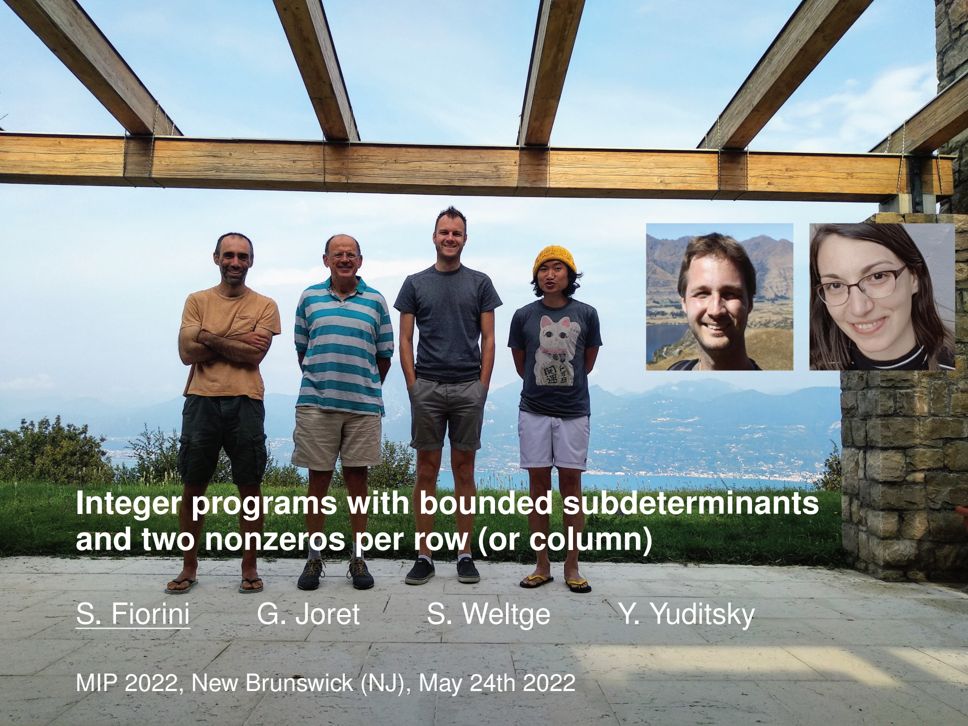


Integer programs with bounded subdeterminants and two nonzeros per row (or column)

S. Fiorini G. Joret S. Weltge Y. Yuditsky

MIP 2022, New Brunswick (NJ), May 24th 2022

What is this talk about

IP

Graph minors

Outline

1. Main result, motivation and previous work
2. Reduction to MWSS
3. The structure theorem
4. Particular case of bounded genus graphs
5. Back to the general case

Outline

1. Main result, motivation and previous work
2. Reduction to MWSS
3. The structure theorem
4. Particular case of bounded genus graphs
5. Back to the general case

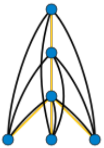
$$\boxed{\text{(IP)}\ \max\{w^\intercal x : Ax = b,\ x \geqslant 0,\ x \in \mathbb{Z}^n\}}$$

Meta-question

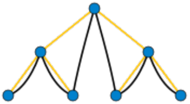
What parameters make (IP) hard / easy?

- number of variables n (Lenstra ’**83**)
- branch-width of $M(A)$ (Cunningham and Geelen ’**07**)
- tree-width of $G(A)$ (Bienstock and Muñoz ’**18**)
- tree-depth of $G(A)$ or $G(A^\intercal)$ (Eiben et al ’**19**, Eisenbrand et al ’**19**, Cslovjecsek et al ’**21**)
- …
- maximum subdeterminant $\Delta(A)$

$\mathrm{td}(K_4) \leqslant 4$



$\mathrm{td}(K_{3,3}) \leqslant 4$



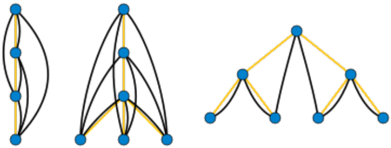
$\mathrm{td}(P_7) \leqslant 3$

$$\boxed{\text{(IP)}\ \max\{w^\intercal x : Ax = b,\ x \geqslant 0,\ x \in \mathbb{Z}^n\}}$$

Meta-question

What parameters make (IP) hard / easy?

- number of variables n (Lenstra '**83**)
- branch-width of $M(A)$ (Cunningham and Geelen '**07**)
- tree-width of $G(A)$ (Bienstock and Muñoz '**18**)
- tree-depth of $G(A)$ or $G(A^\intercal)$ (Eiben et al '**19**, Eisenbrand et al '**19**, Cslovjecsek et al '**21**)
- ...
- maximum subdeterminant $\Delta(A) \longleftarrow$ **this work**



$\mathrm{td}(K_4) \leqslant 4$ $\mathrm{td}(K_{3,3}) \leqslant 4$ $\mathrm{td}(P_7) \leqslant 3$

Our parameters

Definition

For $\Delta \in \mathbb{Z}_{\geqslant 0}$, a matrix A is called *totally Δ-modular* if

$$\det(A') \in \{-\Delta, -\Delta + 1, \dots, 0, \dots, \Delta - 1, \Delta\}$$

for all square submatrices A' of A

Given A, let $\quad \Delta(A) := \min\{\Delta : A \text{ is totally } \Delta\text{-modular}\}$

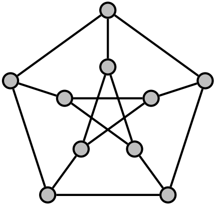
Definition

The *odd cycle packing number* $\mathrm{ocp}(G)$ is the maximum number of vertex-disjoint odd cycles in G

Examples:

- A is totally unimodular (TU) $\iff \Delta(A) \leqslant 1$

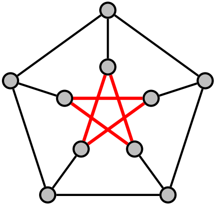
- A is the incidence matrix of graph $G \implies \Delta(A) = 2^{\mathrm{ocp}(G)}$



$$\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}$$

Examples:

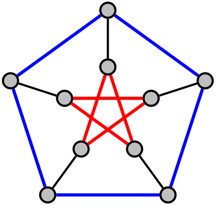
- A is totally unimodular (TU) $\iff \Delta(A) \leqslant 1$
- A is the incidence matrix of graph $G \implies \Delta(A) = 2^{\mathrm{ocp}(G)}$



$$
\begin{pmatrix}
{\color{red}1} & {\color{red}1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & {\color{red}1} & {\color{red}1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & {\color{red}1} & {\color{red}1} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & {\color{red}1} & {\color{red}1} & 0 & 0 & 0 & 0 & 0 \\
{\color{red}1} & 0 & 0 & 0 & {\color{red}1} & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

Examples:

- A is totally unimodular (TU) $\iff \Delta(A) \leqslant 1$
- A is the incidence matrix of graph $G \implies \Delta(A) = 2^{\mathrm{ocp}(G)}$



$$\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}$$

Our main result(s)

Theorem (FJWY '21)

For every integer $\Delta \geq 1$ there exists a strongly polynomial-time algorithm for solving the integer program (IP)

$$\max\{w^\intercal x : Ax \leqslant b,\ x \in \mathbb{Z}^n\}$$

where $w \in \mathbb{Z}^n$, $b \in \mathbb{Z}^m$, and constraint matrix $A \in \mathbb{Z}^{m \times n}$

- *is totally Δ-modular, and*
- *contains at most two nonzero entries in each row (or in each column)*

Theorem (FJWY '21)

For every integer $k \geq 0$ there exists a strongly polynomial-time algorithm for the weighted stable set problem in graphs with $\mathrm{ocp}(G) \leqslant k$

Previous work

1. PTAS for MWSS for $\operatorname{ocp}(G) = O(1)$ (Demaine, Hajiaghayi, Kawarabayashi **'10**, Tazari **'12**)
2. PTAS for MWSS even for $\operatorname{ocp}(G) = O(\sqrt{n/\log\log n})$ (Bock, Faenza, Moldenhauer, Ruiz-Vargas **'14**)
3. (IP) can be solved in strongly polynomial-time if $\Delta = 1$
4. (IP) can be solved in strongly polynomial-time if $\Delta = 2$ (Artmann, Weismantel, Zenklusen **'17**)
5. There is a polynomial-time algorithm that solves (IP) w.h.p. over the choices of b, when A, w are fixed and Δ is constant (Paat, Schlöter, Weismantel **'19**)
6. The diameter of $P := \{x : Ax \leqslant b\}$ is $O(\Delta^2 n^4 \lg n\Delta)$ (Bonifas, Di Summa, Eisenbrand, Hähnle, Niemeier **'14**)
7. $\max\{w^\intercal x : Ax = b,\ x \geqslant 0\}$ can be solved in time $\operatorname{poly}(m, n, \lg \Delta)$ (Tardos **'86**)

Outline

Proximity result of Cook et al.

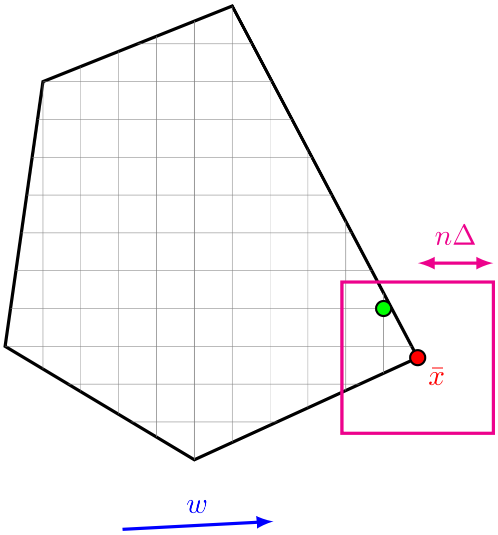
Theorem (Cook, Gerards, Schrijver, Tardos '86)

Let A be a totally Δ-modular $m \times n$ matrix and let b and w be integer vectors such that

- $Ax \leqslant b$ *has an integral solution, and*
- $\max\{w^{\intercal}x : Ax \leqslant b\}$ *exists.*

Then for each optimal solution $\bar{x}$ to $\max\{w^{\intercal}x : Ax \leqslant b\}$, there exists an optimal solution z^* to $\max\{w^{\intercal}x : Ax \leqslant b,\ x \in \mathbb{Z}^n\}$ with

$$||\bar{x} - z^*||_\infty \leqslant n\Delta$$


$n\Delta$
$\bar{x}$
w

1st reduction: reducing to $A \in \{-1, 0, 1\}^{m \times n}$

After permuting rows and columns:

$$A = \left[\begin{array}{c|c}
\overbrace{\begin{matrix} * & * & & & & & & \\ & & * & * & & & & \\ & & & & * & * & & \\ & & & & & & * & * \end{matrix}}^{\leqslant 2\lg\Delta} & \\ \hline
* & \boxed{\begin{matrix} * & & & \\ \pm1 & \pm1 & & \\ \pm1 & & \pm1 & \\ & & \pm1 & \pm1 \end{matrix}}
\end{array}\right]$$

1st reduction:

- Solve LP relaxation $\max\{w^\intercal x : Ax \leqslant b\} \to \bar{x}$
- Guess the first $O(\lg \Delta)$ variables

2nd reduction: reducing to $A \in \{0,1\}^{m \times n}$, $b = 1$

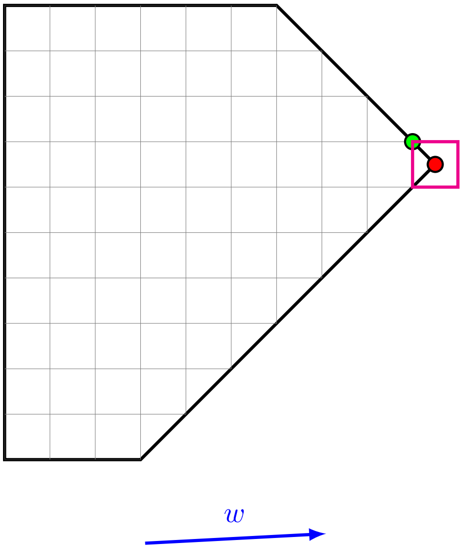
Theorem (FJWY '21)

Let $A \in \{-1,0,1\}^{m \times n}$, $b \in \mathbb{Z}^m$, $w \in \mathbb{Z}^n$. Assume that

- *every row of A has $\leqslant 2$ nonzeros,*
- *$P := \{x : Ax \leqslant b\}$ is bounded and $P \cap \mathbb{Z}^n \neq \varnothing$.*

For every extremal optimal solution $\bar{x}$ to $\max\{w^\intercal x : Ax \leqslant b\}$, there exists an opt. solution z^* to $\max\{w^\intercal x : Ax \leqslant b,\ x \in \mathbb{Z}^n\}$ with

$$||\bar{x} - z^*||_\infty \leqslant \frac{1}{2}$$


w

Final problem

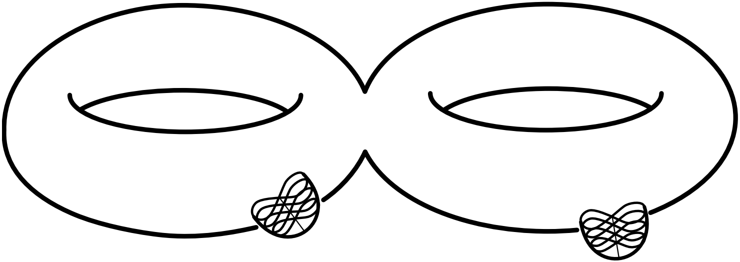
After translating and reformulating, we get

$$\begin{array}{rl} \max & w^\intercal x \\ \text{s.t.} & Ax \leqslant \mathbf{1} \\ & x \in \mathbb{Z}^n \end{array}$$

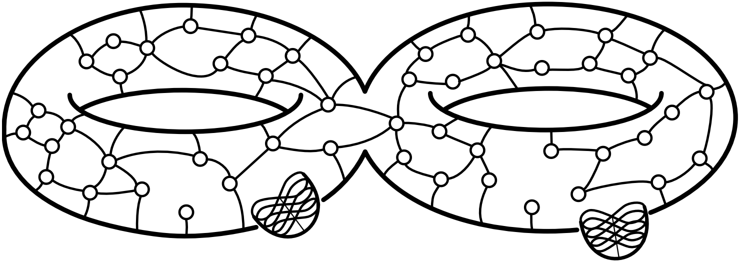
where:

- A is the edge-vertex incidence matrix of some graph G
- $\mathrm{ocp}(G) \leqslant \lg \Delta$
- $w \in \mathrm{cone}(A^\intercal)$

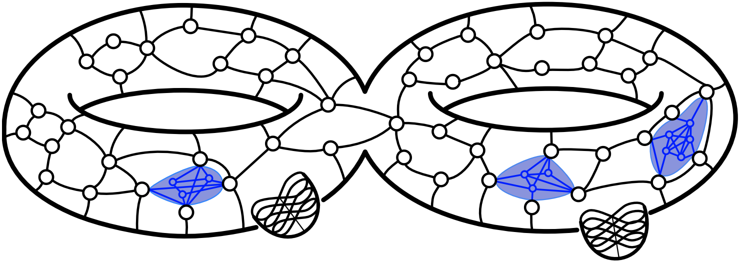
Outline



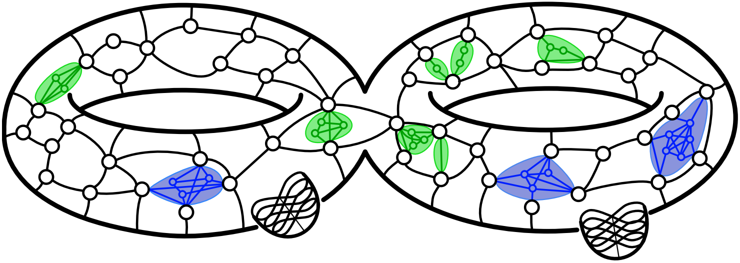
A bounded genus **surface** $\mathbb{S}$



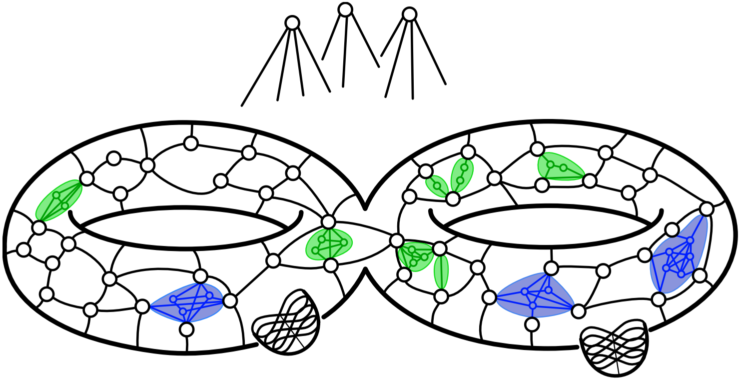
Part of G is embedded in $\mathbb{S}$



Plus a bounded number of large vortices



Plus small vortices

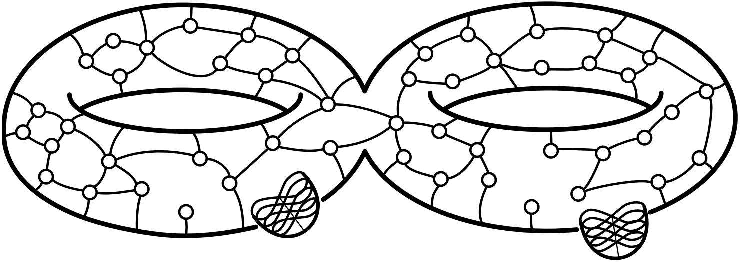


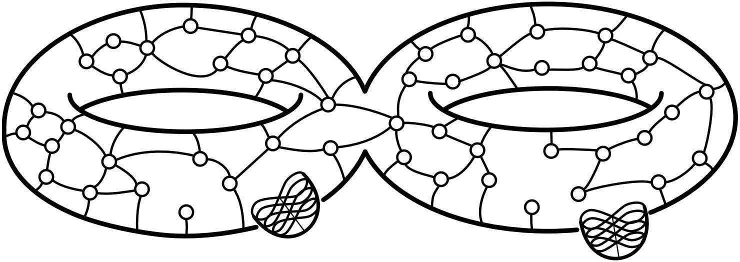
Plus a bounded number of **apices**

Proof uses several *graph minor papers*

1. Reed **’99** and Kawarabayashi and Reed **’10**
2. Geelen, Gerards, Reed, Seymour, Vetta **’09**
3. Kawarabayashi, Thomas, Wollan **’20**
4. Diestel, Kawarabayashi, Müller, Wollan **’12**

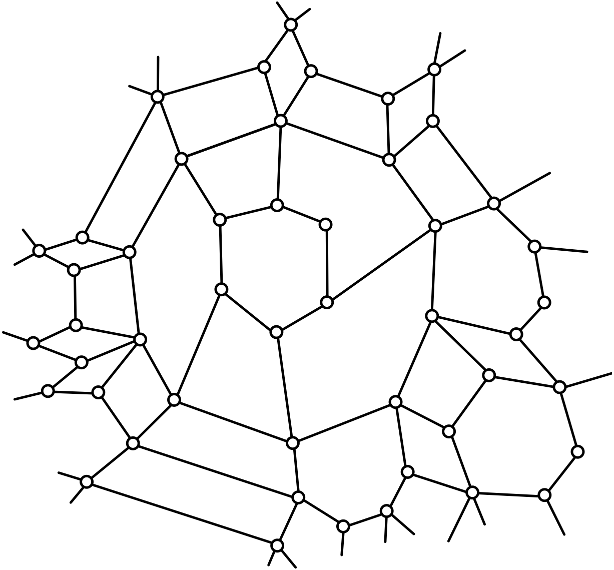
Outline

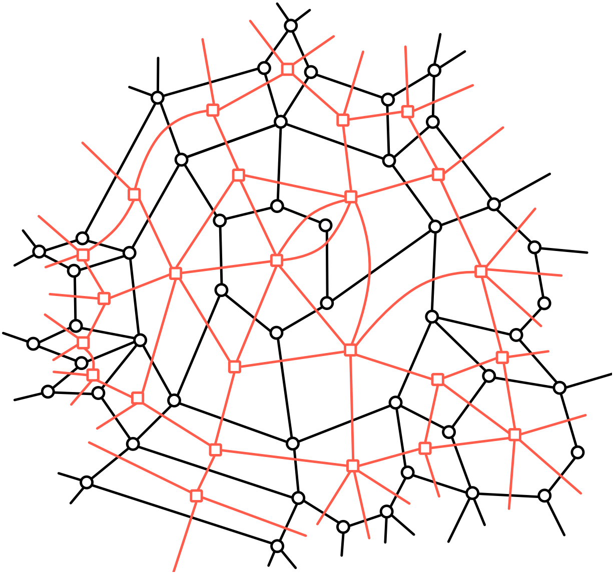


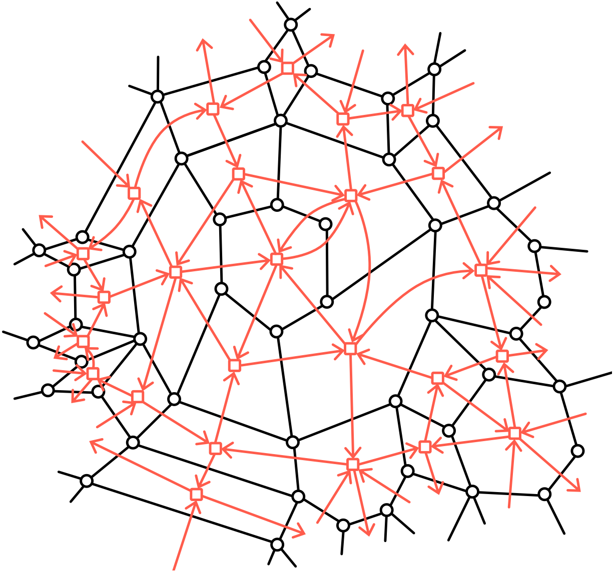


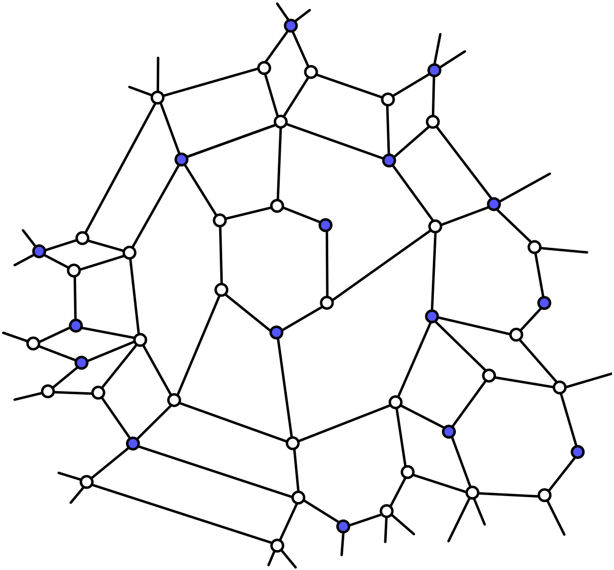
Can assume, using Conforti, F, Huynh, Joret, Weltge **’20**:

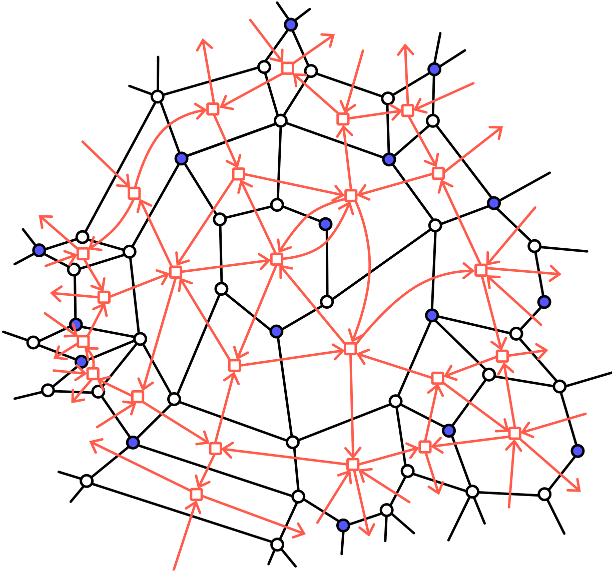
- every odd cycle defines a Möbius band in $\mathbb{S}$ ($\equiv$ is 1**-sided**)

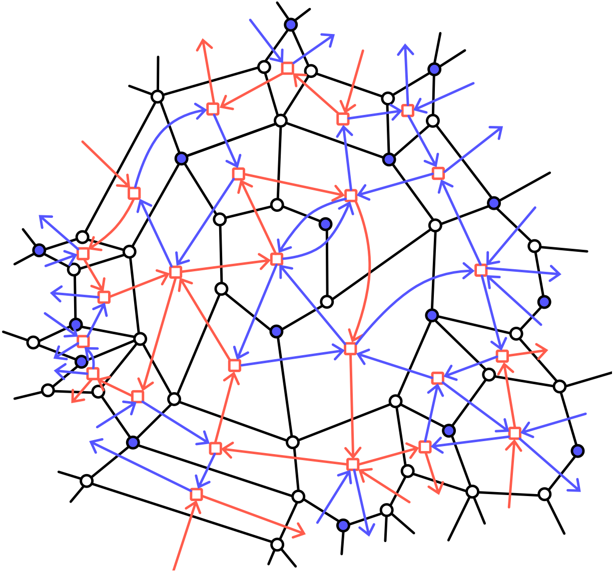


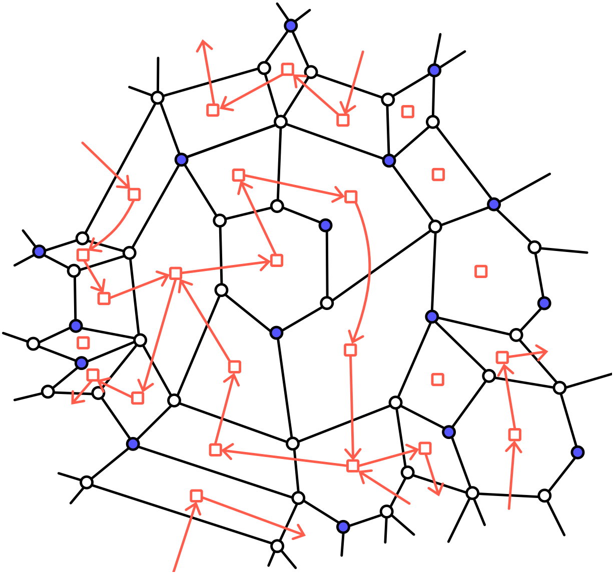












Key insight

Instead of computing a maximum-weight stable set, compute a minimum-cost circulation that is:

- nonnegative and integer
- **homologous** to the all-one circulation

REM: homologous to all-one $\equiv 1$ parity constraint + $g-1$ equations

Key insight

Instead of computing a maximum-weight stable set, compute a minimum-cost circulation that is:

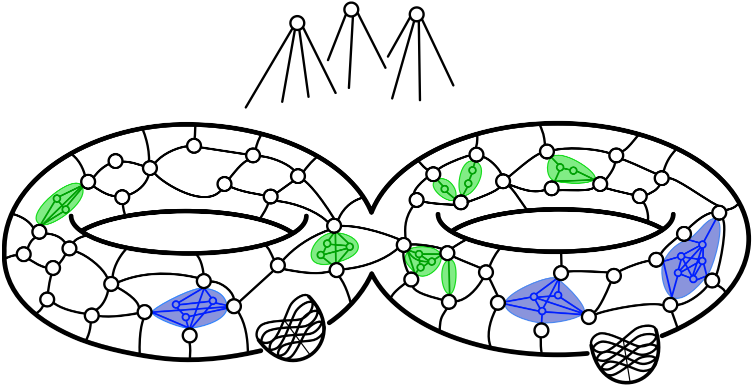
- nonnegative and integer
- **homologous** to the all-one circulation

REM: homologous to all-one $\equiv 1$ parity constraint + $g - 1$ equations

Doable with dynamic programming! ("homologous flows")

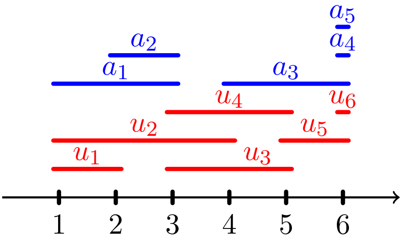
- Conforti, F, Huynh, Joret, Weltge **'20**
- Morell, Seidel, Weltge **'21**

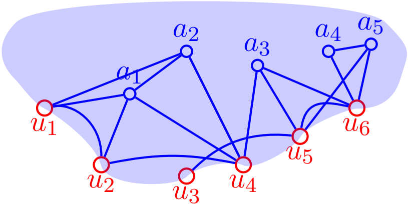
Outline



Large vortices

- Bounded number of large vortices
- Each has a **linear decomposition** of bounded **adhesion**



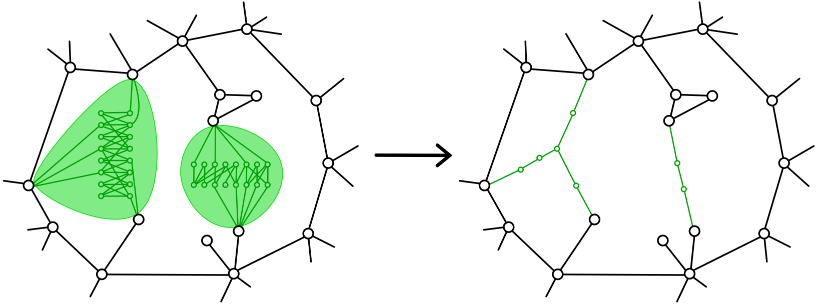


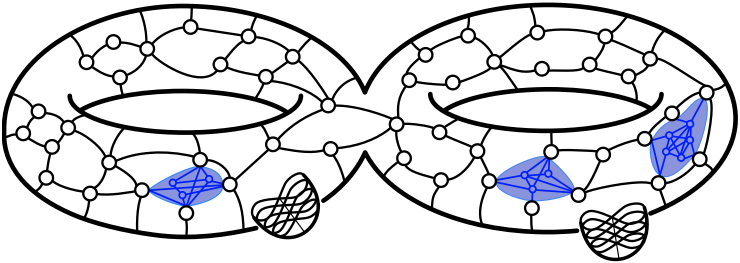
Definition

The *adhesion* of the linear decomposition $(X_1, \ldots, X_n)$ is $\max\{|X_i \cap X_{i+1}| : i < n\}$

Small vortices

From Conforti, F, Huynh, Weltge **'20**:

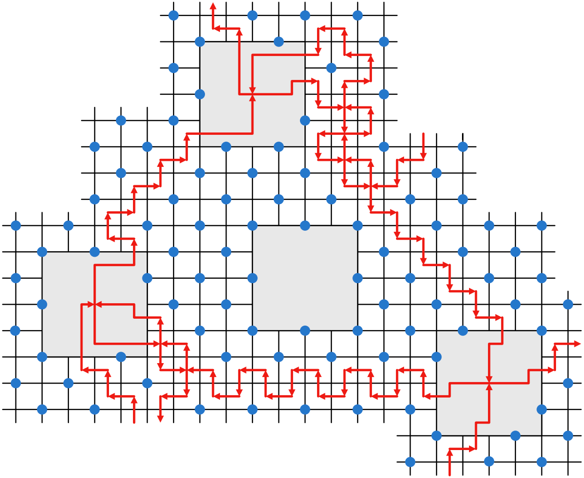




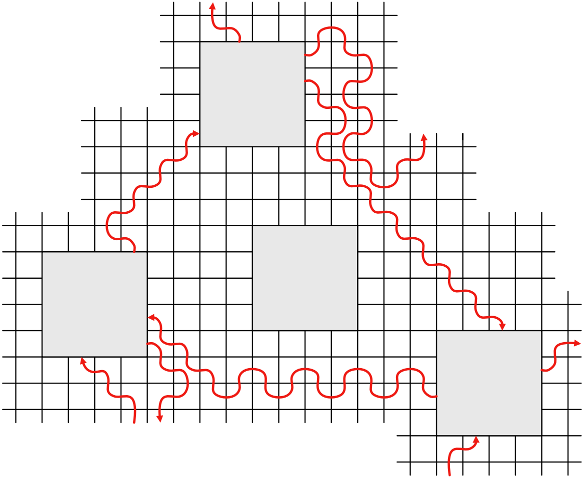
Can assume:

- large vortices are “far apart”, and bipartite

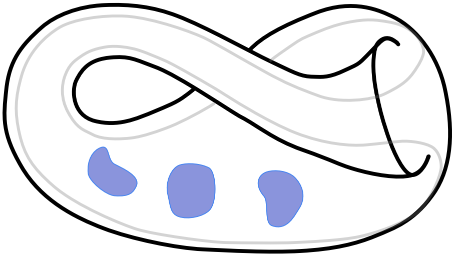
The sketch = “skeleton” of the solution

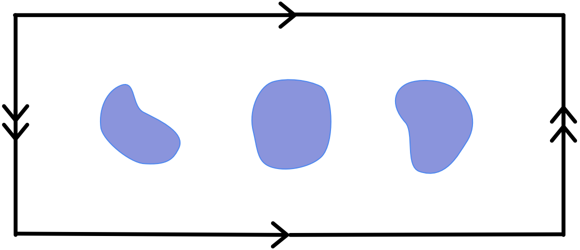


The sketch = “skeleton” of the solution

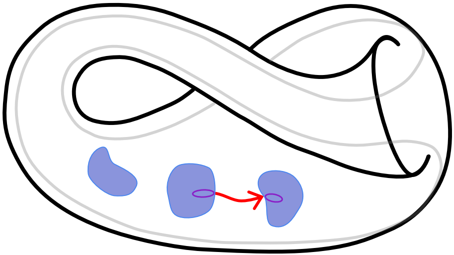


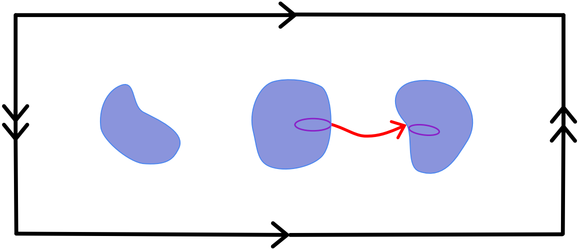
Constructing the sketch curve by curve



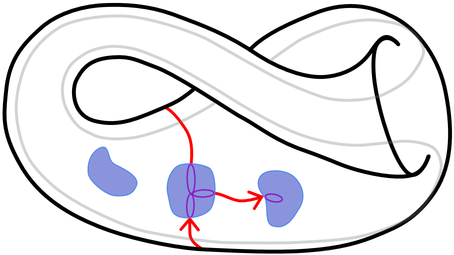


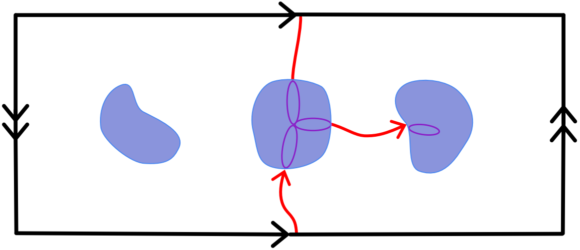
Constructing the sketch curve by curve



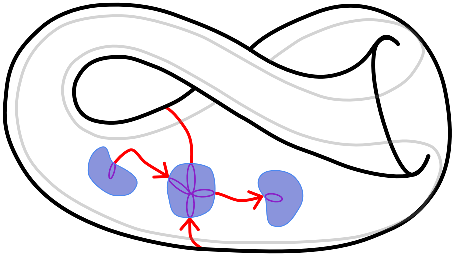


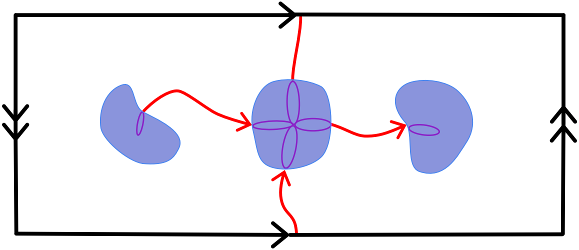
Constructing the sketch curve by curve



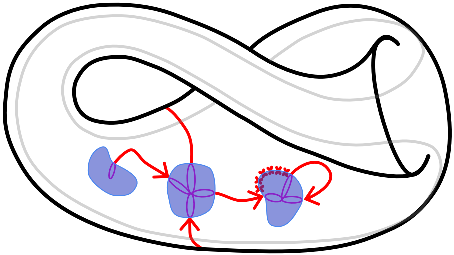


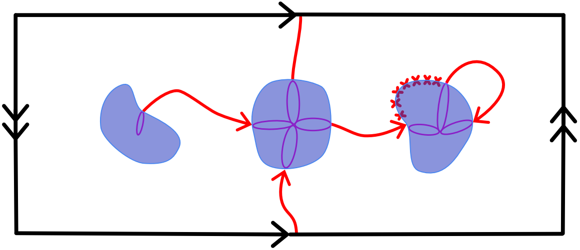
Constructing the sketch curve by curve



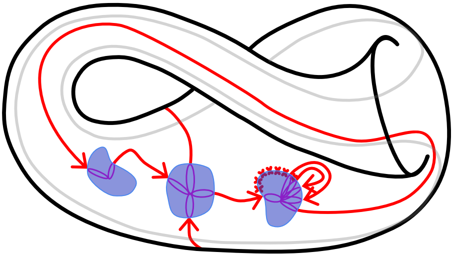


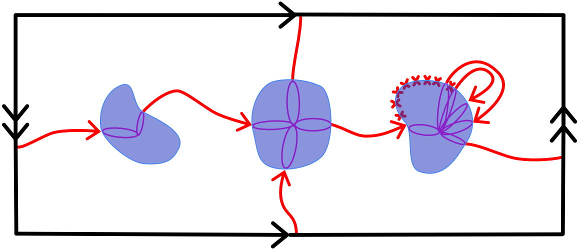
Constructing the sketch curve by curve





Constructing the sketch curve by curve





Main algorithm is a dynamic program (DP):

1. Cells correspond to possible faces of the (partial) sketch
2. Use precedence rule for split operations to bound the number of cells by a polynomial
3. Every curve has two corresponding separators, inside which the solution is guessed
4. The DP remembers “just enough” extra information to guarantee that it constructs solutions that are *feasible*

Subroutines:

- Homologous flows (Morell, Seidel and Weltge **’21**)
- Bipartite stable set instances “between” separators

Open questions

1. Are IPs with bounded Δ polytime solvable?
2. How good is the LP bound when Δ is bounded?
3. MWSS on bounded-OCP graphs and hierarchies (Sherali-Adams, Lasserre, ...)
4. More efficient algorithms? FPT algorithms?
   - MWSS on graphs with bounded OCP
   - MWSS on graphs with bounded OCP and bounded genus

Any questions???