A Tight Lower Bound for Counting Hamiltonian Cycles via Matrix Rank

Radu Curticapean — BARC Copenhagen

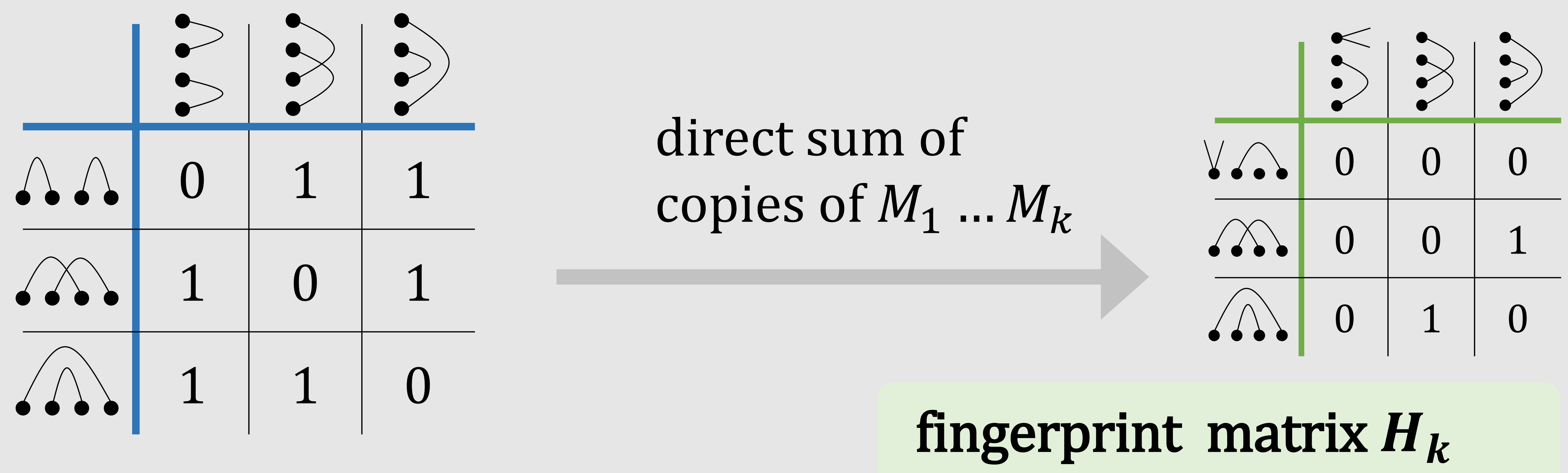
Nathan Lindzey — University of Waterloo

Jesper Nederlof — Eindhoven University of Technology

The complexity of **NP-hard problems** on **small-treewidth instances** often depends on the rank of **problem-related matrices**.

We study this for **Hamiltonian Cycles** and the **matchings connectivity matrix**.

Matchings connectivity matrix $\mathbf{M_b}$ for even $b \in \mathbb{N}$:
indexed by **perfect matchings** on b vertices,
entry at (M, M') is 1 iff $M \cup M'$ forms a **single cycle**, 0 otherwise.



fingerprint matrix H_k
over fingerprints on $[k]$
$H_k(f, f') = 1$ iff f, f' combine

$\mathbb{Z}_2$

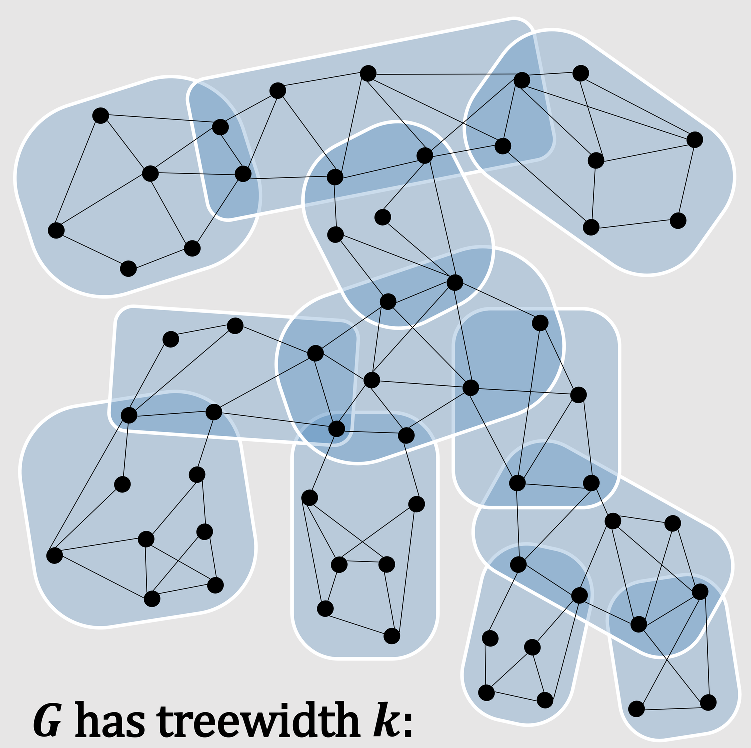
Rank of M_b over $\mathbb{Z}_2$ is $2^{b/2} - 1$. [CKN13]
Implies $O^*(3.414^{pw})$ time for **counting HamCycles mod 2**
(and for determining existence) on graphs of **pathwidth pw**.
Tight under SETH.

$\mathbb{R}$

Rank of M_b over $\mathbb{R}$ is $4^b/poly(b)$. NEW
Uses **representation theory of S_n** and **algebraic combinatorics**.

An $O^*(6^{pw})$ time algorithm for **#HamCycles** was known [BCKN13].
Via our rank bound & new reduction technique: **Tight under SETH.**

Bonus: #HamCycles mod $p \neq 2$ needs $O^*(3.57^{pw})$ time under SETH.
Compare to counting mod 2 in $O^*(3.41^{pw})$ time.



***G* has treewidth *k*:**
tree of k-sized separators,
useful for **dynamic programming**

Known DPs for Hamiltonian Cycles

	standard	refined
	$O(tw!)$	$O^*(c^{tw})$ [C+11] [CKN13] [BCKN13] [W16]
decision		$c = 2 + \sqrt{2}$ optimal (SETH)
counting		$c = 6$ (if $\omega = 2$) optimal?

Our Contributions

Thm 1: $\mathrm{rk}_{\mathbb{R}}(M_b) = \Omega^*(4^b)$, and $\mathrm{rk}_{\mathbb{R}}(H_k) = \Omega^*(6^k)$

Proof uses representation theory of the symmetric group:

Integer partition $\lambda \vdash n$

Standard Young tableau of λ
- numbers $1 \dots n$ in the boxes
- ascending in each row, column

$f(\lambda)$:= # standard Young tableaux of λ

λ is ***hook*** if $\boxplus \not\subseteq \lambda$ $\quad$ λ is ***nice*** if $\boxplus \not\subseteq \lambda$

Example Standard Young tableaux

$6 = 5 + 1$ $\quad$ $\begin{array}{|c|c|c|c|c|} \hline 1 & 2 & 3 & 4 & 5 \\ \hline 6 \\ \cline{1-1} \end{array}$

$6 = 4 + 1 + 1$ $\quad$ $\begin{array}{|c|c|c|c|} \hline 1 & 2 & 4 & 6 \\ \hline 3 \\ \cline{1-1} 5 \\ \cline{1-1} \end{array}$

$6 = 3 + 3$ $\quad$ $\begin{array}{|c|c|c|} \hline 1 & 2 & 4 \\ \hline 3 & 5 & 6 \\ \hline \end{array}$

[RZ95]: In a bipartite setting

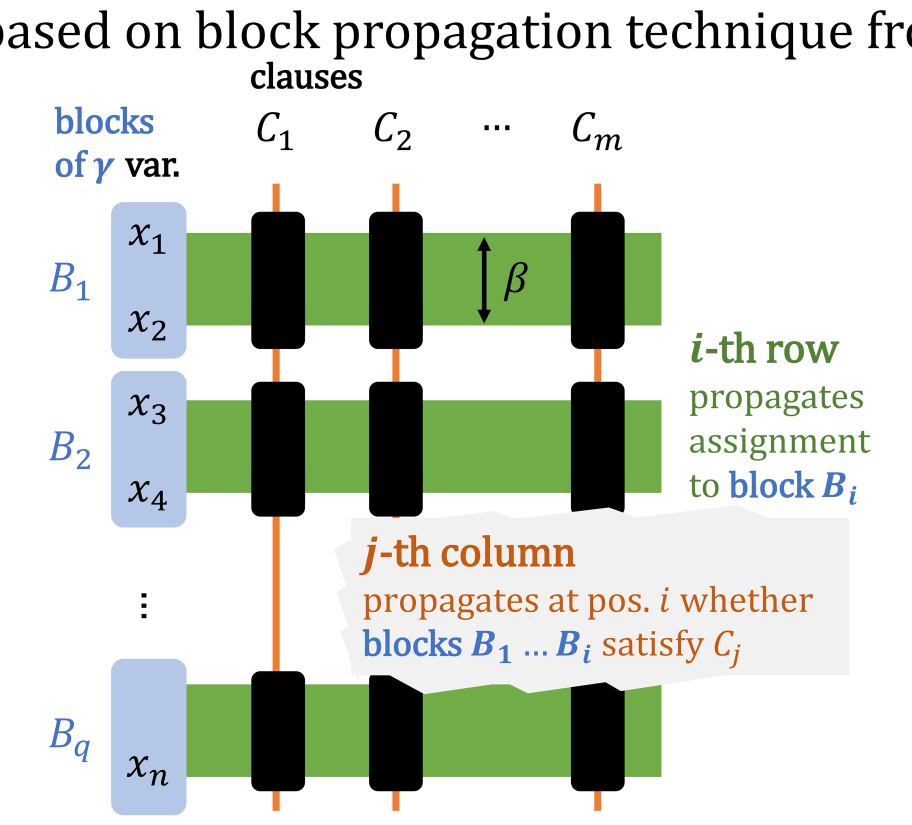
$$\mathrm{rk}_{\mathbb{R}}\, M_k^{\mathrm{bip}} = \sum_{\text{hook } \lambda \vdash k} f(\lambda)^2 = \Theta^*(2^k)$$

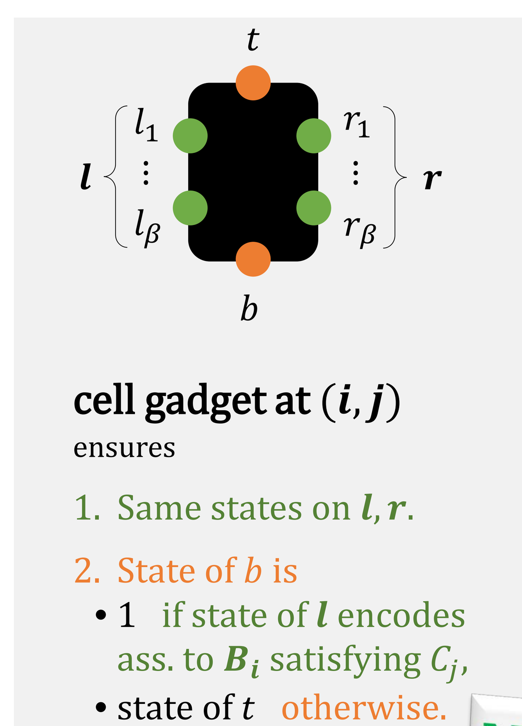
Non-bipartite setting:

$$\mathrm{rk}_{\mathbb{R}}\, M_k = \sum_{\text{nice } \lambda \vdash k/2} f(2\lambda) = \Theta^*(4^k)$$

Thm 2: If p prime and $rk_{\mathbb{Z}_p}(H_k) = \Omega(c^k)$, the number of Hamiltonian cycles cannot be counted in $O^*((2 + c - \varepsilon)^{pw})$, assuming SETH.

Proof based on block propagation technique from [LMS11]





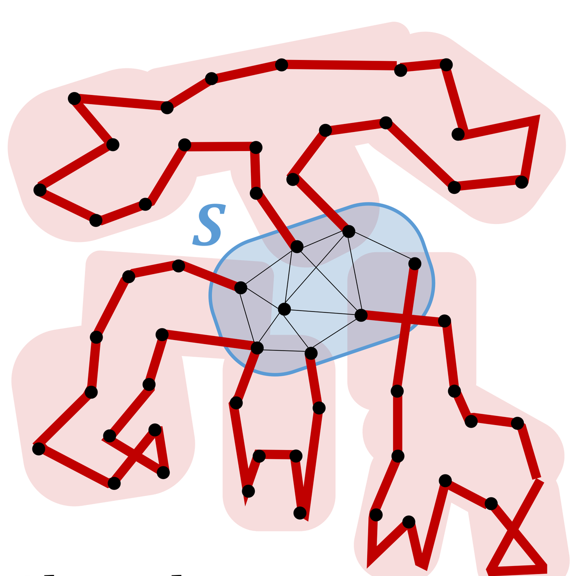
New Idea

Large invertible submatrix allows efficient encoding of partial solutions that propagate through graph due to invertibility

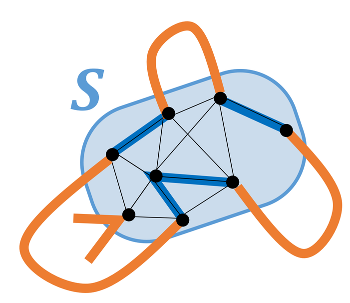
Thm 3: The number of Hamiltonian cycles cannot be computed in $O^*((6 - \varepsilon)^{pw})$, assuming SETH.

Follows from Thm1&2. Tight in the sense that an $O^*(6^{pw})$ time algorithm exists [BCKN13,W16].

Standard DP



Any **partial solution A outside S:**
vertex-disjoint union of paths,
all path endpoints in S.



fingerprint f on S
- degrees $d : S \to \{0,1,2\}$
- perfect matching M on $d^{-1}(1)$

Algorithm:
Traverse separator hierarchy bottom-up.
At separator S, store # of partial solutions below S with fingerprint f.
Total time: #fingerprints $\cdot\, n^{O(1)} \leq O^*(k^{O(k)})$

Refined DP (based on rank)

high-level idea

1. Think of standard DP as a **chain of matrix multiplications**.
   Intermediate vectors are contents of DP table.
2. Find explicit **low-rank factorizations** of these matrices.
   (Gives small set of **representative fingerprints**.)
3. Evaluate matrix multiplication chain using the factorizations.