Parameterized Orientable Deletion

I. Katsikarelis[1],
T. Hanaka[2], M. Lampis[1], Y. Otachi[3], F. Sikora[1]

[1]LAMSADE, Université Paris-Dauphine, France
[2]DISE, Chuo University, Tokyo, Japan
[3]Kumamoto University, Japan

20 June 2018

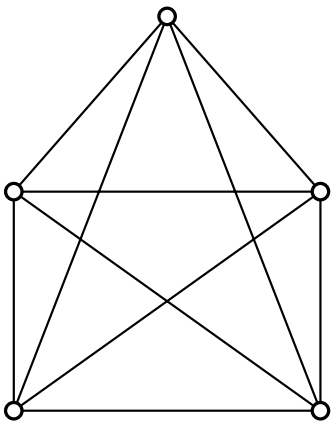
Introduction

d-Orientable Deletion

- Given undirected graph $G = (V, E)$, can we delete at most k vertices so that all remaining edges can be oriented in such a way that no vertex has in-degree $> d$?
- Deciding if an unweighted graph is d-orientable is in P
- Closely related to d-degeneracy (for *acyclic* orientations)
- Vertex Cover corresponds to $d = 0$
- PseudoForest Deletion corresponds to $d = 1$

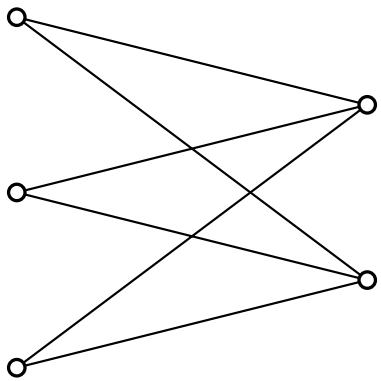
Examples (I)

$d = 2$



K_{2d+1}

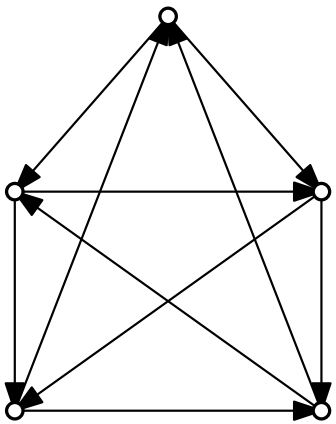
$d = 1$



$K_{2d+1,2d}$

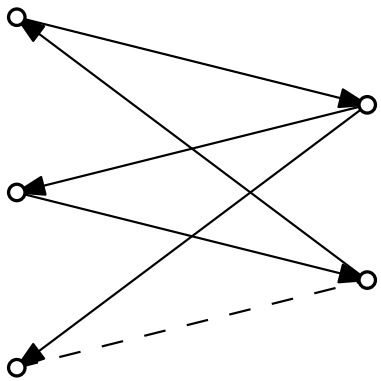
Examples (II)

$d = 2$



K_{2d+1}

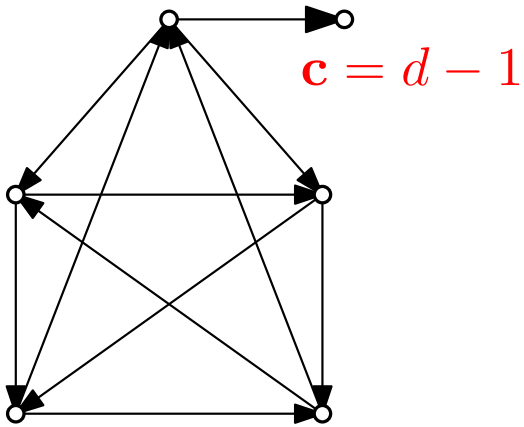
$d = 1$



$K_{2d+1,2d}$

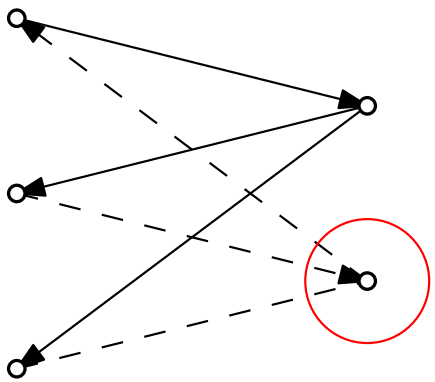
Examples (III)

$d = 2$



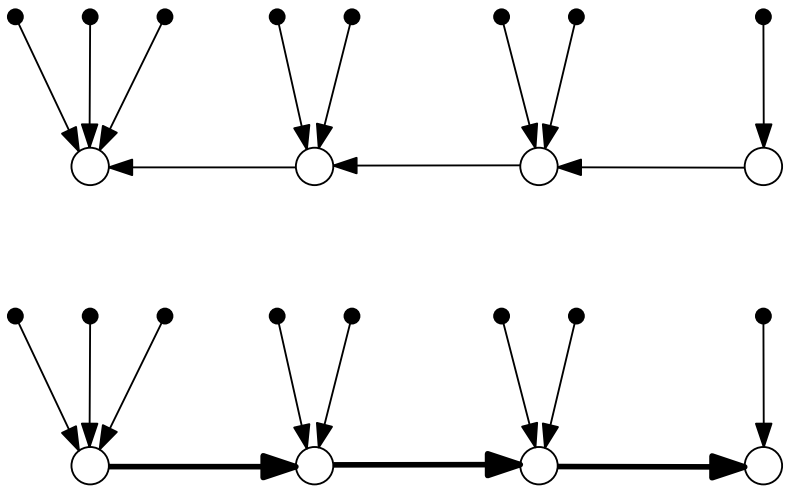
K_{2d+1}

$d = 1$

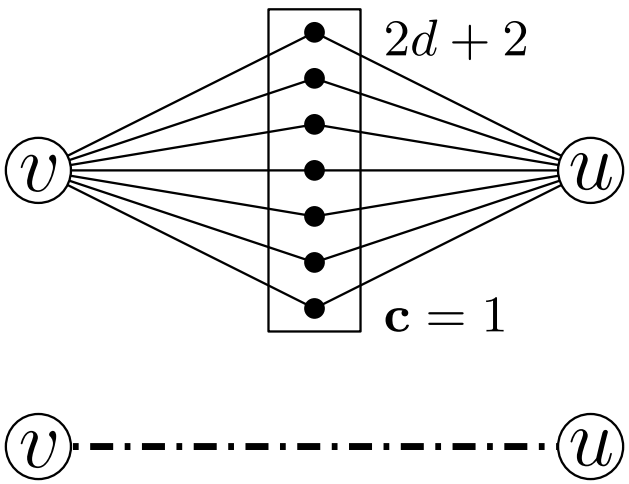


$K_{2d+1,2d}$

Poly-time algorithm



Extra: OR gadget

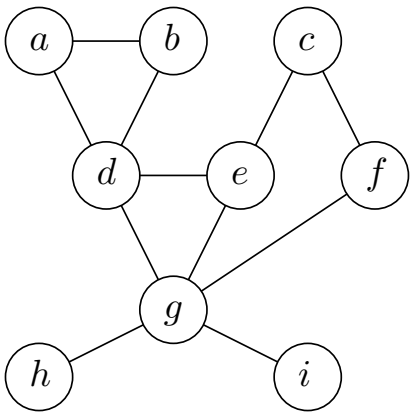


Exponential Time Hypothesis

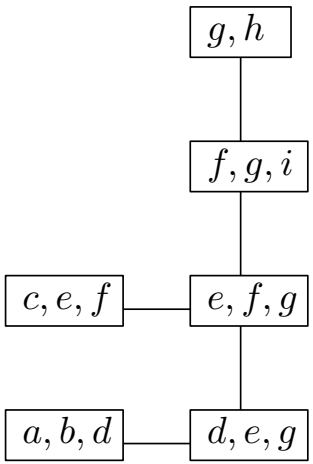
- ETH: there is no $2^{o(n)}$-time algorithm for 3-SAT
- Strong ETH: there is no $O^*((2-\epsilon)^n)$-time algorithm for SAT

Treewidth

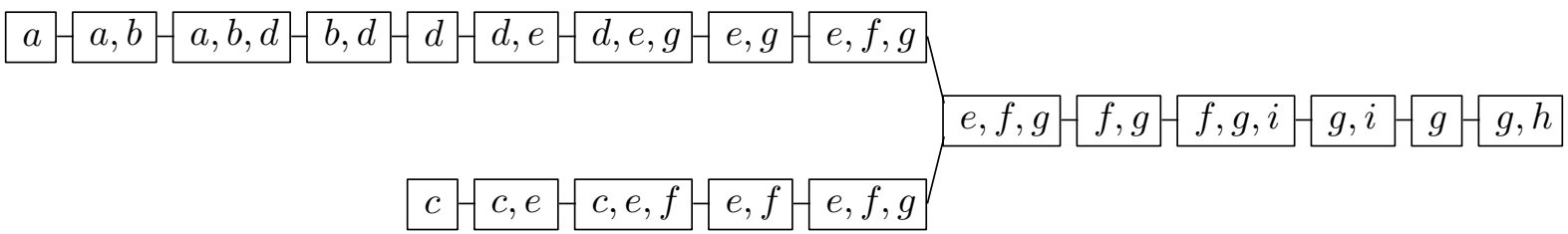
Original graph



Tree decomposition



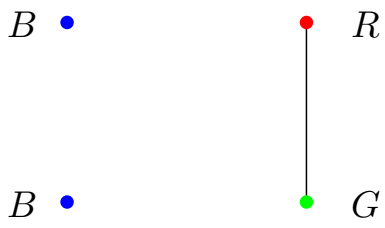
Nice tree decomposition



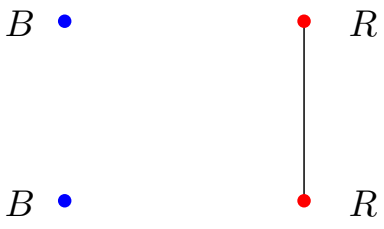
Clique-width

introduce $i(L)$

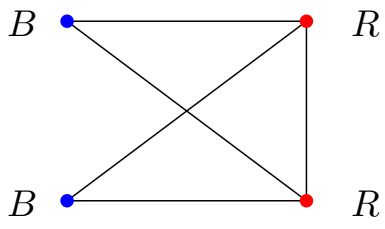
union $G_1 \otimes G_2$



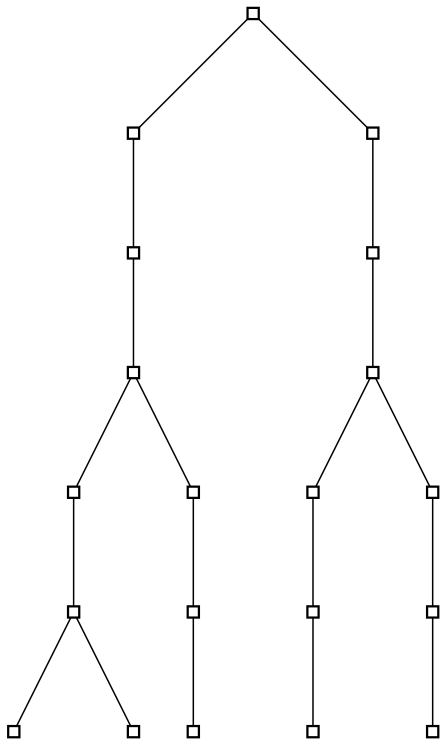
join $\eta(R, G)$



relabel $\rho(G, R)$



join $\eta(B, R)$



expression tree

Overview of Results

- Inapproximability and W-hardness (k)
- Chordal graphs: W-hardness (d, k), FPT algorithm ($d + k$)
- SETH lower bound (treewidth tw)
- FPT algorithm (clique-width $\mathrm{cw} + d$)
- W-hardness (clique-width cw)

SETH lower bound (treewidth)

$$O^*((d+2-\epsilon)^{\mathbf{tw}})$$

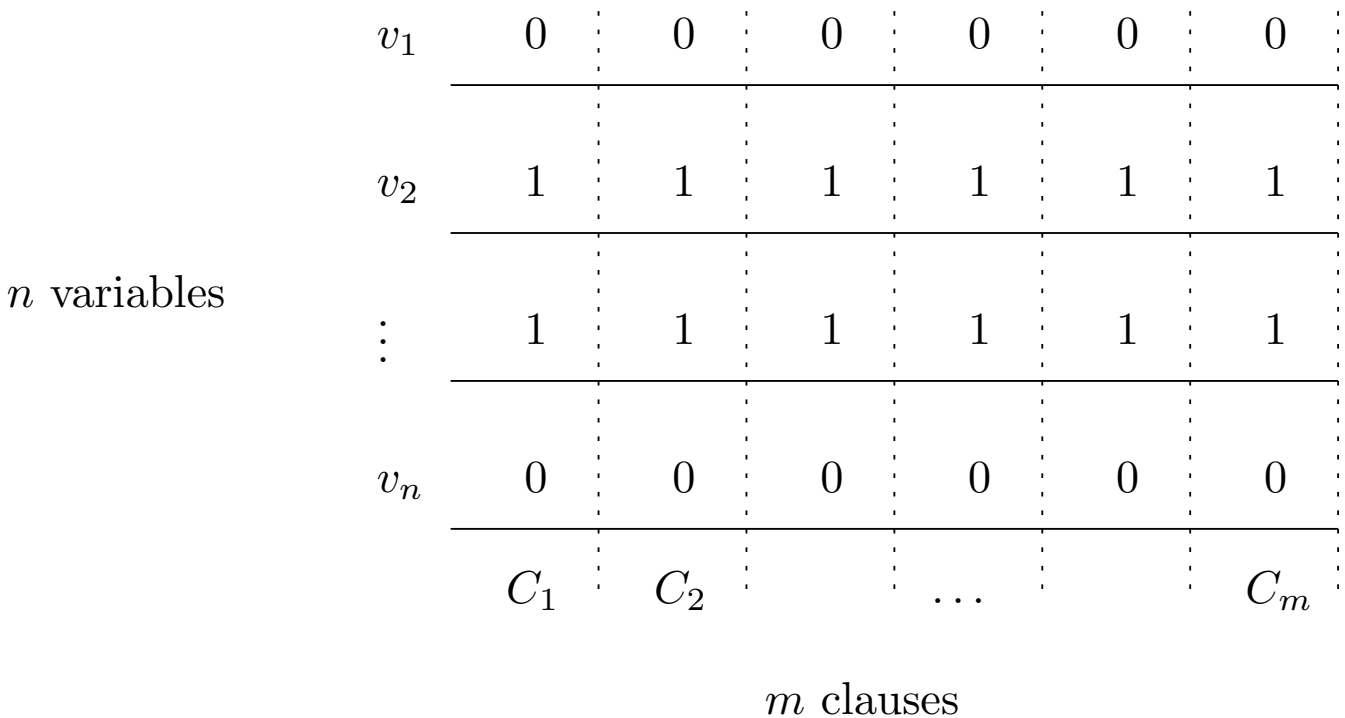
Intuition

- Our work follows the approach for SETH lower bounds for tw by *Lokshtanov et al. (2013)*
- Start with a formula ϕ of SAT on n variables and m clauses and build a graph G of bounded treewidth
- Intuitively, these reductions need to "embed" the 2^n possible assignments into the $(d+2)^{\mathrm{tw}}$ entries of some dynamic program computing the problem in G
- The hard part is to compress n boolean variables into (roughly) $\frac{n}{\log(d+2)}$ units of treewidth, by considering the natural $d+2$ options available for d-orientability of each vertex

Global Construction (I)

2^n

SAT: ϕ



Global Construction (II)

$2^n \to (d+2)^{\mathrm{tw}}$

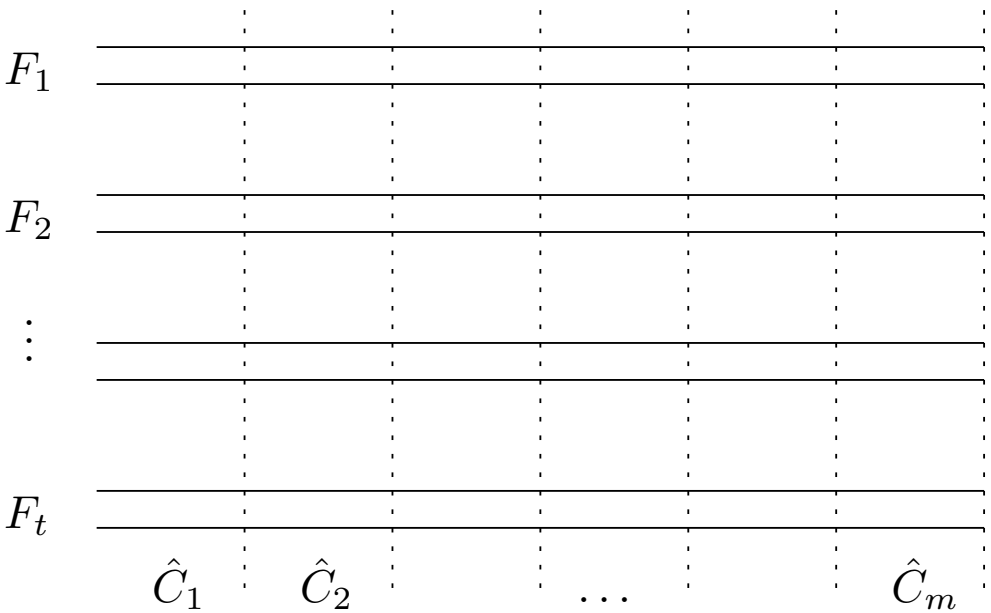
SAT: $\phi \to d$-OrientableDeletion: G, tw

n variables

$\downarrow$

$\mathrm{tw} \approx \frac{n}{\log(d+2)}$

groups



m clauses

Global Construction (III)

$2^n \to (d+2)^{\mathrm{tw}}$

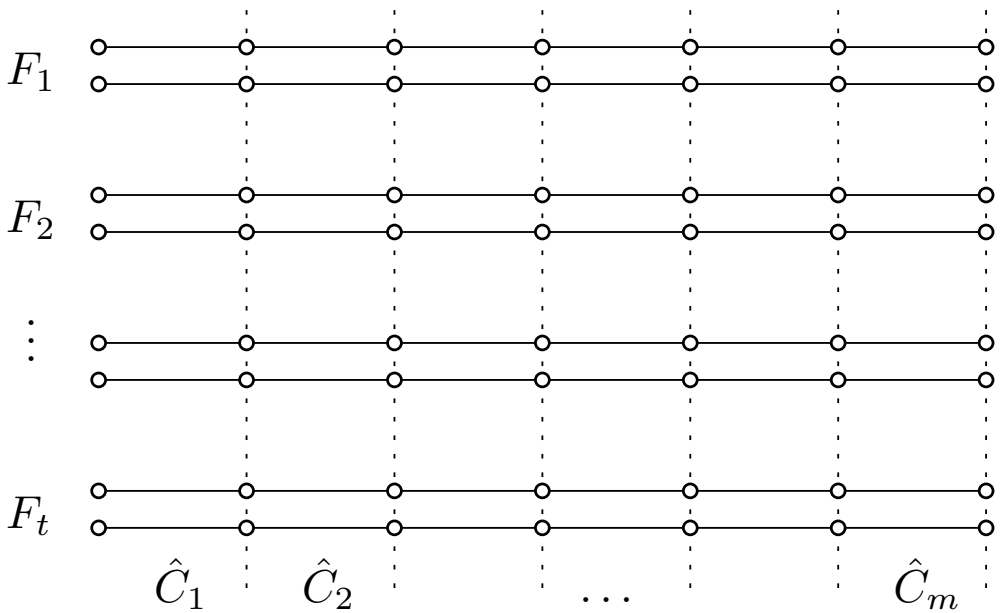
SAT: $\phi \to d$-OrientableDeletion: G, tw

n variables

$\downarrow$

$\mathrm{tw} \approx \frac{n}{\log(d+2)}$

groups



m clauses

Global Construction (IV)

$2^n \to (d+2)^{\mathrm{tw}}$

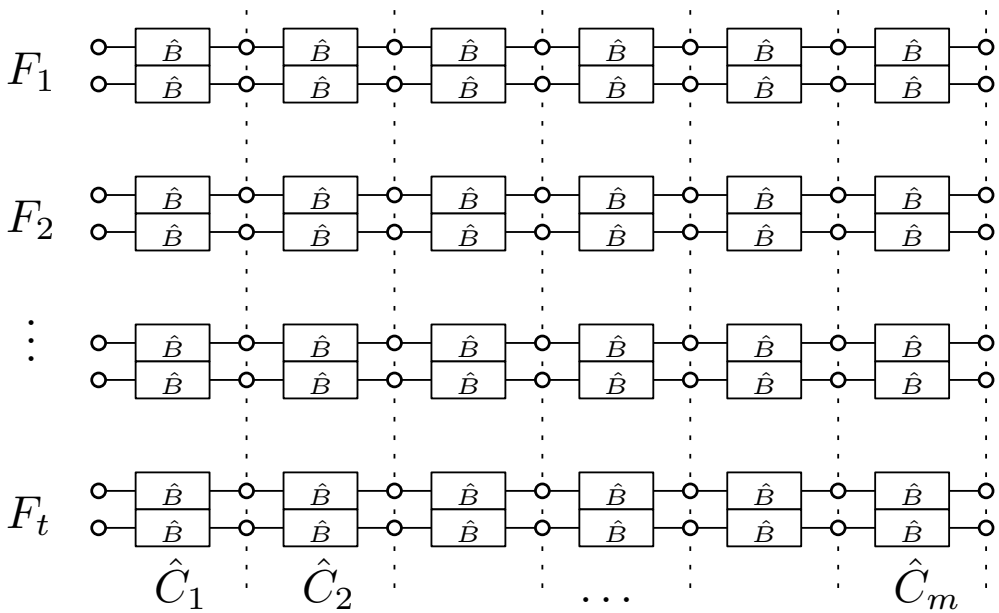
SAT: $\phi \to d$-OrientableDeletion: G, tw

n variables

$\downarrow$

$\mathrm{tw} \approx \frac{n}{\log(d+2)}$

groups



m clauses

Global Construction (V)

$2^n \to (d+2)^{\mathrm{tw}}$

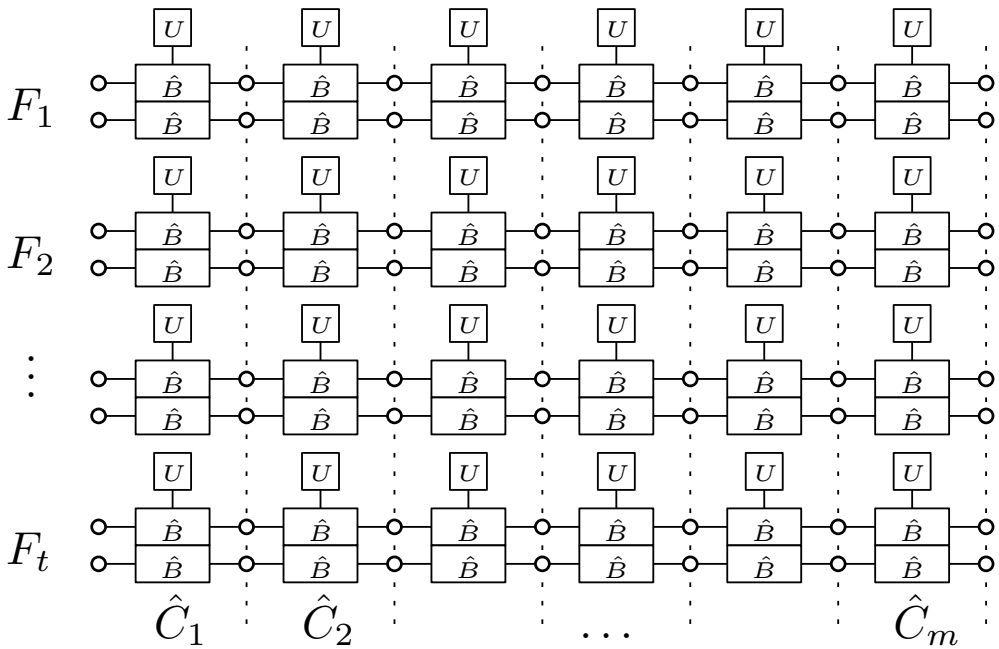
SAT: $\phi \to d$-OrientableDeletion: G, tw

n variables

$\downarrow$

$\mathrm{tw} \approx \frac{n}{\log(d+2)}$

groups



m clauses

Global Construction (VI)

$2^n \to (d+2)^{\mathrm{tw}}$

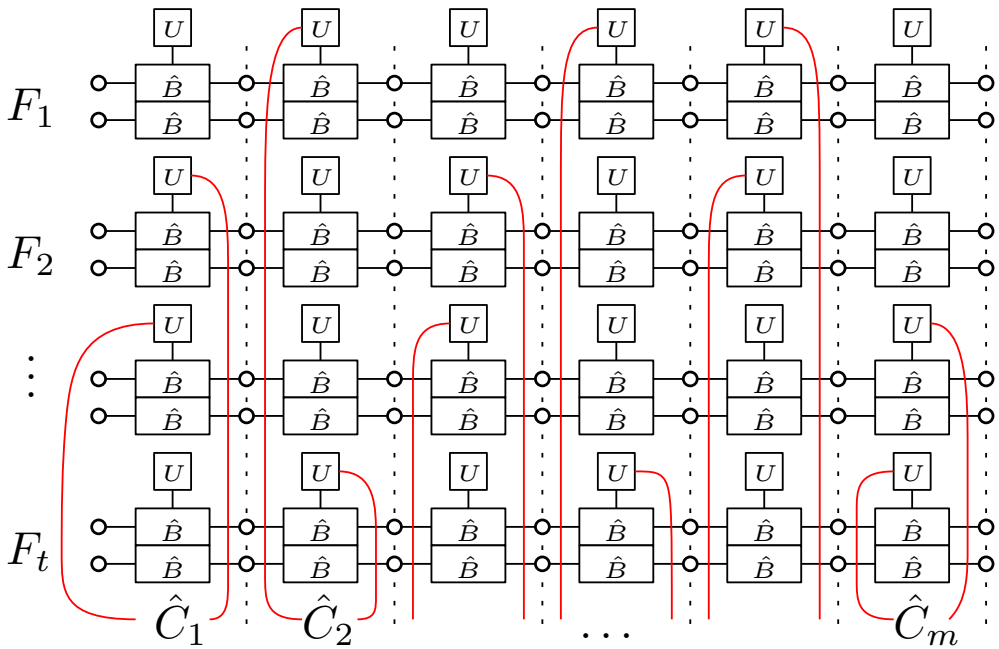
SAT: $\phi \to d$-OrientableDeletion: G, tw

n variables

$\downarrow$

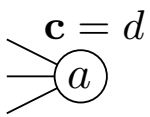
$\mathrm{tw} \approx \frac{n}{\log(d+2)}$

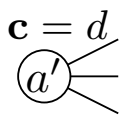
groups



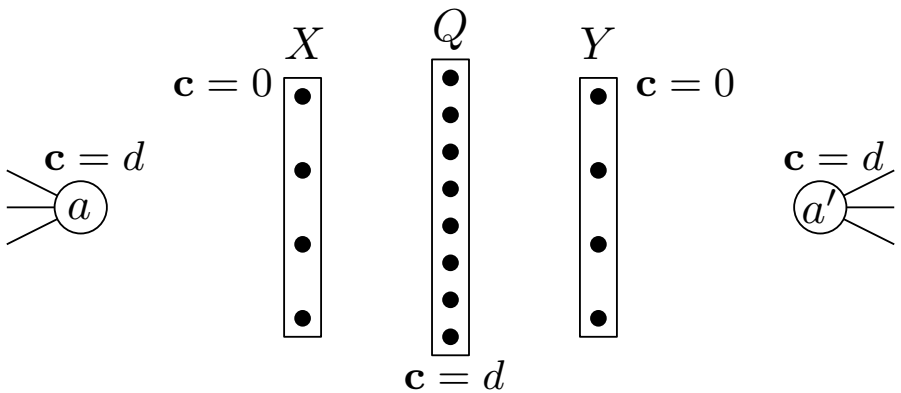
m clauses

Block gadget $\hat{B}$ (I)

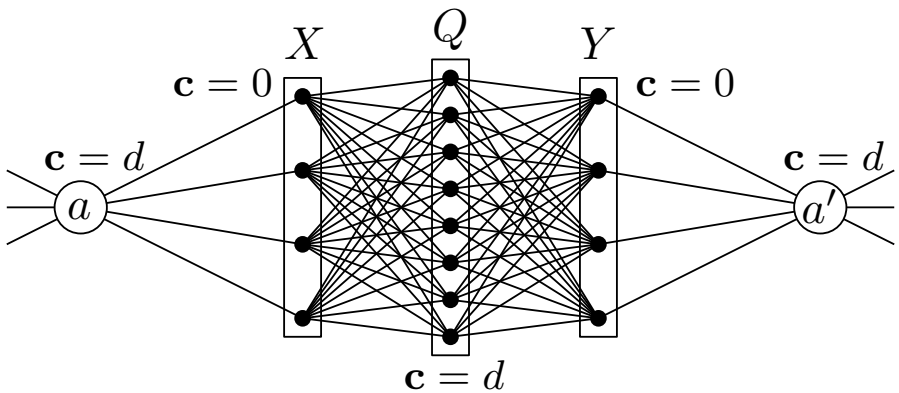




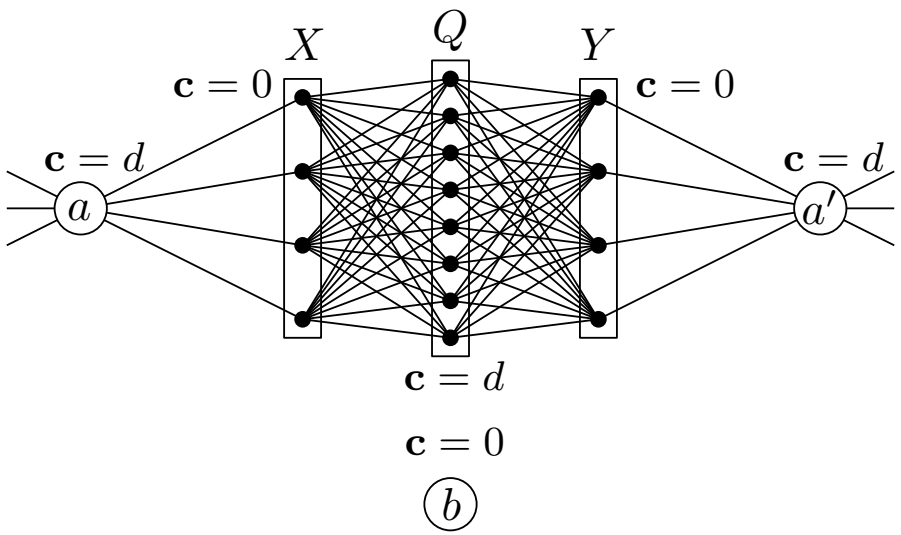
Block gadget $\hat{B}$ (II)



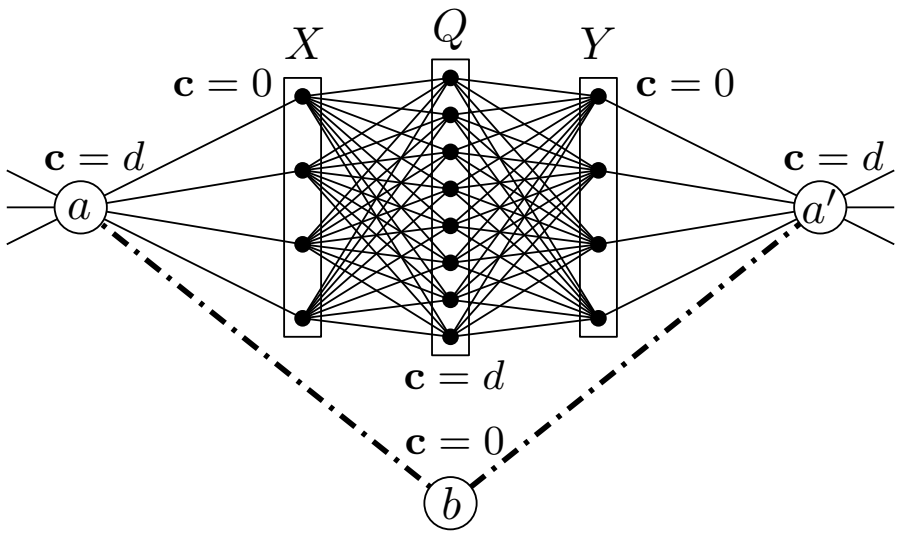
Block gadget $\hat{B}$ (III)



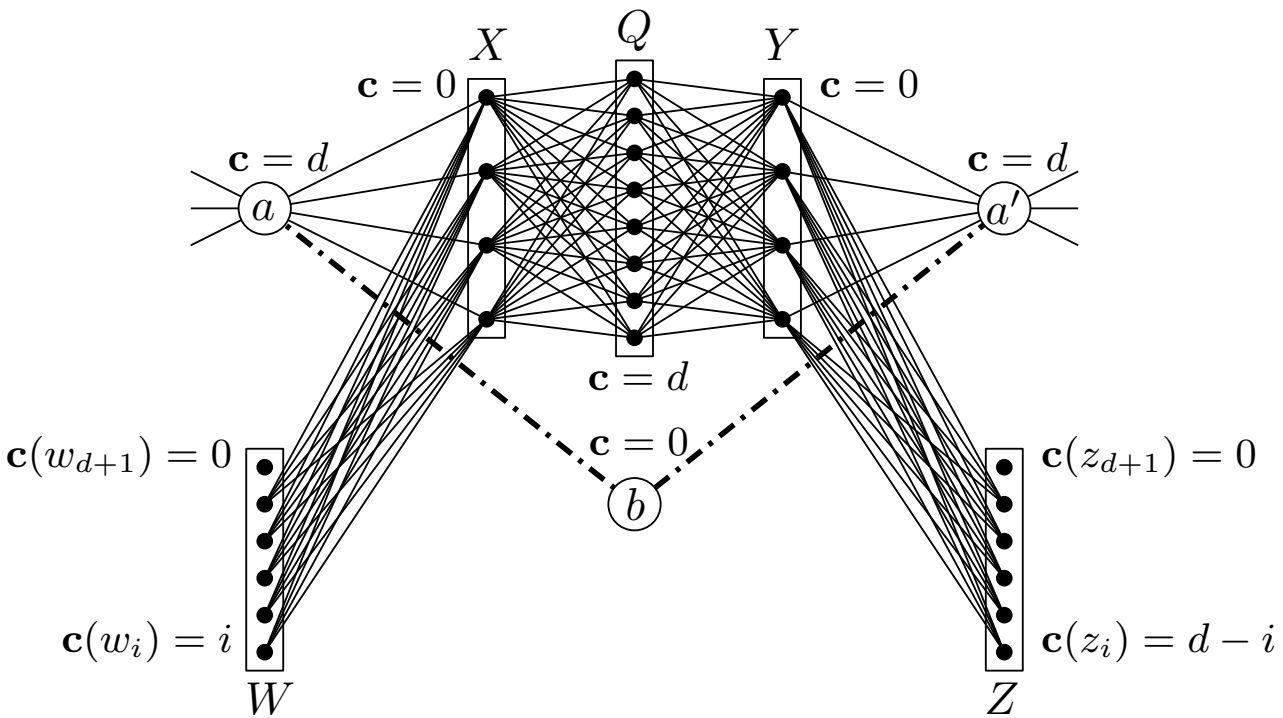
Block gadget $\hat{B}$ (IV)



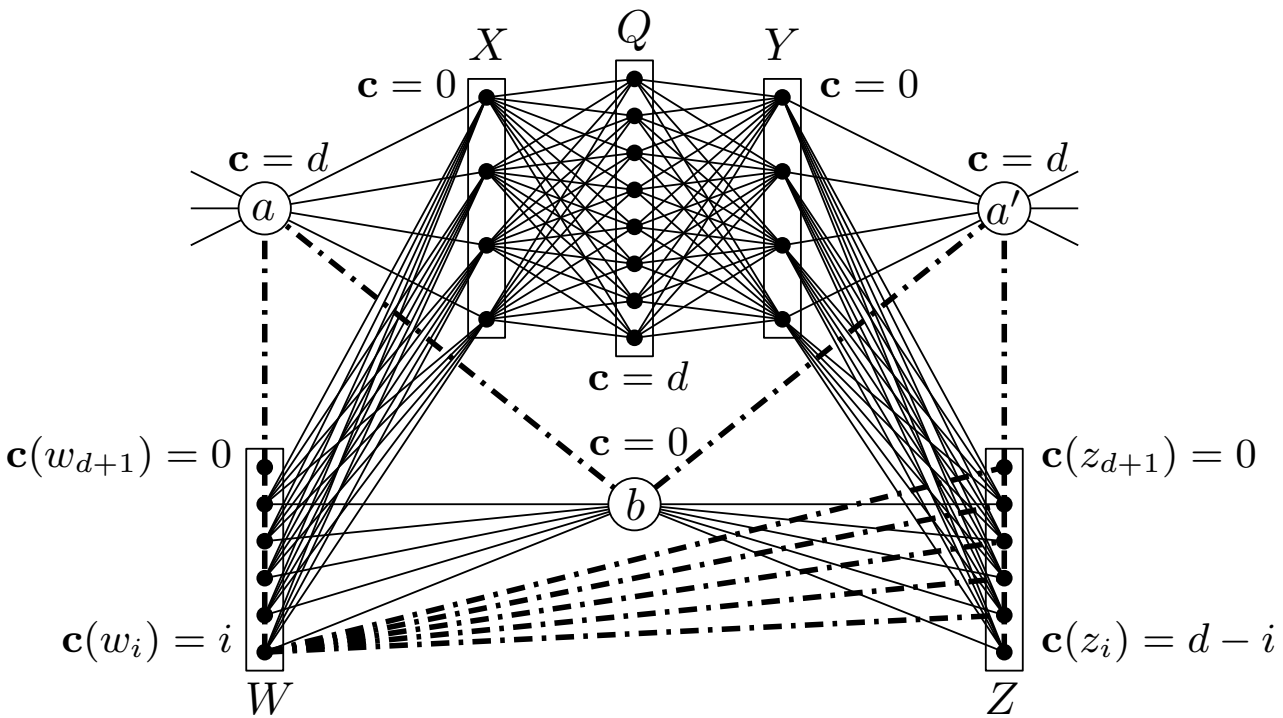
Block gadget $\hat{B}$ (V)



Block gadget $\hat{B}$ (VI)



Block gadget $\hat{B}$ (VII)



Shift (I)

$2^n \to (d+2)^{\mathrm{tw}}$

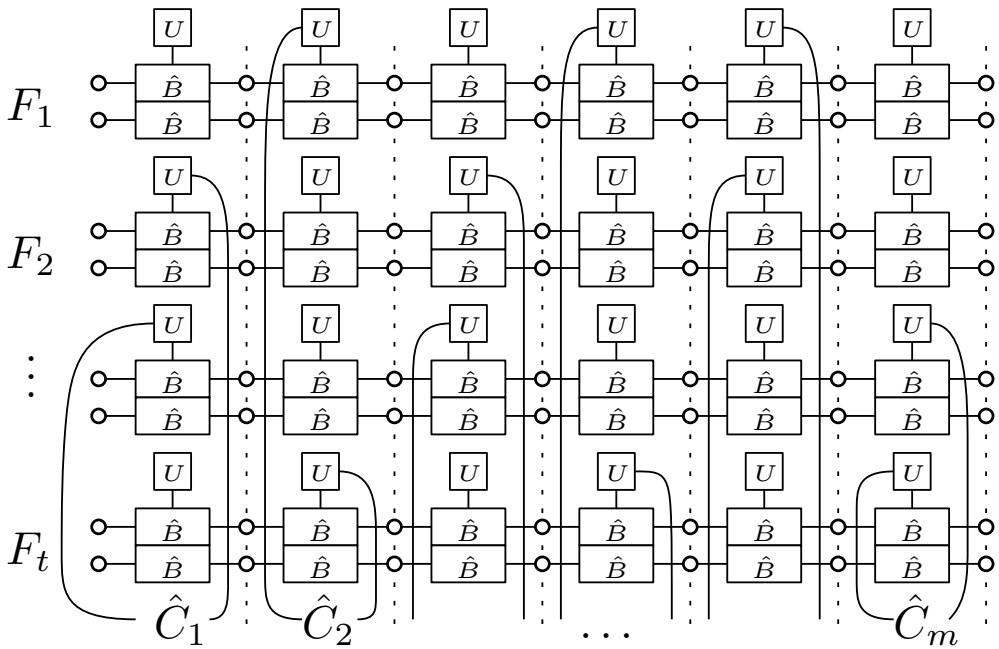
SAT: $\phi \to d$-OrientableDeletion: G, tw

n variables

$\downarrow$

$\mathrm{tw} \approx \frac{n}{\log(d+2)}$

groups



m clauses

Shift (II)

$2^n \to (d+2)^{\mathrm{tw}}$

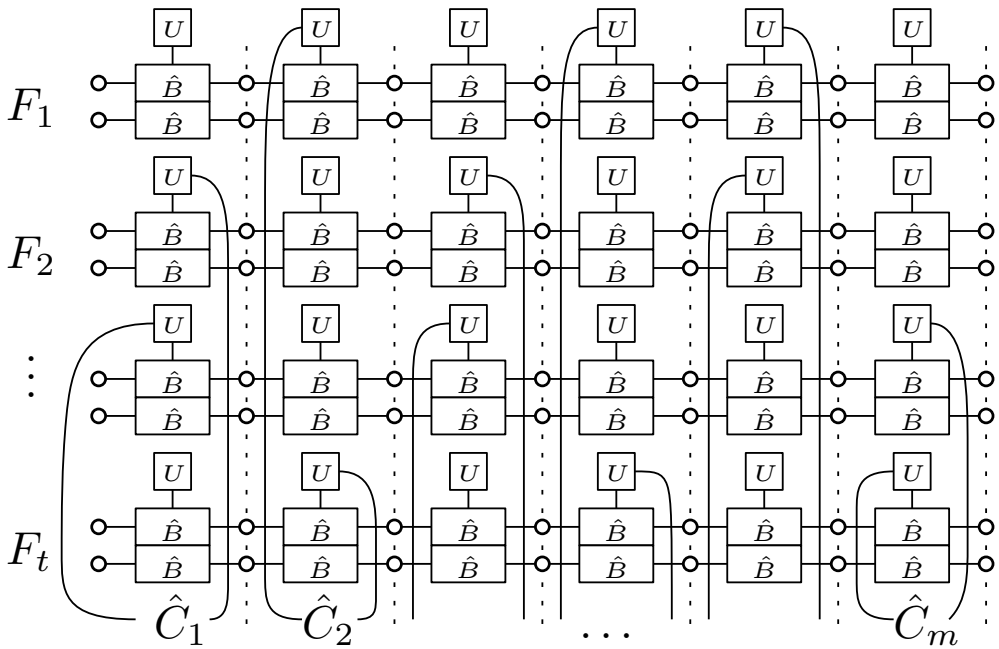
SAT: $\phi \to d$-OrientableDeletion: G, tw

n variables

$\downarrow$

$\mathrm{tw} \approx \frac{n}{\log(d+2)}$

groups



m clauses $\to\approx m(d \cdot \frac{n}{\log(d+2)} + 1)$ columns

Works!

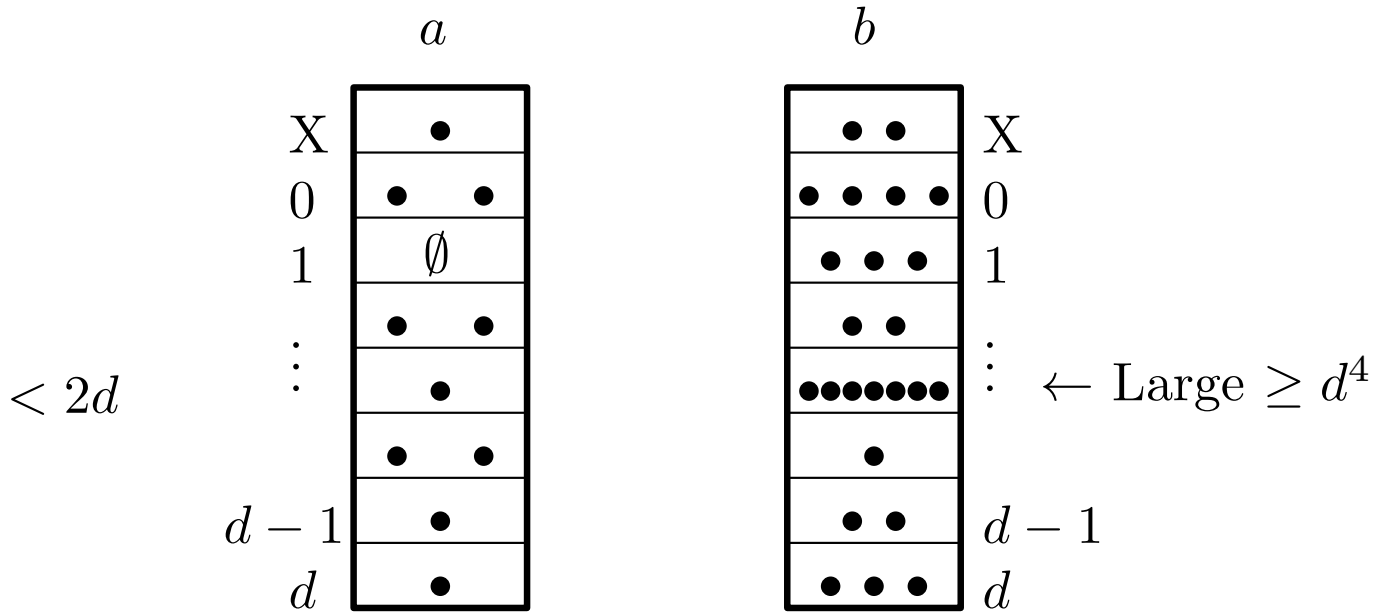
- For any fixed $d \geq 1$, if d-Orientable Deletion can be solved in $O^*((d+2-\epsilon)^{\mathrm{tw}})$ time for some $\epsilon > 0$, then there exists some $\delta > 0$, such that SAT can be solved in $O^*((2-\delta)^n)$ time
- As a corollary, for $d = 1$, this shows that the 3^{tw} algorithm for PseudoForest Deletion is optimal under the SETH

FPT algorithm (clique-width $+d$)

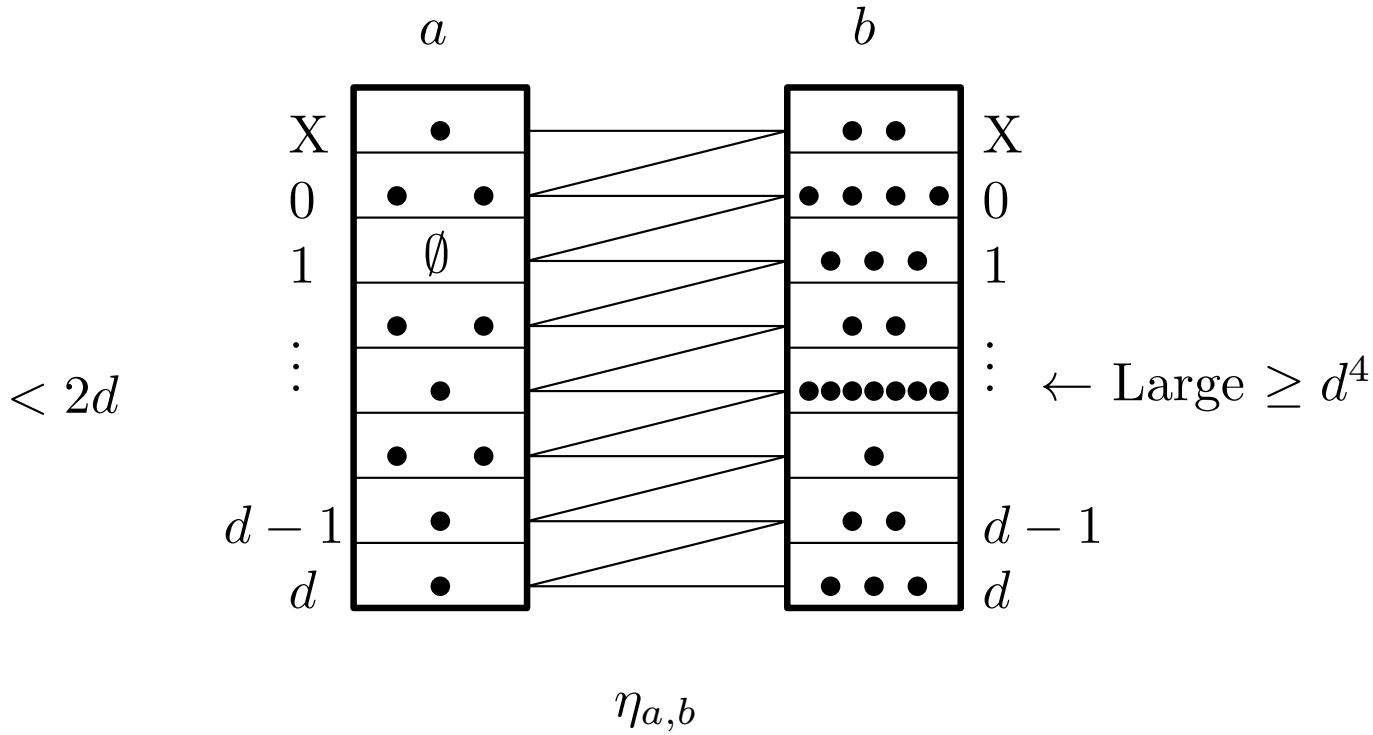
Redundancy of XP algorithm

- The previously known best algorithm runs in (XP) time $n^{O(d \cdot \mathrm{cw})}$
- Its main idea is to compute the number of vertices contained in each label that have in-degree $i \in [0, d]$ (their *in-degree-class*)
- Not all of the natural n^{d+2} *label-states* used by this DP are in fact necessary
- Only in-degree-class sizes up to a certain *threshold* (i.e. d^4) are required, since any optimal solution always orients all new edges towards the vertices of such a "large" in-degree-class

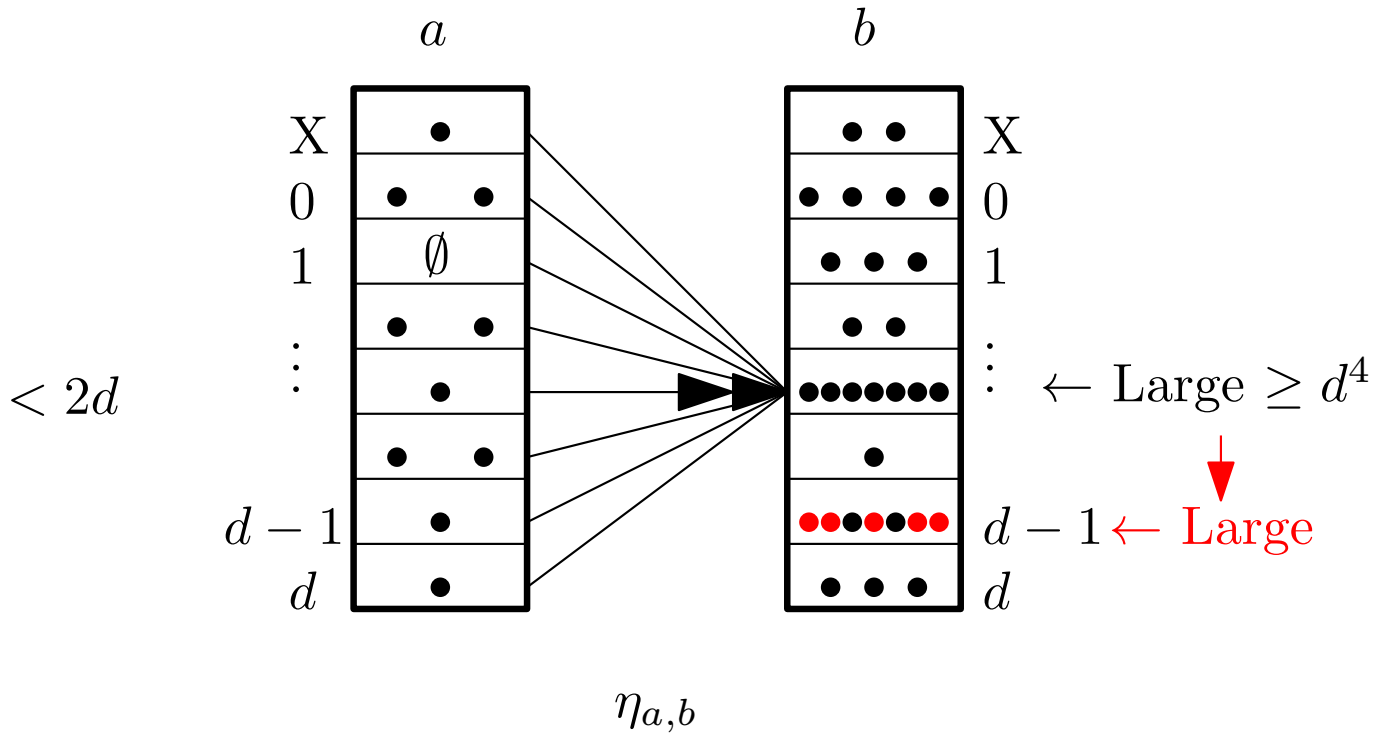
Labels and in-degree-classes



Join


a
b
X
0
1
$\vdots$
$d-1$
d
$\emptyset$
X
0
1
$\vdots$
$d-1$
d
$< 2d$
$\leftarrow$ Large $\geq d^4$
$\eta_{a,b}$

Large Shift


a
b
X
0
1
$\emptyset$
$d-1$
d
X
0
1
$d-1$
d
$< 2d$
$\leftarrow$ Large $\geq d^4$
$\leftarrow$ Large
$\eta_{a,b}$

FPT!

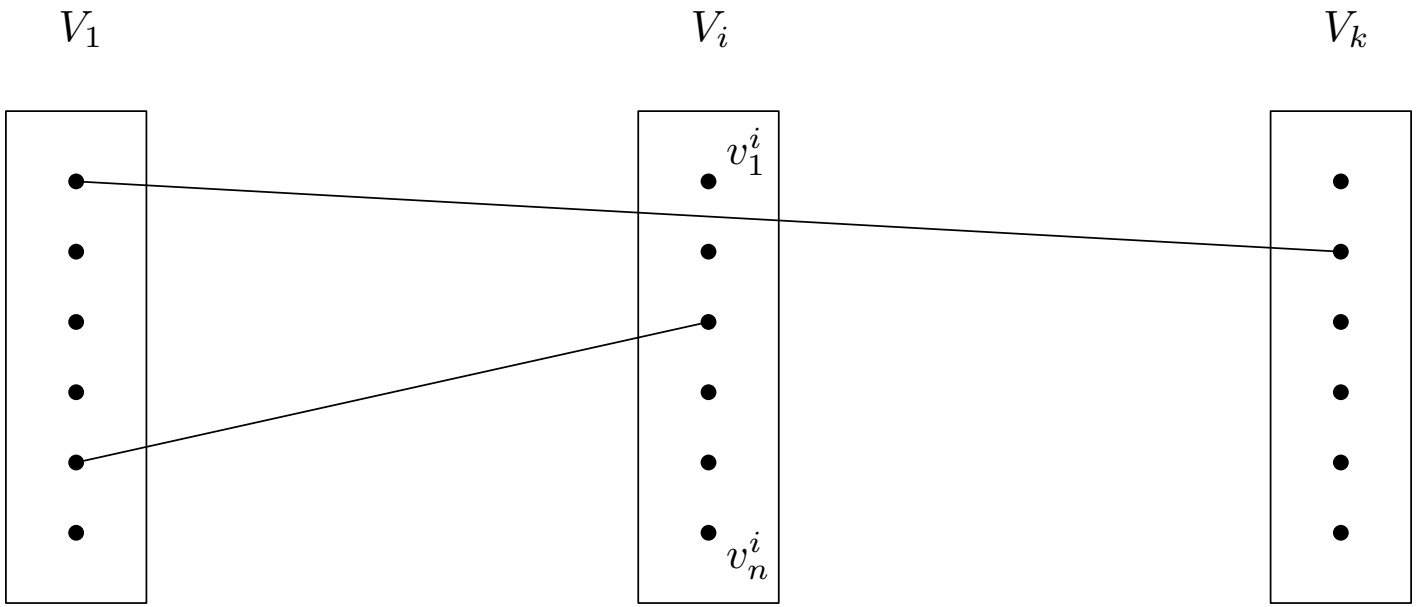
- If any optimal solution orients all edges towards the vertices of a "large" in-degree-class, it is straightforward to obtain the recursive computations for each type of node
- Instead of n^{d+2} label-states, we only require $(d^4)^{d+2}$
- This leads to an $O^*(d^{O(d \cdot \mathrm{cw})})$-time, FPT algorithm parameterized by *both* d and cw

W-hardness (clique-width $\mathbf{cw}$)

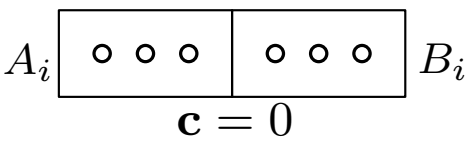
Optimality?

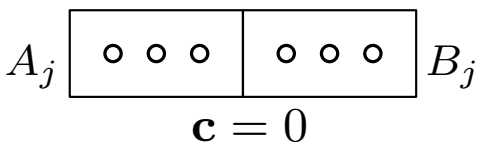
- W[1]-hard parameterized by clique-width cw only
- No $n^{o(\mathrm{cw})}$-time algorithm under the ETH
- The $d^{O(\mathrm{cw})}$-time FPT algorithm is (essentially) optimal
- Reduction from k-Multicolored Independent Set

Multicolored Independent Set

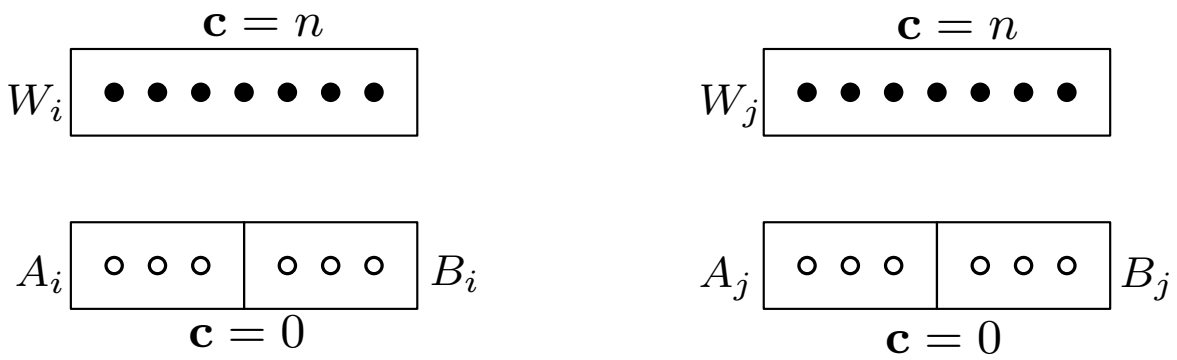


Reduction Overview (I)

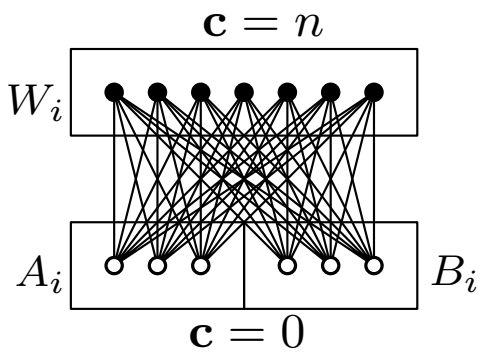


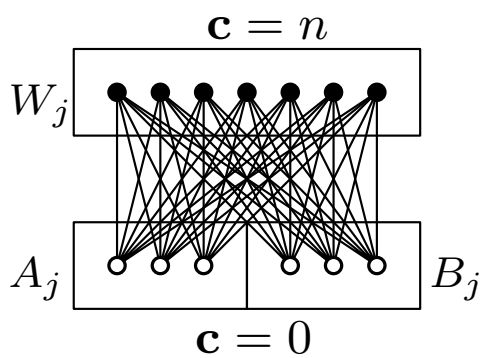


Reduction Overview (II)

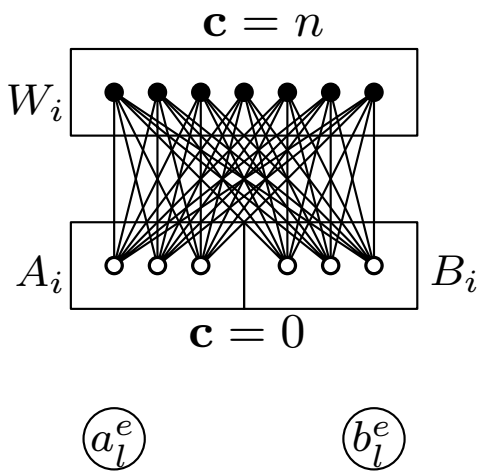


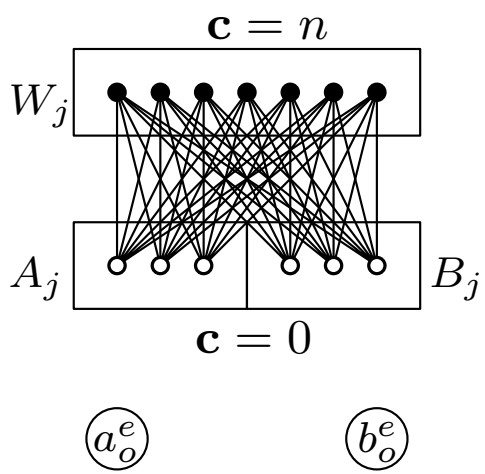
Reduction Overview (III)



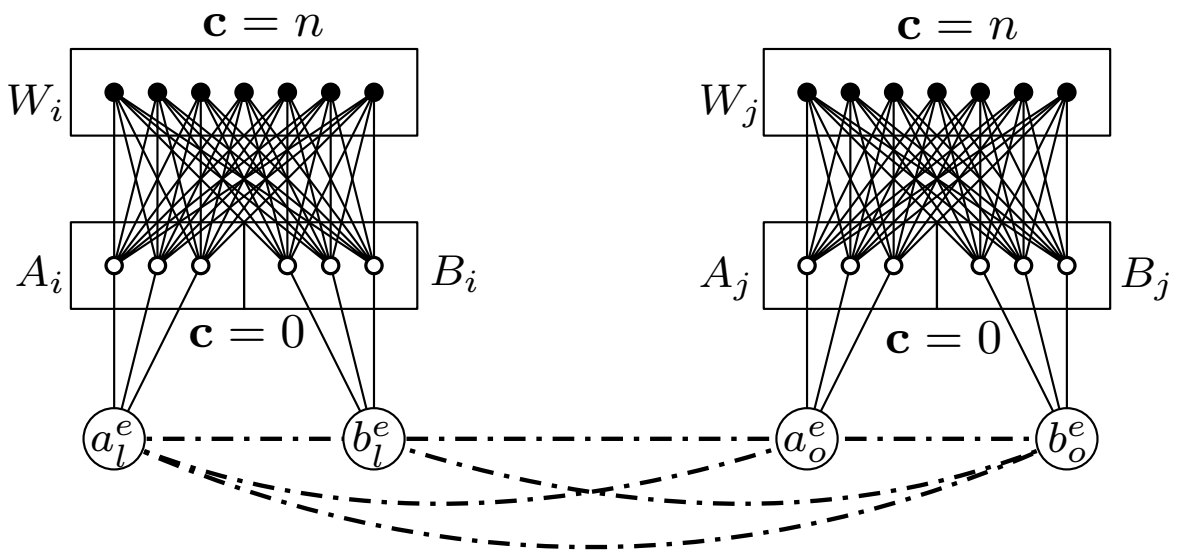


Reduction Overview (IV)

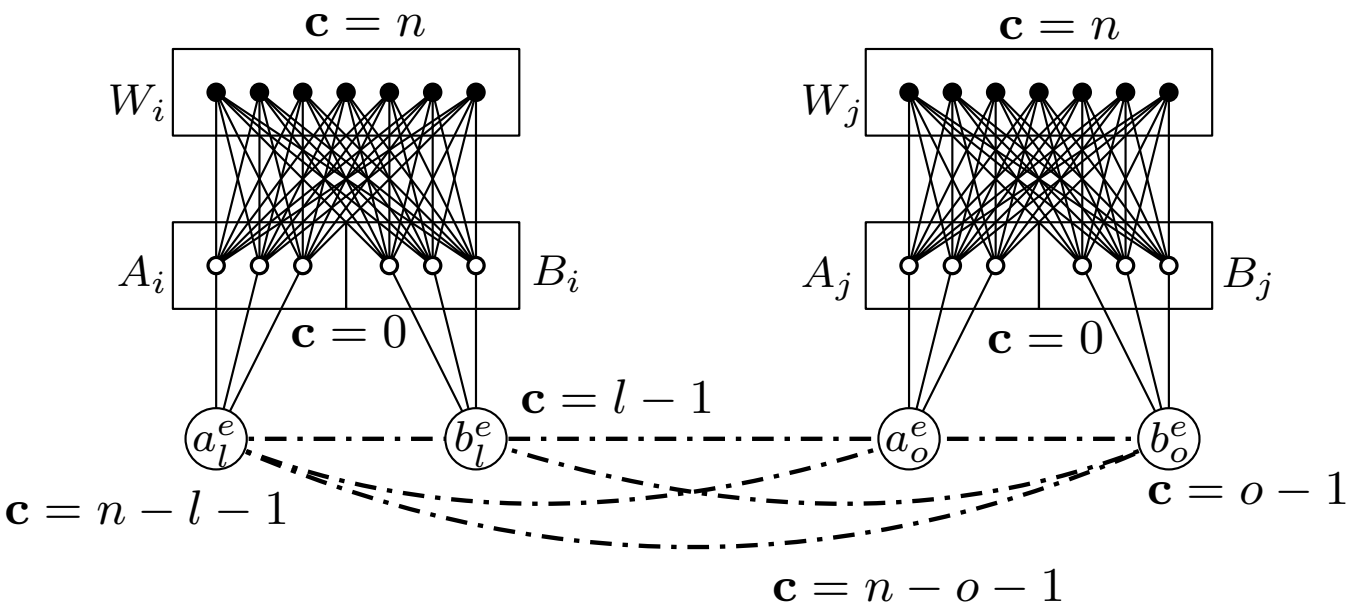




Reduction Overview (V)



Reduction Overview (VI)



Thank you!

Questions?