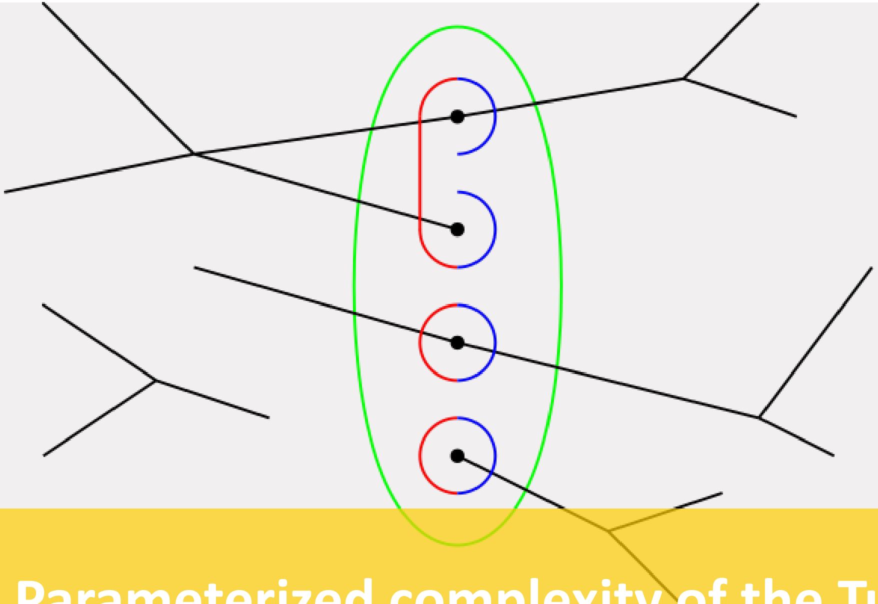
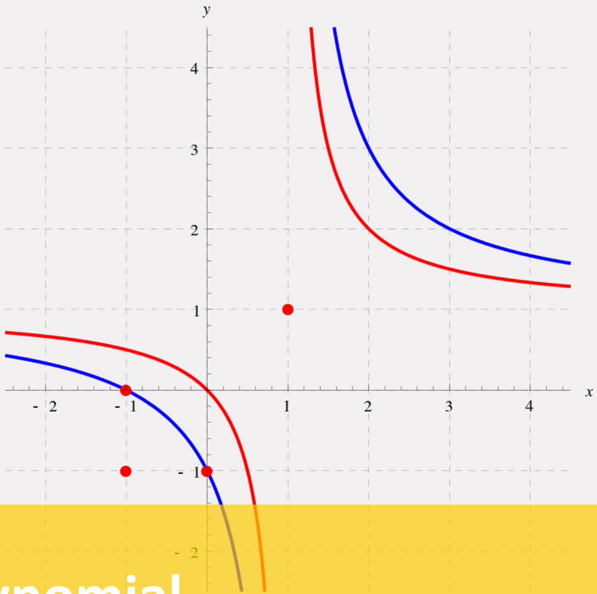





# Parameterized complexity of the Tutte polynomial

Isja Mannens, Jesper Nederlof

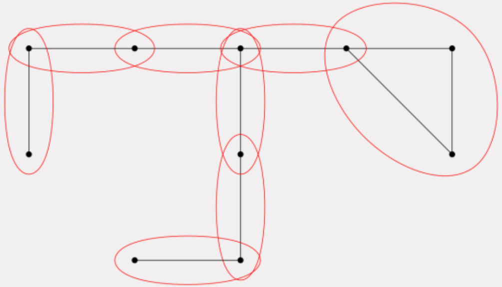
Computer Science department, Algorithms & Complexity

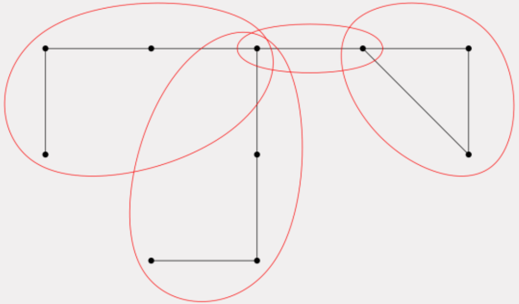
# Overview

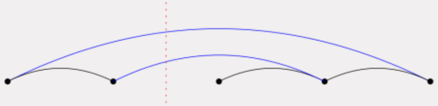
- Preliminaries
- The Tutte polynomial
- Counting forests

# Width Measures

Treewidth ≤ Pathwidth ≤ Cutwidth







**Treewidth**

- Cover the graph in bags
- Bags have a tree-like structure
- Width is given by the size of the bags

**Pathwidth**

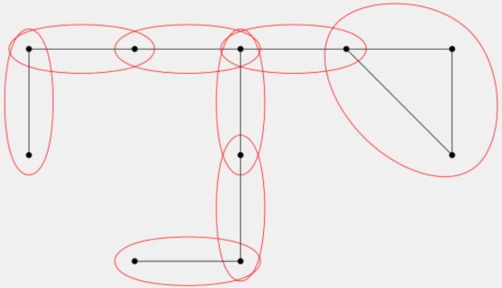
- Cover the graph in bags
- Bags have a path-like structure
- Width is given by the size of the bags

**Cutwidth**

- Order the vertices
- Width is given by #edges crossing a any cut

# Parameterized complexity

Treewidth



- Classical complexity:
  - $n^2$, $2^n$, etc.

- Parameterized complexity:
  - $2^{tw}n^{O(1)}$, $n^{tw}$, etc.
  - FPT: $f(tw)n^{O(1)}$
  - XP: $f(tw)n^{g(tw)}$

Both polynomial time!

# The Tutte polynomial

- Input:
  - A graph G
  - $(x, y) \in \mathbb{C}^2$
- Output:
  - $T(G; x, y) = \sum_{A \subseteq E} (x-1)^{k(A)-k(E)} (y-1)^{k(A)+|A|-|V|}$ (k(A) = #components of (V,A))
- How fast can we compute $T$ for a fixed pair $(x, y)$?
- Why should we care?

# The Tutte polynomial, why should we care?

- Any graph parameter that can be defined by a deletion-contraction recurrence ($f(G) = f(G\backslash \mathrm{e}) + f(G/e)$)

Some specific cases (for connected G):

- $\mathrm{T}(\mathrm{G}; 2, 1) =$ # forests
- $\mathrm{T}(\mathrm{G}; 1, 2) =$ # connected subgraphs
- $\mathrm{T}(\mathrm{G}; 1, 1) =$ # spanning trees
- Etc.

# The Tutte polynomial, NP-hardness

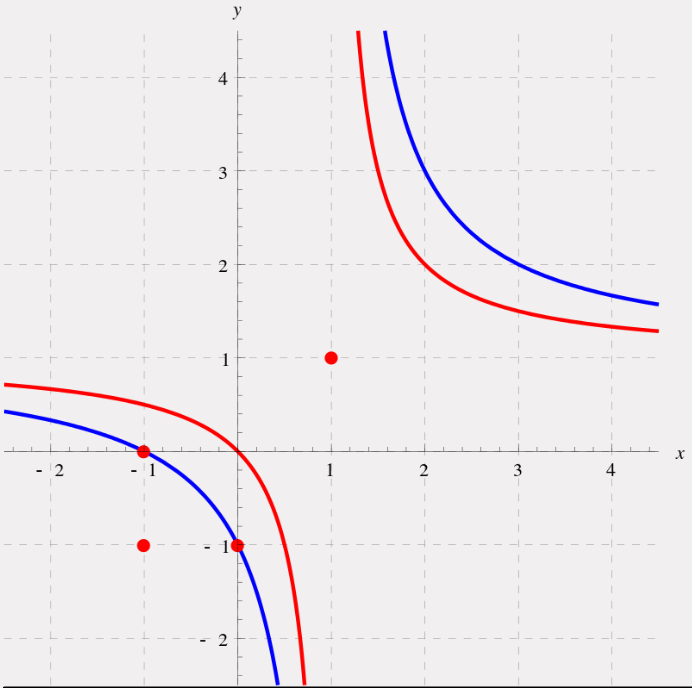
- Special curves:

$$H_\alpha = \{(x,y): (x-1)(y-1) = \alpha\}$$

Easy points (for $j = e^{\frac{2\pi i}{3}}$):

$$\{(1,1), \quad (-1,-1), \quad (0,-1), \quad (-1,0),$$
$$(i,-i), \quad (-i,i), \quad (j,j^2), \quad (j^2,j)\} \cup H_1$$

- Theorem (JVW): NP-hard everywhere else



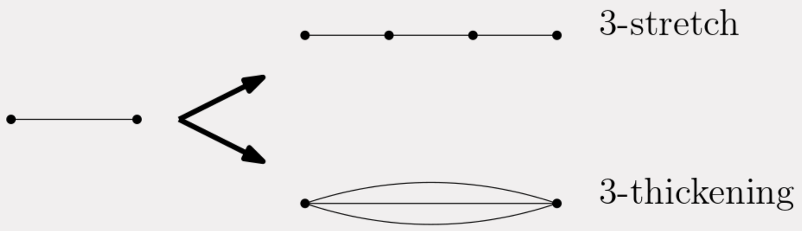

https://commons.wikimedia.org/wiki/File:Tractable_points_of_the_Tutte_polynomial_in_the_real_plane.svg

# The Tutte polynomial, NP-hardness

Theorem (JVW): NP-hard everywhere else

Proof overview:

- Idea: lift hardness from some known hard point on $H_\alpha$
- Assume $(x, y) \in H_\alpha$ in polynomial time
- Apply 'k-stretch/thickening' to compute n different points on $H_\alpha$



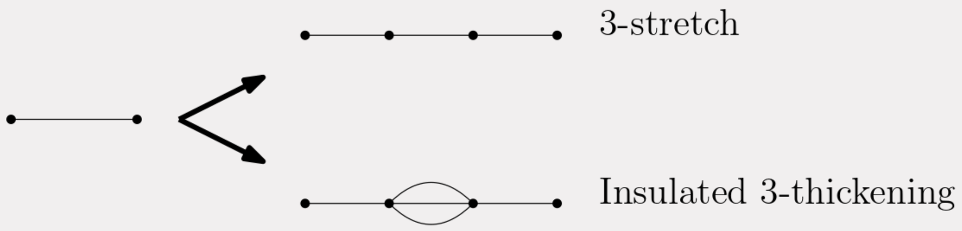

- Interpolate polynomial along $H_\alpha$

# The Tutte polynomial, Parameterized Complexity

Theorem (new): Parameterized hardness dichotomy

Proof overview:

- Idea: lift hardness from some known hard point on $H_\alpha$
- Assume $(x, y) \in H_\alpha$ in *'certain'* time
- Apply 'k-stretch/insulated k-thickening' to compute n different points on $H_\alpha$




- Interpolate polynomial along $H_\alpha$

# The Tutte polynomial, our results

- Lower bounds in terms of ctw and upper bounds in terms of tw
- Since
  $tw \leq pw \leq ctw$
  these bounds hold for all three parameters

| Curve | Lower bound | Upper bound |
|---|---|---|
| $H_0^x$ | $ctw^{o(ctw)}$ | $O(tw^{tw})$ |
| $H_0^y$ | $2^{o(ctw)}$ | $O(64^{tw})$ |
| $H_1$ | | $O(n^{O(1)})$ |
| $H_\alpha$ for $\alpha \in \mathbb{Z}_{\geq 2}$ | $O(\alpha^{ctw})$ | $O(\alpha^{tw})$ |
| $H_\alpha$ for $\alpha \in \mathbb{Z}_{<0}$ | $ctw^{o(ctw)}$ | $O(tw^{tw})$ |
| $H_\alpha$ for $\alpha \in \mathbb{C} \backslash \mathbb{Z}$ | $2^{o(ctw)}$ | $O(tw^{tw})$ |

⇧

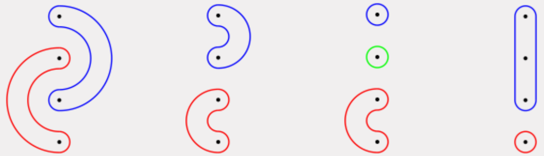
Not tight (yet)

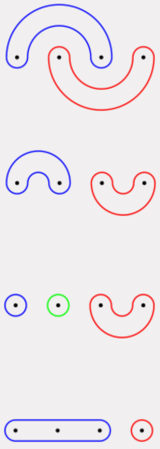
# Counting forests (T(G; 2, 1))

Problem: Count the number of edgesets $A \subseteq E$, such that $(V, A)$ is a forest

- This is a special case of the Tutte polynomial ($T(G; 2,1)$)
- We can do this in $O(64^{tw})$ time
- Surprisingly $T(G; 1,2)$ cannot be computed in $ctw^{o(ctw)}$ time
- We use the 'rank-based approach'
  - Uses some problem specific matrix M
  - Running time is (often) linear in the rank of M

# Forest Compatibility matrix





| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |

- Index by forests (partitions into connected components)
- If two forests (partitions) induce a cycle, we put a 0
- Otherwise, we put a 1

- dim(M) is the $n^{th}$ Bell number

**rank(M) is the $n^{th}$ Catalan number!**
(Non-crossing partitions form a basis of the matrix)

# Forest Compatibility matrix

General proof structure:

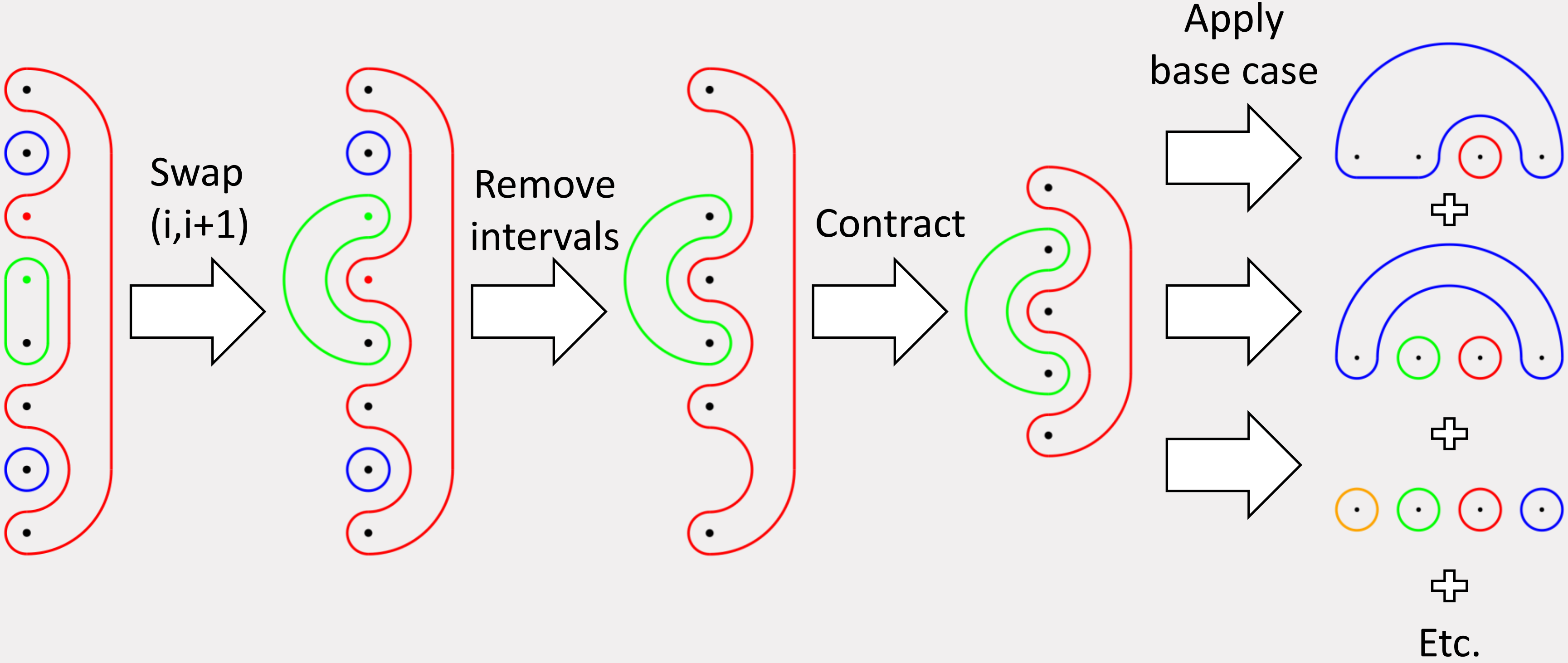
- Let $N_p$ be the set of (rows corresponding to) non-crossing partitions relative to permutation *p*
- Show that $N_{(i,i+1)} \subseteq \text{span}(N_{id})$
- By induction $N_p \subseteq \text{span}(N_{id})$ for any permutation *p*
    - Every permutation can be written as a product of 2-cycles (i, i+1)
- Any partition is non-crossing for some permutation
- Therefore, the set of non-crossing partitions span all rows

# Forest Compatibility matrix

General proof structure:

- Let $N_p$ be the set of (rows corresponding to) non-crossing partitions relative to permutation *p*
- Show that $N_{(i,i+1)} \subseteq \text{span}(N_{id})$
- By induction $N_p \subseteq \text{span}(N_{id})$ for any permutation *p*
  - Every permutation can b written as a product of 2-cycles (i, i+1)
- Any partition is non-crossing for some permutation
- Therefore, the set of non-crossing partitions span all rows

# Forest Compatibility matrix, uncrossing a swap

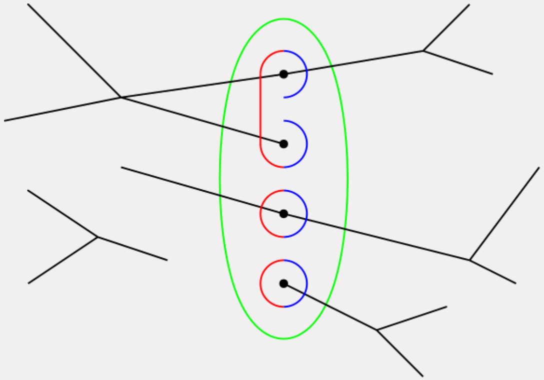


# What's next?

- Complete dichotomy
  - Sharpen bounds for $H_\alpha$ for $\alpha \in \mathbb{C}\backslash\mathbb{Z}$ and $H_0^y$
- Different Parameters?
- Dichotomy mod p

Thank you for listening!

# Counting forests (T(G; 2, 1))



v

* → #forests to the left, that agree with

w

Mv =

* → #ways to extend to the left

Reduce by finding minimum support preimage v’ of w under M, i.e. Mv’ = w