is G subgraph of $C_\infty \boxtimes P_\infty$?)
4. RowTreeWidth1: Does G have row-treewidth 1?
   (Same as: Is G subgraph of $T \boxtimes P_\infty$ for a tree T?)

Problems

Given a graph G:

1. GridEmbedding: Is G subgraph of $P_\infty \Box P_\infty$?
2. KingGraphEmbedding: Is G subgraph of $P_\infty \boxtimes P_\infty$?
3. RowPathWidth1: Does G have row-pathwidth 1?
   (Same as: Is G subgraph of $C_\infty \boxtimes P_\infty$?)
4. RowTreeWidth1: Does G have row-treewidth 1?
   (Same as: Is G subgraph of $T \boxtimes P_\infty$ for a tree T?)



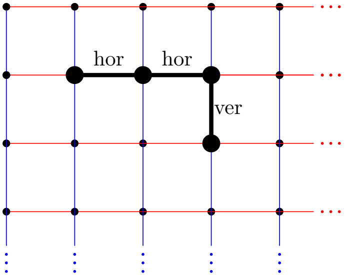
Goal: These are all NP-hard, even for very restricted graphs G.

(Well-known for (1), new for (2-4).)

Grid Embedding with Fixed Orientation

Our hardness-proofs are based on common subproblem:

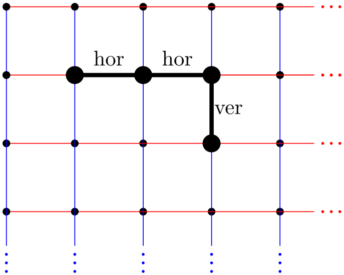
GRIDEMBEDDINGWITHFIXEDORIENTATION: Given G, edges labelled 'hor' or 'ver', is $G \subseteq P_\infty \square P_\infty$ with edges as indicated?



Grid Embedding with Fixed Orientation

Our hardness-proofs are based on common subproblem:

GRIDEMBEDDINGWITHFIXEDORIENTATION: Given G, edges labelled 'hor' or 'ver', is $G \subseteq P_\infty \Box P_\infty$ with edges as indicated?



Theorem

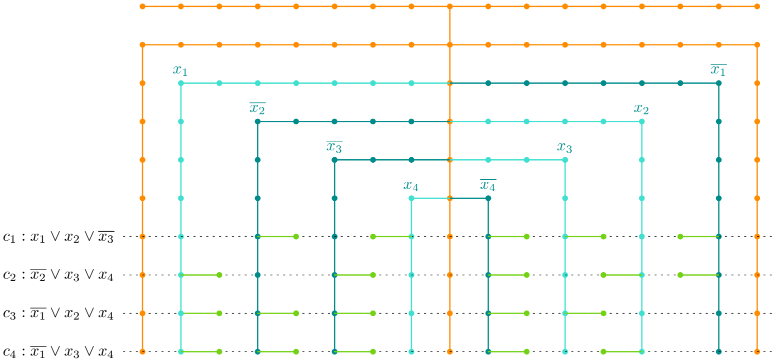
GRIDEMBEDDINGWITHFIXEDORIENTATION *is NP-hard.*

Grid Embedding with Fixed Orientation

Theorem

GridEmbeddingWithFixedOrientation *is NP-hard.*

Proof: Use *Logic Engine* (Eades, Whitesides 96)

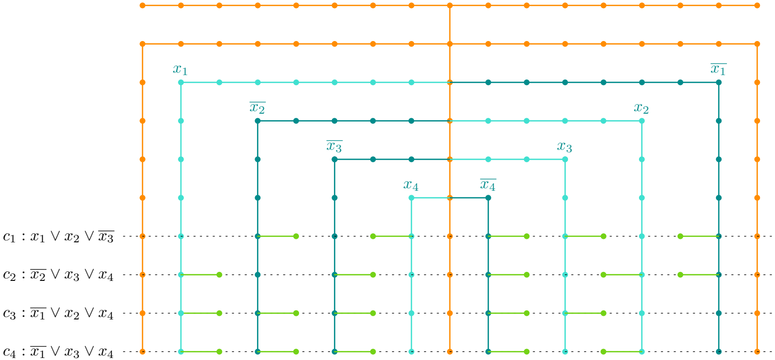


Grid Embedding with Fixed Orientation

Theorem

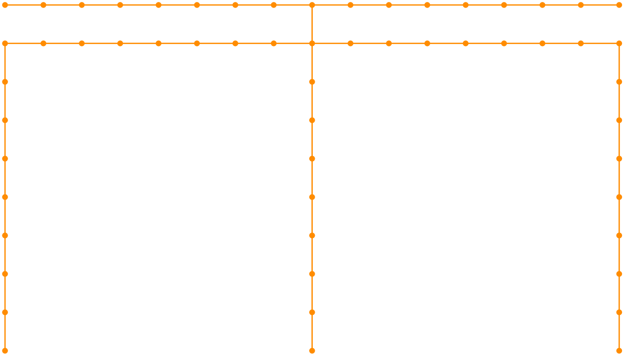
GridEmbeddingWithFixedOrientation *is NP-hard.*

Proof: Use *Logic Engine* (Eades, Whitesides 96)



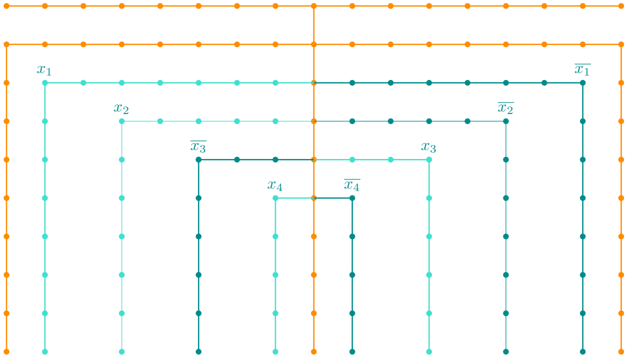
Input: NAE-3SAT instance $c_1 = x_1 \vee x_2 \vee \overline{x_3}, c_2 = \dots$

Logic Engine



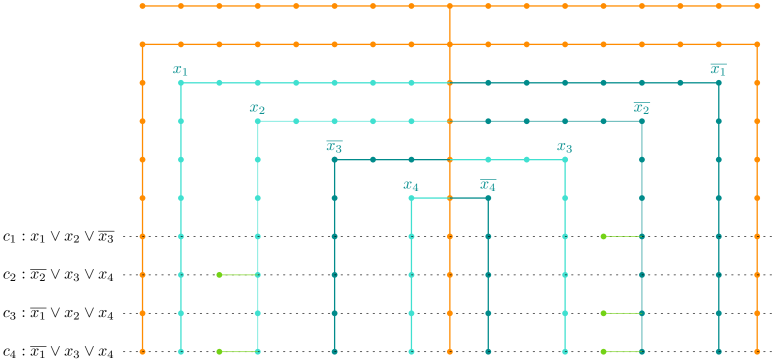
Frame: No choices up to symmetry since edge-orientations fixed.

Logic Engine



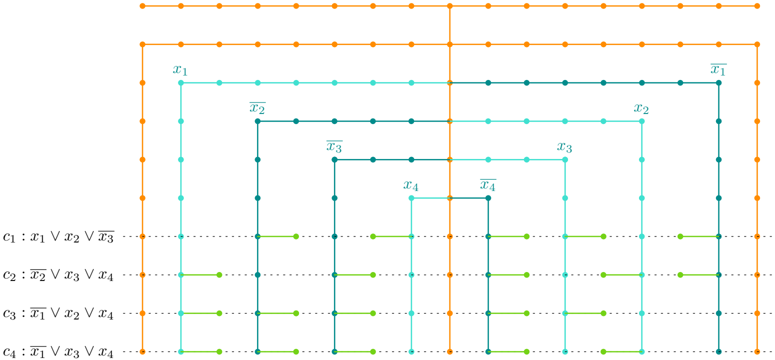
Frame: No choices up to symmetry since edge-orientations fixed.
Armature: One per variable, can flip horizontally

Logic Engine



Frame: No choices up to symmetry since edge-orientations fixed.
Armature: One per variable, can flip horizontally
Clause-rows: Frame + armature expand over one row per clause
Flags: Add if $\ell_i \notin c_j$, can flip horizontally

Logic Engine



Frame: No choices up to symmetry since edge-orientations fixed.
Armature: One per variable, can flip horizontally
Clause-rows: Frame + armature expand over one row per clause
Flags: Add if $\ell_i \notin c_j$, can flip horizontally
Easy to see: Can embed $\Leftrightarrow$ solution to NAE-3SAT.

Fixing orientations

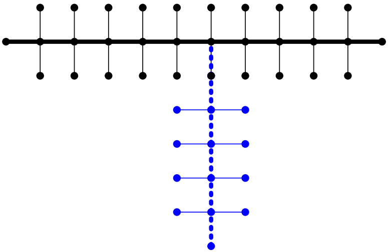
So GridEmbeddingWithFixedOrientation is NP-hard.

- Now: GridEmbedding is NP-hard.
- Idea: Modify tree so that orientations are forced (up to rotation)

Fixing orientations

So GridEmbeddingWithFixedOrientation is NP-hard.

- Now: GridEmbedding is NP-hard.
- Idea: Modify tree so that orientations are forced (up to rotation)

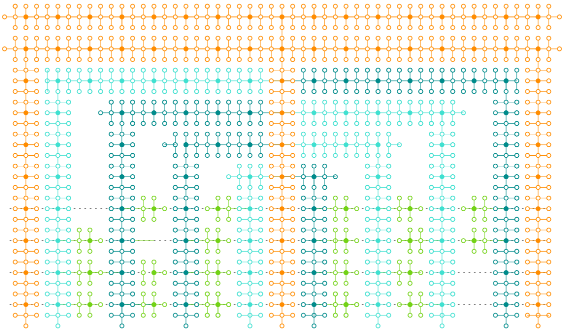


- All bold edges have same orientation.
- All dotted edges have other orientation.
- So to force orientations, turn paths into spines.

Grid Embedding

Theorem (based on (Bhatt, Cosmodakis 87))

GRIDEMBEDDING *is NP-hard even for trees.*



King Graph Embedding

Theorem

Testing whether $G \subseteq P_\infty \boxtimes P_\infty$ is NP-hard, even if G is a tree.

- **Idea 1:** Modify construction for GRIDEMBEDDING.

King Graph Embedding

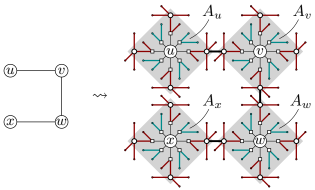
Theorem

Testing whether $G \subseteq P_\infty \boxtimes P_\infty$ is NP-hard, even if G is a tree.

- **Idea 1:** Modify construction for GRIDEMBEDDING.
- **Idea 2:** Prove a more general statement.

Theorem

Can convert any graph G into G' s.t. $G \subseteq P_\infty \square P_\infty \Leftrightarrow G' \subseteq P_\infty \boxtimes P_\infty$.



King Graph Embedding

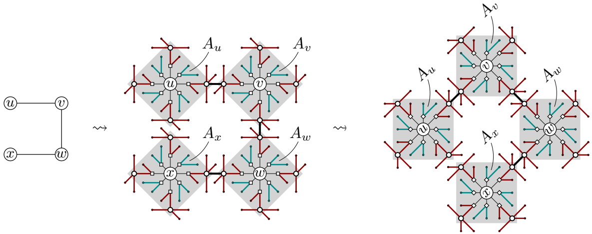
Theorem

Testing whether $G \subseteq P_\infty \boxtimes P_\infty$ is NP-hard, even if G is a tree.

- **Idea 1:** Modify construction for GRIDEMBEDDING.
- **Idea 2:** Prove a more general statement.

Theorem

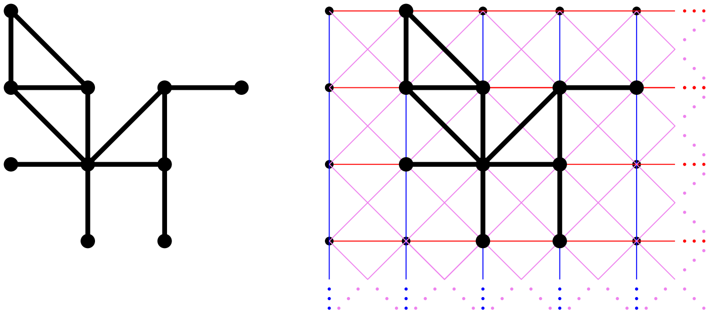
Can convert any graph G into G' s.t. $G \subseteq P_\infty \square P_\infty \Leftrightarrow G' \subseteq P_\infty \boxtimes P_\infty$.



Row pathwidth

Theorem

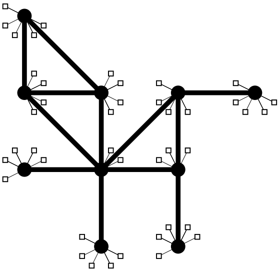
Let G be a graph. Let G' be obtained by ...
Then $G \subseteq P_\infty \boxtimes P_\infty \Leftrightarrow G' \subseteq C_\infty \boxtimes P_\infty \Leftrightarrow$ row-pathwith$(G') = 1$

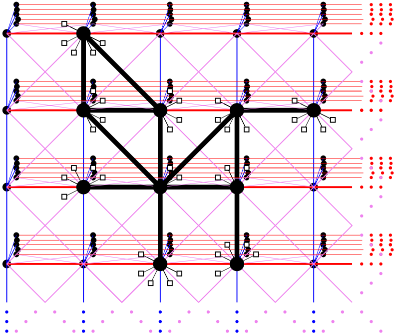


Row pathwidth

Theorem

Let G be a graph. Let G' be obtained by adding lots of leaves.
Then $G \subseteq P_\infty \boxtimes P_\infty \Leftrightarrow G' \subseteq C_\infty \boxtimes P_\infty \Leftrightarrow$ row-pathwith$(G') = 1$

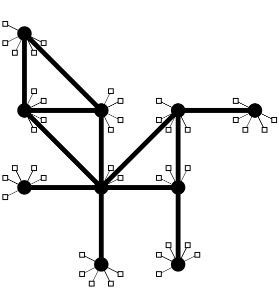


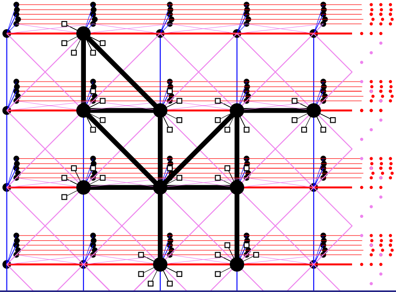


Row pathwidth

Theorem

Let G be a graph. Let G′ be obtained by adding lots of leaves.
Then $G \subseteq P_\infty \boxtimes P_\infty \Leftrightarrow G' \subseteq C_\infty \boxtimes P_\infty \Leftrightarrow \textit{row-pathwith}(G') = 1$





Corollary

Computing the row pathwidth is NP-hard, even for a tree, and even if we only want to test whether it is 1.

Onto row-treewidth

Goal: It is NP-hard to test whether $G \subseteq T \boxtimes P_\infty$ for a tree T.

Problem: Need different tool to force edge-orientations.

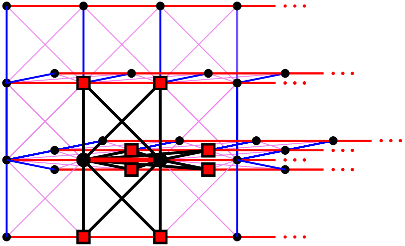
Onto row-treewidth

Goal: It is NP-hard to test whether $G \subseteq T \boxtimes P_\infty$ for a tree T.

Problem: Need different tool to force edge-orientations.

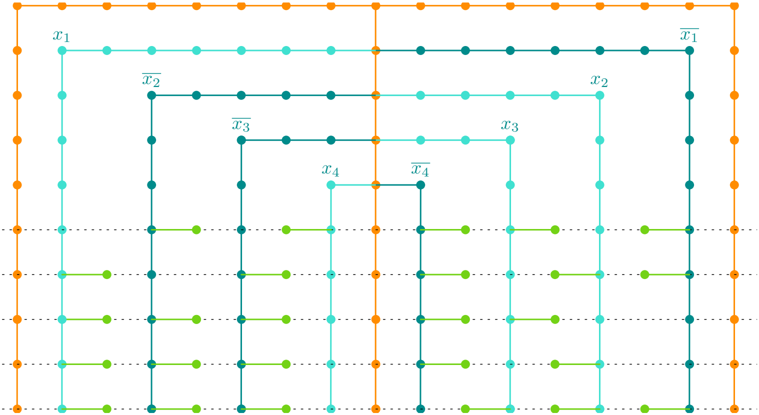
Observation

Let $e = (v, w)$ be an edge of a graph G embedded in $T \boxtimes P_\infty$.
If v, w have ≥ 5 common neighbours, then e is horizontal.



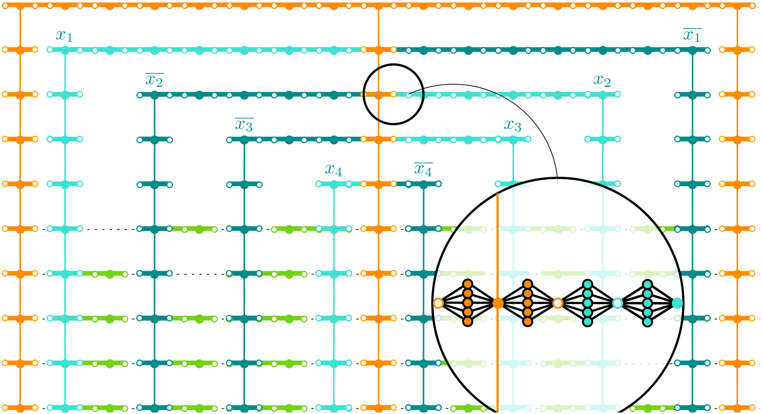
e horizontal

NP-hardness of row treewidth



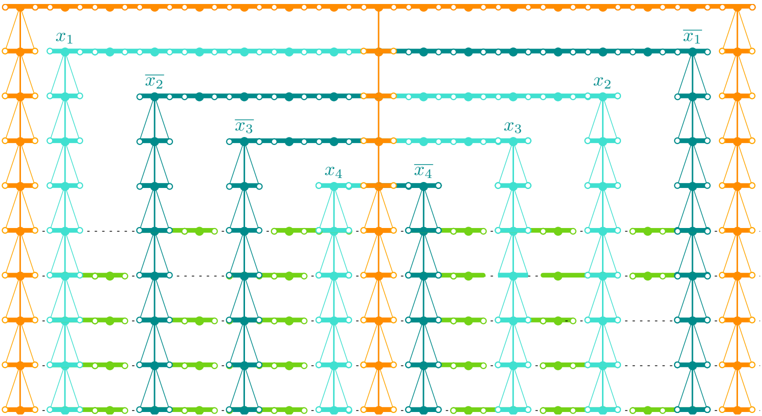
- Use G from GridEmbeddingWithFixedOrientation

NP-hardness of row treewidth



- Use G from GRIDEMBEDDINGWITHFIXEDORIENTATION
- Triple the width, add deg-2 vertices at want-to-be-horizontals

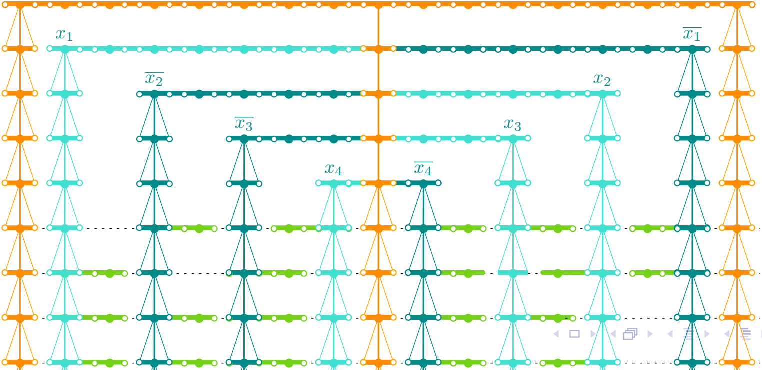
NP-hardness of row treewidth



- Use G from GridEmbeddingWithFixedOrientation
- Triple the width, add deg-2 vertices at want-to-be-horizontals
- Add two diagonals at want-to-be-vertical edges (and argue that this forces vertical)

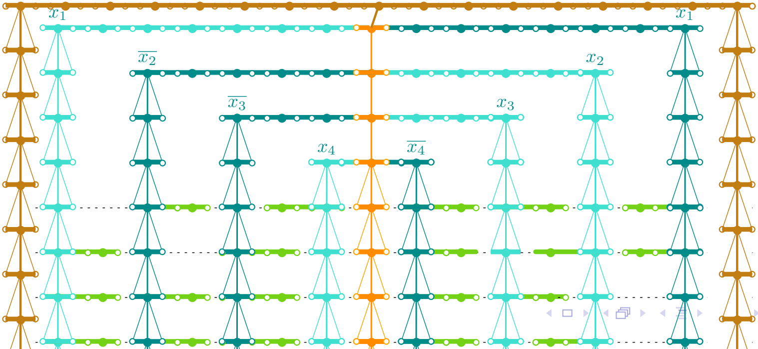
NP-hardness of row treewidth

Done? Not quite. (Frame could ‘get out of the way’.)



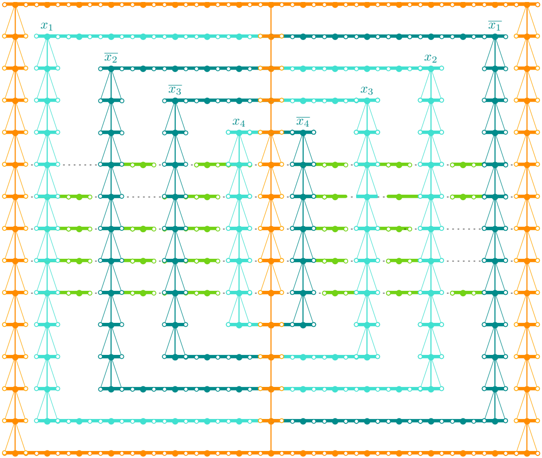
NP-hardness of row treewidth

Done? Not quite. (Frame could 'get out of the way'.)



NP-hardness of row treewidth

Done? Not quite. (Frame could 'get out of the way'.)



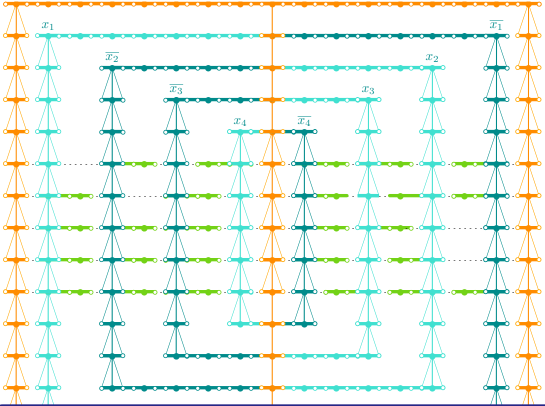
Solution: Close tines into cycles.

Show: Logic engine then projects to path.

So $G' \subseteq T \boxtimes P_\infty$ $\Leftrightarrow G \subseteq P \square P$ with fixed orientation

NP-hardness of row treewidth

Done? Not quite. (Frame could 'get out of the way'.)



Solution: Close tines into cycles.

Show: Logic engine then projects to path.

So $G' \subseteq T \boxtimes P_\infty$ $\Leftrightarrow G \subseteq P \square P$ with fixed orientation

Theorem

Computing the row treewidth of G is NP-hard, even for a planar graph, and even if we only want to test whether it is 1.

Positive results?

So: Everything is NP-hard.

What do we do if a problem is NP-hard?

Aspiration

RowTreewidth *is polynomial if G satisfies* $\langle \ldots \rangle$.
RowTreewidth *is FPT in parameter* $\langle \ldots \rangle$.

Positive results?

So: Everything is NP-hard.

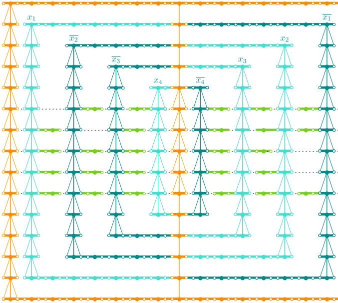
What do we do if a problem is NP-hard?

Aspiration

RowTreewidth *is polynomial if G satisfies* $\langle \ldots \rangle$.
RowTreewidth *is FPT in parameter* $\langle \ldots \rangle$.

Our construction rules out nearly everything:

- Only test whether answer is ‘1’
- Constant treewidth and pathwidth
- Constant maximum degree



Positive results?

So: Everything is NP-hard.

What do we do if a problem is NP-hard?

Aspiration

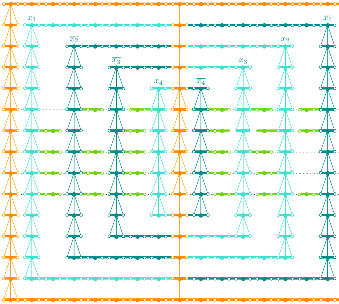
RowTreewidth *is polynomial if G satisfies* $\langle \ldots \rangle$.
RowTreewidth *is FPT in parameter* $\langle \ldots \rangle$.

Our construction rules out nearly everything:

- Only test whether answer is ‘1’
- Constant treewidth and pathwidth
- Constant maximum degree

Only few (very specialized) positive results

(see paper)



A few more (negative) results

- No $O(1)$-approximation for row treewidth and row pathwidth (under small set expansion conjecture)
- NP-hard to test whether a tree has *row treedepth* 1.
  - treedepth 1 = subgraph of star $K_{1,n}$
  - completely different reduction

A few more (negative) results

- No $O(1)$-approximation for row treewidth and row pathwidth (under small set expansion conjecture)
- NP-hard to test whether a tree has *row treedepth* 1.
  - treedepth 1 = subgraph of star $K_{1,n}$
  - completely different reduction

In summary, everything is really really hard.