Tight bounds for counting colorings parameterised by cutwidth via matrix ranks

Jesper Nederlof
(slides by Carla Groenland and Isja Mannens)
Utrecht University

Joint work with Isja Mannens, Jesper Nederlof and Krisztina Szilágyi
STACS 2022

May 2, 2022

Tight bounds for counting colorings parameterised by cutwidth **via matrix ranks**

Jesper Nederlof
(slides by Carla Groenland and Isja Mannens)
Utrecht University

Joint work with Isja Mannens, Jesper Nederlof and Krisztina Szilágyi
STACS 2022

May 2, 2022

Rank-based approach: General setting

Rank-based approach: General setting

- Consider a divide and conquer scheme for comp. problem PROB
- Suppose it divides global solutions in families A, B of partial solutions
  - Thus, in the *conquer* step we need to find $a \in A$ and $b \in B$ that *fit*, i.e. combine into a global solution.

Rank-based approach: General setting

- Consider a divide and conquer scheme for comp. problem PROB
- Suppose it divides global solutions in families A, B of partial solutions
  - Thus, in the *conquer* step we need to find $a \in A$ and $b \in B$ that *fit*, i.e. combine into a global solution.
- Consider the *compatibility matrix* M
  - Rows are indexed by A, columns by B
  - $M[a, b]$ is 1 if a, b combine to global solution; 0 otherwise

Rank-based approach: General setting

- Consider a divide and conquer scheme for comp. problem PROB
- Suppose it divides global solutions in families A, B of partial solutions
  - Thus, in the *conquer* step we need to find $a \in A$ and $b \in B$ that *fit*, i.e. combine into a global solution.
- Consider the *compatibility matrix* M
  - Rows are indexed by A, columns by B
  - $M[a, b]$ is 1 if a, b combine to global solution; 0 otherwise

Rank-based approach: *Number of relevant partial solutions can be reduced be at most the ...-rank of M. Moreover, with proper gadgeteering (so far always in the context of parameterization by width measures), a matching conditional lower bound can be designed.*

Rank-based approach: General setting

- Consider a divide and conquer scheme for comp. problem PROB
- Suppose it divides global solutions in families A, B of partial solutions
  - Thus, in the *conquer* step we need to find $a \in A$ and $b \in B$ that *fit*, i.e. combine into a global solution.
- Consider the *compatibility matrix* M
  - Rows are indexed by A, columns by B
  - $M[a, b]$ is 1 if a, b combine to global solution; 0 otherwise

Rank-based approach: *Number of relevant partial solutions can be reduced be at most the ...-rank of M. Moreover, with proper gadgeteering (so far always in the context of parameterization by width measures), a matching conditional lower bound can be designed.*

- For PROB, this is the *largest permutation submatrix* $lps(M)$

Rank-based approach: General setting

- Consider a divide and conquer scheme for comp. problem PROB
- Suppose it divides global solutions in families A, B of partial solutions
  - Thus, in the *conquer* step we need to find $a \in A$ and $b \in B$ that *fit*, i.e. combine into a global solution.
- Consider the *compatibility matrix* M
  - Rows are indexed by A, columns by B
  - $M[a, b]$ is 1 if a, b combine to global solution; 0 otherwise

Rank-based approach: *Number of relevant partial solutions can be reduced be at most the ...-rank of M. Moreover, with proper gadgeteering (so far always in the context of parameterization by width measures), a matching conditional lower bound can be designed.*

- For PROB, this is the *largest permutation submatrix* $lps(M)$
- For the #PROB, this is the rank of $rk(M)$ over the reals.

Rank-based approach: General setting

- Consider a divide and conquer scheme for comp. problem PROB
- Suppose it divides global solutions in families A, B of partial solutions
  - Thus, in the *conquer* step we need to find $a \in A$ and $b \in B$ that *fit*, i.e. combine into a global solution.
- Consider the *compatibility matrix* M
  - Rows are indexed by A, columns by B
  - $M[a, b]$ is 1 if a, b combine to global solution; 0 otherwise

Rank-based approach: *Number of relevant partial solutions can be reduced be at most the ...-rank of M. Moreover, with proper gadgeteering (so far always in the context of parameterization by width measures), a matching conditional lower bound can be designed.*

- For PROB, this is the *largest permutation submatrix* $lps(M)$
- For the #PROB, this is the rank of $rk(M)$ over the reals.
- For $\oplus p$-PROB, this is the rank of $rk_p(M)$ over $\mathbb{Z}_p$.

Rank-based approach: Specific cases

Rank-based approach: Specific cases

Gian-Carlo Rota: "Publish the same result several times".

Rank-based approach: Specific cases

Gian-Carlo Rota: "Publish the same result several times".

Forest connectivity matrix M_{for}^{k}:

Rank-based approach: Specific cases

Gian-Carlo Rota: "Publish the same result several times".

Forest connectivity matrix M^k_{for}:

- Rows and columns are indexed by forests on vertex set $[k]$. Matrix indicates whether the (multi-set) union forms a tree.

Rank-based approach: Specific cases

Gian-Carlo Rota: "Publish the same result several times".

Forest connectivity matrix M^k_{for}:

- Rows and columns are indexed by forests on vertex set $[k]$. Matrix indicates whether the (multi-set) union forms a tree.

  [CNPPRW'11] $rk_2(M^n_{for}) = 2^{k-1} \rightarrow$ single-exponential runtime for many connectivity problems parameterized by tree-width.

Rank-based approach: Specific cases

Gian-Carlo Rota: "Publish the same result several times".

Forest connectivity matrix M^k_{for}:

- Rows and columns are indexed by forests on vertex set $[k]$. Matrix indicates whether the (multi-set) union forms a tree.

  [CNPPRW'11] $rk_2(M^n_{for}) = 2^{k-1} \rightarrow$ single-exponential runtime for many connectivity problems parameterized by tree-width.

  [BCKN'13] $rk(M^n_{for}) = 4^k \rightarrow$ matching counting algo's

Rank-based approach: Specific cases

Gian-Carlo Rota: "Publish the same result several times".

Forest connectivity matrix M^k_{for}:

- Rows and columns are indexed by forests on vertex set $[k]$. Matrix indicates whether the (multi-set) union forms a tree.

  [CNPPRW'11] $rk_2(M^n_{for}) = 2^{k-1} \rightarrow$ single-exponential runtime for many connectivity problems parameterized by tree-width.

  [BCKN'13] $rk(M^n_{for}) = 4^k \rightarrow$ matching counting algo's

Matchings connectivity matrix M^k_{match}

Rank-based approach: Specific cases

Gian-Carlo Rota: "Publish the same result several times".

Forest connectivity matrix M^k_{for}:

- Rows and columns are indexed by forests on vertex set $[k]$. Matrix indicates whether the (multi-set) union forms a tree.

  [CNPPRW'11] $rk_2(M^n_{for}) = 2^{k-1} \rightarrow$ single-exponential runtime for many connectivity problems parameterized by tree-width.

  [BCKN'13] $rk(M^n_{for}) = 4^k \rightarrow$ matching counting algo's

Matchings connectivity matrix M^k_{match}

- Rows and columns indexed by perfect matchings on vertex set $[k]$. Matrix indicated whether the union forms a single cycle.

Rank-based approach: Specific cases

Gian-Carlo Rota: "Publish the same result several times".

Forest connectivity matrix M^k_{for}:

- Rows and columns are indexed by forests on vertex set $[k]$. Matrix indicates whether the (multi-set) union forms a tree.

  [CNPPRW'11] $rk_2(M^n_{for}) = 2^{k-1}$ $\rightarrow$ single-exponential runtime for many connectivity problems parameterized by tree-width.

  [BCKN'13] $rk(M^n_{for}) = 4^k$ $\rightarrow$ matching counting algo's

Matchings connectivity matrix M^k_{match}

- Rows and columns indexed by perfect matchings on vertex set $[k]$. Matrix indicated whether the union forms a single cycle.

  [CKN'13] $rk_2(M^k_{match}) = lps(M^k_{match}) = 2^{k/2-1}$ $\rightarrow$ tight upper and lower bound for $\textsc{Ham Cyc}/pw$ and $\oplus$2-$\textsc{Ham Cyc}/pw$.

Rank-based approach: Specific cases

Gian-Carlo Rota: "Publish the same result several times".

Forest connectivity matrix M^k_{for}:

- Rows and columns are indexed by forests on vertex set $[k]$. Matrix indicates whether the (multi-set) union forms a tree.

  [CNPPRW'11] $rk_2(M^n_{for}) = 2^{k-1} \rightarrow$ single-exponential runtime for many connectivity problems parameterized by tree-width.

  [BCKN'13] $rk(M^n_{for}) = 4^k \rightarrow$ matching counting algo's

Matchings connectivity matrix M^k_{match}

- Rows and columns indexed by perfect matchings on vertex set $[k]$. Matrix indicated whether the union forms a single cycle.

  [CKN'13] $rk_2(M^k_{match}) = lps(M^k_{match}) = 2^{k/2-1} \rightarrow$ tight upper and lower bound for $\textsc{Ham Cyc}/pw$ and $\oplus 2\text{-}\textsc{Ham Cyc}/pw$.

  [CLN'18] $rk(M^k_{match}) \geq 4^k \rightarrow$ tight lower bound for $\#\textsc{Ham Cyc}/pw$.

Tight bounds **for counting colorings** parameterised by cutwidth **via matrix ranks**

Jesper Nederlof
(slides by Carla Groenland and Isja Mannens)
Utrecht University

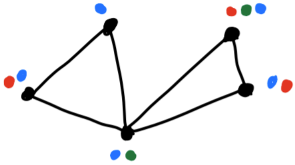
Joint work with Isja Mannens, Jesper Nederlof and Krisztina Szilágyi
STACS 2022

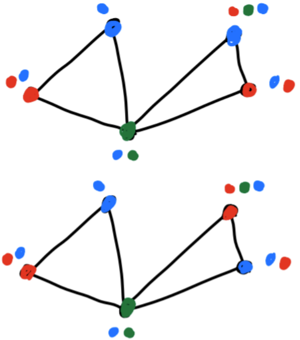
May 2, 2022

q-list coloring

Given: Graph $G = (V, E)$ and for each $v \in V$, a list $L(v) \subseteq \{1, \dots, q\}$.

Want: $c(v) \in L(v)$ for all $v \in V$ such that $c(u) \neq c(v)$ for $uv \in E$.





#q-LIST COL MOD p

Given. G graph, lists $L(v) \subseteq \{1, \dots, q\}$ for all $v \in V(G)$, $k \in \mathbb{F}_p$.

Output. What is the number of list colorings of G modulo p?

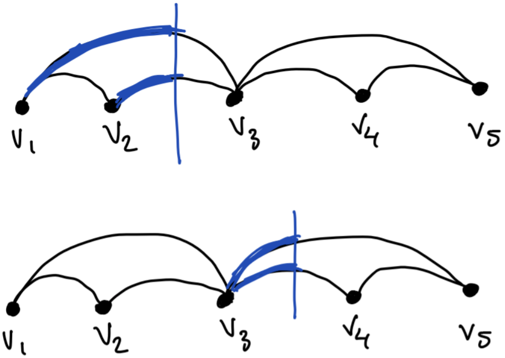
Tight bounds for counting colorings parameterised by cutwidth via matrix ranks

Jesper Nederlof
(slides by Carla Groenland and Isja Mannens)
Utrecht University

Joint work with Isja Mannens, Jesper Nederlof and Krisztina Szilágyi
STACS 2022

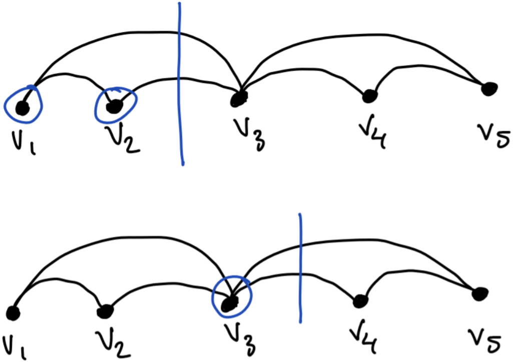
May 2, 2022

Cutwidth



$\max_\sigma \min_i$ # edges crossing ith cut

Pathwidth



$-1 + \max_\sigma \min_i$ # left endpoints of edges crossing ith cut

Tight bounds for counting colorings parameterised by cutwidth via matrix ranks

Jesper Nederlof
(slides by Carla Groenland and Isja Mannens)
Utrecht University

Joint work with Isja Mannens, Jesper Nederlof and Krisztina Szilágyi
STACS 2022

May 2, 2022

Hypothesis used for Lower bound

Strong Exponential Time Hypothesis (SETH)

For all $\epsilon > 0$, there is some $k \geq 3$ such that k-SAT cannot be solved in time $O((2-\epsilon)^n)$.

q-coloring parameterised by width

q-COL: does a given graph admit a q-coloring?

Treewidth. Dynamic programming: $q^{\mathrm{tw}}\mathrm{poly}(n)$.

Also extends to $\#q$-LIST COL.

No $(q-\epsilon)^{\mathrm{tw}}\mathrm{poly}(n)$ algorithm under SETH.

(Lokshtanov, Marx and Saurabh, 2011)

Cutwidth. Randomised algorithm: $2^{\mathrm{ctw}}\mathrm{poly}(n)$.

Does not extend to $\#q$-LIST COL.

No $(2-\epsilon)^{\mathrm{ctw}}\mathrm{poly}(n)$ algorithm under SETH.

(Nederlof, Jansen, 2018)

Present paper builds on some insights from this paper

Our result

For n-vertex graphs of cutwidth ctw, there is an algorithm running in time

$$\begin{cases} q^{ctw}\mathrm{poly}(n) & \text{if } p \text{ does not divide } q-1, \\ (q-1)^{ctw}\mathrm{poly}(n) & \text{if } p \text{ divides } q-1. \end{cases}$$

Furthermore, under SETH there is no $(q-\epsilon)^{ctw}\mathrm{poly}(n)$ resp. $(q-1-\epsilon)^{ctw}\mathrm{poly}(n)$ algorithms in these cases.

Coloring compatibility matrix

$\mathrm{col}(X)$ = set of list q-colorings of $G[X]$.

Bipartite graph (X, Y, E), $x \in \mathrm{col}(X)$ and $y \in \mathrm{col}(Y)$.
Compatible: $x \sim y$ if $x(u) \neq y(v)$ for all $uv \in E$.

Coloring compatibility matrix

$\mathrm{col}(X)$ = set of list q-colorings of $G[X]$.

Bipartite graph (X, Y, E), $x \in \mathrm{col}(X)$ and $y \in \mathrm{col}(Y)$.
Compatible: $x \sim y$ if $x(u) \neq y(v)$ for all $uv \in E$.

Coloring compatibility matrix

$$M[x, y] = \begin{cases} 1 & \text{if } x \sim y, \\ 0 & \text{if } x \not\sim y. \end{cases}$$

Coloring compatibility matrix

$\mathrm{col}(X)$ = set of list q-colorings of $G[X]$.

Bipartite graph (X, Y, E), $x \in \mathrm{col}(X)$ and $y \in \mathrm{col}(Y)$.
Compatible: $x \sim y$ if $x(u) \neq y(v)$ for all $uv \in E$.

Coloring compatibility matrix

$$M[x, y] = \begin{cases} 1 & \text{if } x \sim y, \\ 0 & \text{if } x \not\sim y. \end{cases}$$

$rk_p(M)$ depends on whether p divides $q - 1$.

Rank of coloring compatibility matrix

Bipartite graph (X, Y, E), for $x \in \mathrm{col}(X)$ and $y \in \mathrm{col}(Y)$ q-colorings

$$M[x, y] = \begin{cases} 1 & \text{if } x \sim y, \\ 0 & \text{if } x \not\sim y. \end{cases}$$

$rk_p(M) \leq (q-1)^{|E|}$ if p divides $q-1$ and $\leq q^{|E|}$ otherwise.

Rank of coloring compatibility matrix

Bipartite graph (X, Y, E), for $x \in \mathrm{col}(X)$ and $y \in \mathrm{col}(Y)$ q-colorings

$$M[x,y] = \begin{cases} 1 & \text{if } x \sim y, \\ 0 & \text{if } x \not\sim y. \end{cases}$$

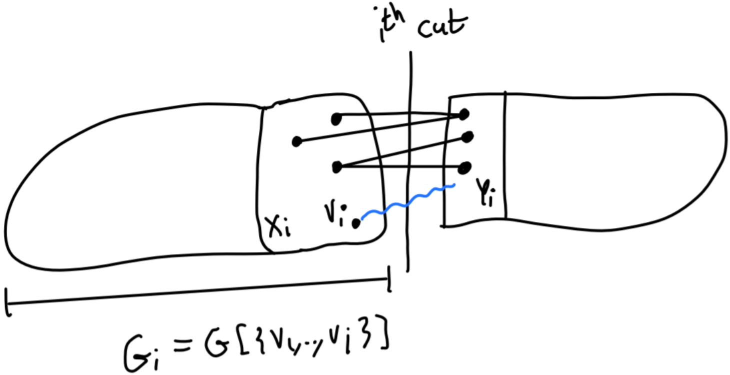
$rk_p(M) \leq (q-1)^{|E|}$ if p divides $q-1$ and $\leq q^{|E|}$ otherwise.

$$\begin{array}{c c} & \begin{array}{cccc} y(v)=1 & y(v)=2 & y(v)=3 & y(v)=4 \end{array} \\ \begin{array}{c} x(u)=1 \\ x(u)=2 \\ x(u)=3 \\ x(u)=4 \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Cutwidth order $v_1, \ldots, v_n$.

$G_i = G[\{v_1, \ldots, v_i\}]$, the ith cut gives a bipartite graph (L_i, R_i).

$X_i = L_i \cup \{v_i\}$, $Y_i = N[X_i] = R_i$.



Dynamic programming

Cutwidth order $v_1, \ldots, v_n$.

$G_i = G[\{v_1, \ldots, v_i\}]$, the ith cut gives a bipartite graph (L_i, R_i).

$X_i = L_i \cup \{v_i\}, \; Y_i = N[X_i] = R_i$.

For $x \in \mathrm{col}(X_i)$,

$$\begin{aligned}
T_i[x] &= \text{number of extensions of } x \text{ to } G_i \\
&= \text{number of } c \in \mathrm{col}(G_i - X_i) \text{ with } c \sim x \\
&= \sum_{c \in \mathrm{col}(G_i - X_i) : c \sim x} T_{i-1}[z_{|X_{i-1}}].
\end{aligned}$$

Dynamic programming

Cutwidth order $v_1, \ldots, v_n$.

$G_i = G[\{v_1, \ldots, v_i\}]$, the ith cut gives a bipartite graph (L_i, R_i).

$X_i = L_i \cup \{v_i\}, \;\; Y_i = N[X_i] = R_i$.

For $x \in \mathrm{col}(X_i)$,

$$\begin{aligned} T_i[x] &= \text{number of extensions of } x \text{ to } G_i \\ &= \text{number of } c \in \mathrm{col}(G_i - X_i) \text{ with } c \sim x \\ &= \sum_{c \in \mathrm{col}(G_i - X_i) : c \sim x} T_{i-1}[z_{|X_{i-1}}]. \end{aligned}$$

Table size $|\mathrm{col}(X_i)| = q^{|X_i|} \leq q^{ctw+1}$.

Representative sets

Idea. *Compute smaller table 'representative' of bigger table.*

Coloring compatibility matrix M_i of bipartite graph on between X_i and Y_i.
For $y \in \mathrm{col}(Y_i)$,

$$\sum_{x \in \mathrm{col}(X_i)} T_i[x] M_i[x, y] = (T_i^t M_i)[y]$$

gives the number of $c \in \mathrm{col}(G_i)$ compatible with y.

Representative sets

Idea. *Compute smaller table 'representative' of bigger table.*

Coloring compatibility matrix M_i of bipartite graph on between X_i and Y_i. For $y \in \mathrm{col}(Y_i)$,

$$\sum_{x \in \mathrm{col}(X_i)} T_i[x] M_i[x, y] = (T_i^t M_i)[y]$$

gives the number of $c \in \mathrm{col}(G_i)$ compatible with y.

T_i' is M_i-**representative** for T_i if $T_i^t M_i \equiv_p (T_i')^t M_i$.

Reducing the table size

T_i' is M_i-representative for T_i if $T_i^t M_i \equiv_p (T_i')^t M_i$.

If we know $T_i'[x] = 0$, we do not need to compute it.

Linear algebra $\implies$ there exists such T_i' with $|\mathrm{supp}(T_i')| \leq \mathrm{rank}_p(M_i)$.

Reducing the table size

T_i' is M_i-representative for T_i if $T_i^t M_i \equiv_p (T_i')^t M_i$.

If we know $T_i'[x] = 0$, we do not need to compute it.

Linear algebra $\implies$ there exists such T_i' with $|\mathrm{supp}(T_i')| \leq \mathrm{rank}_p(M_i)$.

	1	2	3
$1 : a$	0	1	1
$2 : b$	1	0	1
$3 : c$	1	1	0

	1	2	3
$1 : a + c$	0	1	1
$2 : b + c$	1	0	1
$3 : 0$	1	1	0

Reducing the table size

T_i' is M_i-representative for T_i if $T_i^t M_i \equiv_p (T_i')^t M_i$.

If we know $T_i'[x] = 0$, we do not need to compute it.

Linear algebra $\implies$ there exists such T_i' with $|\mathrm{supp}(T_i')| \leq \mathrm{rank}_p(M_i)$.

	1	2	3
$1 : a$	0	1	1
$2 : b$	1	0	1
$3 : c$	1	1	0

	1	2	3
$1 : a + c$	0	1	1
$2 : b + c$	1	0	1
$3 : 0$	1	1	0

- Can we efficiently compute T_i'?
- Can we exploit the zeros?

Reducing the table size

T'_i is M_i-representative for T_i if $T_i^t M_i \equiv_p (T'_i)^t M_i$.

If we know $T'_i[x] = 0$, we do not need to compute it.

Linear algebra $\implies$ there exists such T'_i with $|\mathrm{supp}(T'_i)| \leq \mathrm{rank}_p(M_i)$.

	1	2	3
$1 : a$	0	1	1
$2 : b$	1	0	1
$3 : c$	1	1	0

	1	2	3
$1 : a + c$	0	1	1
$2 : b + c$	1	0	1
$3 : 0$	1	1	0

- Can we efficiently compute T'_i?
- Can we exploit the zeros?

Maintain $R \subseteq X_i$ such that $T'_i[x] = 0$ if $x(v) = q$ for some $v \in R$.

T'_i *fully reduced* if $\{v \in X_i : \deg(v) = 1\} \subseteq R$.

Table size

Number of $x : X \to \{1, \ldots, q\}$ with $x(v) \neq q$ for all $v \in R$ is at most

$$(q-1)^{|R|} q^{|X|-|R|}.$$

Table size

Number of $x : X \to \{1, \ldots, q\}$ with $x(v) \neq q$ for all $v \in R$ is at most

$$(q-1)^{|R|} q^{|X|-|R|}.$$

For each x we have a table entry $T_i[x]$.
Since $|X \setminus R| \leq \frac{1}{2}(ctw - |R|)$, this is at most $(q-1)^{ctw}$.

Upper bound sketch

Algorithm. Initialise for $i = 1$. For $i = 2, \dots, n$,

- Ensure T'_{i-1} fully reduced.
- Compute T'_i from T'_{i-1}.

Upper bound sketch

Algorithm. Initialise for $i = 1$. For $i = 2, \ldots, n$,

- Ensure T'_{i-1} fully reduced.
- Compute T'_i from T'_{i-1}.

Lemma 1. If $v \in X_i$ degree 1, then can compute T' representative for T with set of reduced vertices $R \cup \{v\}$ in time $O((q-1)^{|R|} q^{|X_i|-|R|})$.

(Proof of Lemma 1 uses p divides $q-1$.)

Upper bound sketch

Algorithm. Initialise for $i = 1$. For $i = 2, \ldots, n$,

- Ensure T'_{i-1} fully reduced.
- Compute T'_i from T'_{i-1}.

Lemma 1. If $v \in X_i$ degree 1, then can compute T' representative for T with set of reduced vertices $R \cup \{v\}$ in time $O((q-1)^{|R|} q^{|X_i|-|R|})$.

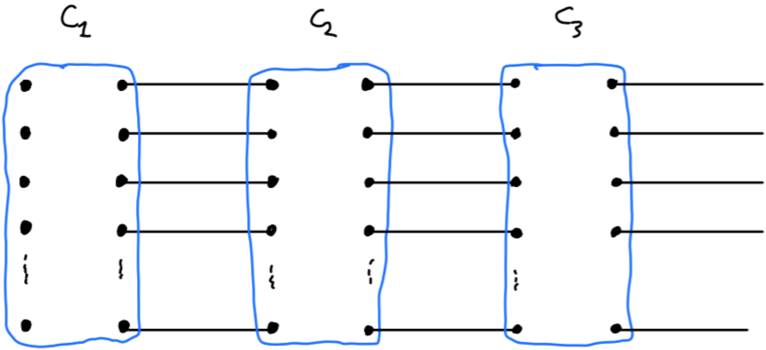
(Proof of Lemma 1 uses p divides $q-1$.)

Lemma 2. If T'_{i-1} representative for T_{i-1} and fully reduced, then can compute T'_i representative for T_i in time $O((q-1)^{ctw})$.

Lower bound

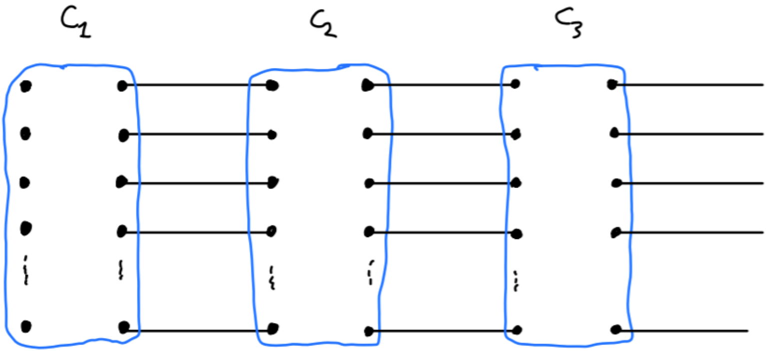
Under SETH, $\#\mathrm{CSP}(q, r)$ mod p cannot be solved in $(q-\epsilon)^n \mathrm{poly}(n, m)$ for some r [Lampis, '20 (and others?)]

n variables, m constraints
Constraints $\{1, \ldots, q\}^r \to \{0, 1\}$ depend on at most r variables.
Count number of satisfying assignments mod p.



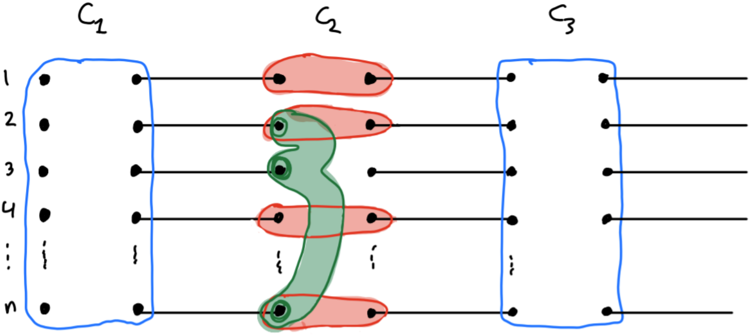
One column per constraint; one row per variable.

Cutwidth $= n + O(1)$, number of vertices $= \mathrm{poly}(n, m)$.

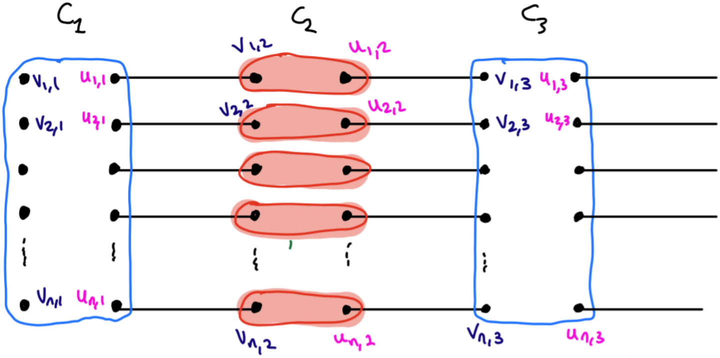


Want: number of q-colorings equals number of satisfying assignments.

"Identify a q-coloring with an assignment; check and copy."



Constraint 2 depends on variables 2, 3, n



Colour of $v_{i,2}$ = colour of $v_{i,3}$ for all i

Exploiting invertibility

Fix c_2, c_3 colorings of $v_{1,2}$ and $v_{1,3}$ respectively. The number of extensions of these colorings to the red graph equals

Exploiting invertibility

Fix c_2, c_3 colorings of $v_{1,2}$ and $v_{1,3}$ respectively. The number of extensions of these colorings to the red graph equals

$$\sum_{c_2'} f[c_2, c_2']M[c_2', c_3] = fM = M^{-1}M = I,$$

if we can set $f[c_2, c_2'] = M^{-1}[c_2', c_2]$.

Counting connected edge sets

Given $G = (V, E)$, how many $X \subseteq E(G)$ are there for which (V, X) is connected?

Tutte polynomial can link this to number of q-colorings.

$\implies$ under SETH there is no $(p - \epsilon)^{ctw}\text{poly}(n)$ algorithm.

'Correct' running times: $p^{ctw}\text{poly}(n)$, $p^{pw}\text{poly}(n)$, $p^{tw}\text{poly}(n)$.

Summary

Running time for counting modulo p the number of q-list colorings of n-vertex graphs of cutwidth ctw is

$$\begin{cases} q^{ctw}\text{poly}(n) & \text{if } p \text{ does not divide } q-1, \\ (q-1)^{ctw}\text{poly}(n) & \text{if } p \text{ divides } q-1. \end{cases}$$

The rank over $\mathbb{F}_p$ of the matrix $J_q - I_q$ (zeros on the diagonal, ones everywhere else) is

$$\begin{cases} q & \text{if } p \text{ does not divide } q-1, \\ q-1 & \text{if } p \text{ divides } q-1. \end{cases}$$