Constructive algorithm for pathwidth of matroids

Eunjung KIM, CNRS / University Paris-Dauphine
jointwork with Sang-il Oum and Jisu Jeong (KAIST)

Workshop on Satisfiability Lower bounds and Tight Results for Parameterized and Exponential Time algorithm
Background
Matroid: primer

- \((E, I)\) - a ground set and a family of subsets of \(E\) called the independent sets - satisfying:

A. \(\emptyset \in I\)

B. \(X \subset Y\) and \(Y \in I \Rightarrow X \in I\)

C. \(\forall X, Y \in I\) with \(|X| < |Y|\),

\[\exists y \in Y\) s.t. \(X \cup \{y\} \in I\)

- A matroid \((E, I)\) is representable in \(F\) if vectors in a vector space over \(F\)

... such that \(X \in I\) iff corresponding vectors are independent.
Vectors Arrangement

Input: vectors $v_1, v_2, \ldots, v_n \in \mathbb{F}^r$, a positive integer $k$

Goal: find a permutation of $v_1, v_2, \ldots, v_n$ such that for every $i$, 
$$\dim(<v_1 + \ldots + v_i> \cap <v_{i+1}, \ldots, v_n>) \leq k.$$

$\mathbb{F}$ is a finite field

**linear layout** (or pathwidth) of width $\leq k$

$$\dim(<v_1, v_4, v_5, v_2> \cap <v_3, v_6>) \leq k$$
Input: vectors $v_1, v_2, \ldots, v_n \in F^r$, a positive integer $k$

Goal: find a subcubic tree $T$ with a bijection $L: \{\text{leaves}\} \rightarrow \{\text{vectors}\}$ such that for every $e$ in $T$, $\dim(<v_1, v_3, v_6, v_2> \cap <v_4, v_5>) \leq k$
From graphs to matroids

\[ \text{pw } (\text{cycle matroid of } G') = \text{pw } (G) \]

Kashap 2008

When \( G \) is not a tree

\[ \text{bw } (\text{cycle matroid of } G) = \text{bw } (G) \]

Hicks, McMurray, Nolan 2007
Mazoit, Thomassé 2007
XP algorithms

- When $k$ is a fixed constant, polynomial-time algorithm exists for matroids in general (given an independence oracle)

- Pathwidth: Nagamochi 2012

- Branchwidth: Oum, Seymour 2007

What about FPT algorithms?
$\mathcal{G}_k = \{ G : \text{pw}(G) \leq k \}$ is minor-closed

$\mathcal{F} = \{ \text{minimal graphs, } \not\preceq_m \mathcal{G}_k \}$

$G \in \mathcal{G}_k$

$\mathcal{F}$ is finite in time $f(k) \cdot n^2$

Testing if $F \not\preceq_m G$ for all $F \in \mathcal{F}$

Under graph minor relation $\preceq_m$, any antichain is finite

Testing if $F \preceq_m G$ in $g(|F|) \cdot n^2$ time

(Kawarabayashi, Kobayashi, Yusuke, Reed 2012)
\( \text{pw}(M) \leq k \) via minor testing

\[ M_k = \{ M : \text{pw}(M) \leq k \} \text{ is minor-closed} \]

\[ F = \{ \text{minimal matroids}, \preceq_m M_k \} \]

\[ M \in M_k \]

\[ N \preceq_m M, \ \forall N \in F \]

\[ \text{In time } f(k) \cdot \text{poly}(n) \]

\[ \text{test if } N \preceq_m M \]

\[ \forall N \in F \]

Under matroid minor relation \( \preceq_m \), any antichain is finite

\[ \mathcal{F} \text{ is finite} \]

Testing if \( N \preceq_m M \) in \( g(|N|) \cdot \text{poly}(n) \) time

✔

✘
**pw(M) ≤ k via minor testing**

- **Conjecture (RS):** for each finite field $F$, $F$-representable matroids are w.q.o under $\preceq_m$
- **Geelen, Gerards, Whittle (2014):** announced the proof.

- **M is binary $\leftrightarrow U_{2,4} \preceq_m M$$
- **Any algorithm (with independence oracle) requires exponential time (Seymour 1981)**

- **Infinite antichain is not difficult to construct.**

- **Under matroid minor relation $\preceq_m$, any antichain is finite**

- **Testing if $N \preceq_m M$ in time $g(|N|) \cdot \text{poly}(n)$**

- **Conjecture:** for any finite field $F$ and any $F$-representable $N$, testing $N \preceq_m M$ in time $g(|N|) \cdot \text{poly}(n)$
- **Probably GGW taking care of it?**
$\text{pw}(M) \leq k$ when $\mathcal{F}$-represented

$\mathcal{M}_k = \{ M : \text{pw}(M) \leq k \& \text{\mathcal{F}-represented} \}$ is minor-closed

$\mathcal{F} = \{ \text{minimal} \ \mathcal{F}\text{-represented} \ \text{matroids}, \not\preceq_m \mathcal{M}_k \}$

Under matroid minor relation $\not\preceq_m$, any antichain of $\mathcal{F}$-representable matroids is finite.

If $M \in \mathcal{M}_k$

$N \not\preceq_m M, \forall N \in \mathcal{F}$

In time $f(k) \cdot \text{poly}(n)$ test if $N \not\preceq_m M$

$\forall N \in \mathcal{F}$

$\mathcal{F}$ is finite

Testing if $N \not\preceq_m M$ in $g(|N|) \cdot \text{poly}(n)$ time
\[ \text{pw}(M) \leq k \text{ when } F\text{-represented} \]

\[ M_k = \{ M: \text{pw}(M) \leq k \text{ & } F\text{-represented}\} \text{ is minor-closed} \]

\[ F = \{ \text{minimal } F\text{-represented matroids, } \not\preceq_m M_k \} \]

\[ M \in M_k \]
\[ N \not\preceq_m M, \forall N \in F \]

Under matroid minor relation \( \preceq_m \), any antichain of \( F\)-representable matroids bounded bw is finite.

\[ F \text{ is has branchwidth } \leq k+1 \]

\[ \text{In time } f(k) \cdot \text{poly}(n) \text{ test if } N \not\preceq_m M \]
\[ \forall N \in F \]

\[ F \text{ is finite} \]

Testing if \( N \preceq_m M \) in \( g(k) \cdot n^3 \) time when \( \text{bw}(M) \leq k \) (Hliněný 2006)
Issues with the approach

- Too many obstructions - at least \((k!)^2\) (Koutsonas, Thilikos, Yamazaki 2014)
- No algorithm to generate all minor obstructions.
- Even if the complete obstruction list is known, minor testing algorithm (Hliněný 2006) hides gigantic function on \(k\) - relies on MSO checking.
- Only decision - no algorithm to actually produce a layout.
Our result

$O( f(k) \cdot n^3)$-time algorithm to decide a linear layout $v_1, v_2, ..., v_n$ of the input $n$ vectors in $\mathbb{F}^m$, for any finite field $\mathbb{F}$, such that

$$\dim (<v_1, v_2, ..., v_i> \cap <v_{i+1}, ..., v_n>) \leq k$$

for all $i$ and output one if exists.

- Dynamic programming for pathwidth (Bodlaender, Kloks 1996)
- Does not depend on a heavy machinery
- Constructive
- Uses 3-approximation for branch-decomposition and can be made self-contained with $O(n)$ overhead
Algorithm for Vector Arrangement
Dynamic programming on a branch-decomposition

- Want to encode all feasible linear layouts of $M_1$ & $M_2$
- ...in a compact way
- ...in a way s.t. encoding for $M_1 + M_2$ can be constructed from encodings for $M_1$ and $M_2$. 

Diagram:

```
        enc(M₁+M₂)
       /    \\       \\
      /       \\
  enc(M₁)   enc(M₂)

M₁   M₂
```
Encoding a linear layout

When \( <M_1> \cap <M_2> = \emptyset \)

Issues

1. How to shorten the length of the dimension sequence
2. How to handle the boundary space; \( <M_1> \cap <M_2> \neq \emptyset \)
Compressing a dimension sequence

Typical sequence: idea from Bodlaender-Kloks (1996)

#typical sequences consisting of \{0,1,2,\ldots,k\} \leq (8/3)2^{2k}. 

k

Diagram of a typical sequence with fluctuations and markers at intervals.
Handling a boundary space

When \( B := <M_1> \cap <M_2> \neq \emptyset \)

\[
\dim(L \cap R) - \dim(L \cap R \cap B)
\]

\[
\dim(L + L') \cap (R + R') - \dim(L + L') \cap (R + R') \cap B
\]
Encoding of a linear layout

For each “gap”, there is a triple \((L, R, \lambda)\)

\[ L: \text{“Left subspace”} \text{ shown on } B \]
\[ L_3 = <v_1, v_4, v_5> \cap B \]

\[ R: \text{“Right subspace”} \text{ shown on } B \]
\[ R_3 = <v_2 + v_3 + v_6> \cap B \]

\[ \lambda: \text{Extra connectivity not shown in } B \]
\[ \lambda_3 = \dim<v_1, v_4, v_5> \cap <v_2 + v_3 + v_6> - \dim<v_1, v_4, v_5> \cap <v_2 + v_3 + v_6> \cap B \]
Applications
An alternative definition of pathwidth

• A path-decomposition of $\mathbb{F}^r$

$\pi = (S_1,S_2,...,S_m)$ of subspaces of $\mathbb{F}^r$

with an injective function $\mu:\{1,2,...,n\} \rightarrow \{1,2,...,m\}$ s.t.

$V_i \subseteq S_{\mu(i)}$.

• Width of $\pi := \max (S_1 + ... + S_i) \cap (S_{i+1} + ... + S_m)$

$\exists$ linear layout of width $\leq k$

$\Updownarrow$

$\exists$ path-dec. of width $\leq k$. 
Subspaces Arrangement

Input: subspaces $V_1, V_2, \ldots, V_n$ of $\mathbb{F}$, a positive integer $k$

Goal: find a permutation of $V_1, V_2, \ldots, V_n$ such that for every $i$, 
$\dim(<V_1+\ldots+V_i> \cap <V_2,\ldots,V_n>) \leq k$.

$\mathbb{F}$ is a finite field

linear layout of width $\leq k$

$\dim(<V_1,V_4,V_5,V_2> \cap <V_3,V_6>) \leq k$
Application to Linear rank-width

- Find a linear ordering of vertices of G so that the rank of adjacency matrix between \{v_1, \ldots v_i\} and \{v_{i+1}, \ldots v_n\} is at most k.

```
\begin{array}{c|c}
 v_1 & 1 \\
v_4 & 0 \\
v_5 & 1 \\
v_2 & 1 \\
v_3 & 1 \\
v_6 & 1 \\
\end{array}
```

Path-width of \{V_1, V_2, \ldots, V_n\} = 2*(linear rank-width of G)
Application to Linear rank-width

For any fixed $k$, $O(n^3)$-time algorithm to find a linear rank decomposition of width $\leq k$ or confirms that linear rank-width $> k$. 
Application to linear clique-width

- Linear clique-width, “linearized version of clique-width”
- Linear rank-width $\leq k \Rightarrow$ linear clique-width $\leq 2^k + 1$.

For any fixed $k$, $O(n^3)$-time algorithm to find a linear clique-width expression of width $\leq 2^k + 1$ or confirms that linear clique-width $> k$. 
Further questions

• FPT algorithms for pathwidth / branchwidth on general matroids? (given independence oracle)

• Can $O(n^3)$ factor in the runtime improved? i.e. $O(n^w)$

THANK YOU!