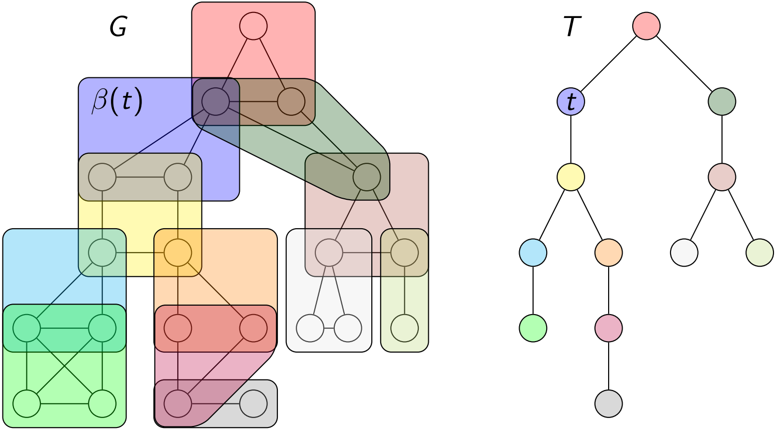
Treewidth Inapproximability and Tight ETH Lower Bound

Édouard Bonnet

ENS Lyon, LIP

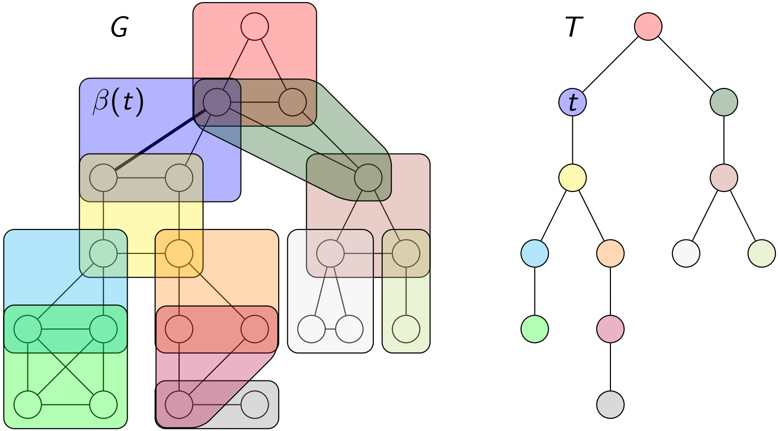
LOGALG 2025, Vienna

Tree-decomposition of G



Tree T and map $\beta : V(T) \to 2^{V(G)}$ such that

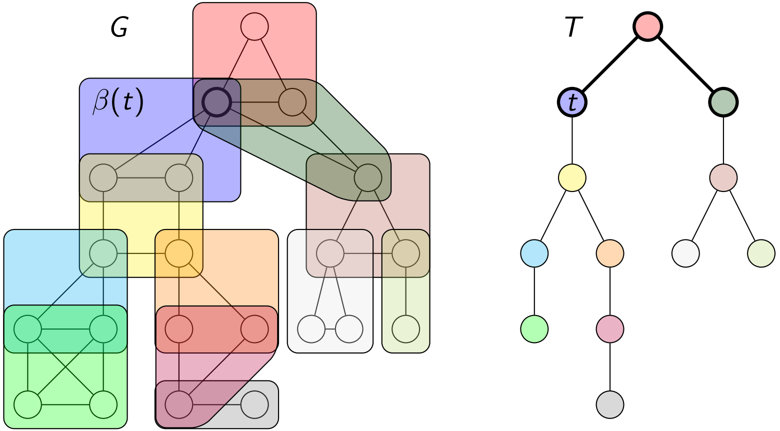
Tree-decomposition of G



Tree T and map $\beta : V(T) \to 2^{V(G)}$ such that

- Every edge of G has both endpoints in some *bag* $\beta(t)$, and

Tree-decomposition of G



Tree T and map $\beta : V(T) \to 2^{V(G)}$ such that

- Every edge of G has both endpoints in some *bag* $\beta(t)$, and
- the *trace* of every vertex of G in T is a non-empty subtree.

Treewidth

Minimum largest bag size over all tree decompositions minus 1

- rediscovered several times in the 70's and 80's...
- made central by the *Graph Minors* series
- $f(\text{tw})n$-time algorithms for MSO-definable problems

Treewidth

Minimum largest bag size over all tree decompositions minus 1

- rediscovered several times in the 70's and 80's...
- made central by the *Graph Minors* series
- $f(\text{tw})n$-time algorithms for MSO-definable problems

Computing a tree decomposition?

Treewidth

Minimum largest bag size over all tree decompositions minus 1

- rediscovered several times in the 70's and 80's...
- made central by the *Graph Minors* series
- $f(\text{tw})n$-time algorithms for MSO-definable problems

Computing a tree decomposition? NP-hard but various algorithms

width $2\text{tw}+1$ in $2^{O(\text{tw})}n$

width $5\text{tw}+4$ in $2^{6.76\text{tw}}n\log n$

width tw in $2^{O(\text{tw}^2)}n^4$

width $O(\text{tw}\sqrt{\log \text{tw}})$ in $n^{O(1)}$

width tw in 1.74^n

width tw in $2^{O(\text{tw}^3)}n$

width $(1+\varepsilon)\text{tw}$ in $2^{O(\frac{\text{tw}\log\text{tw}}{\varepsilon})}n^4$

New hardness results for Treewidth

Ruling out a PTAS:

Theorem
1.00005-approximating Treewidth *is NP-hard.*

New hardness results for Treewidth

Ruling out a PTAS:

Theorem
1.00005-approximating Treewidth *is NP-hard.*

Tight ETH lower bound:

Theorem
Under the ETH, Treewidth *needs* $2^{\Omega(n)}$ *time on n-vertex graphs.*

New hardness results for Treewidth

Ruling out a PTAS:

Theorem
1.00005-approximating Treewidth *is NP-hard.*

Tight ETH lower bound:

Theorem
Under the ETH, Treewidth *needs* $2^{\Omega(n)}$ *time on n-vertex graphs.*

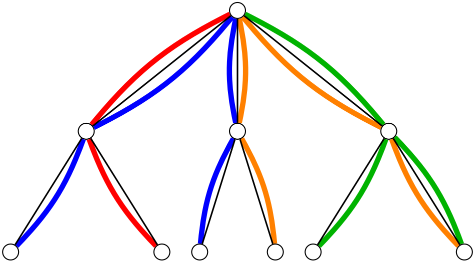
No approximation scheme in subexponential time:

Theorem
Unless the ETH fails, there are $\delta > 1$ *and* $c > 0$ *such that* δ*-approximating* Treewidth *requires* $2^{\Omega(n/\log^c n)}$ *time.*

Every clique is contained in a bag

Fact

For every graph G, tree-decomposition (T, β) of G, and clique X of G, there is a $t \in V(T)$ such that $X \subseteq \beta(t)$.

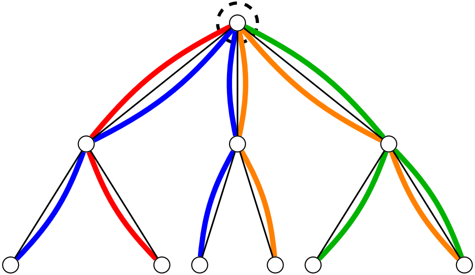


Helly property for subtrees in a tree

Every clique is contained in a bag

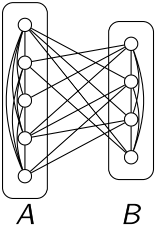
Fact

For every graph G, tree-decomposition (T, β) of G, and clique X of G, there is a $t \in V(T)$ such that $X \subseteq \beta(t)$.



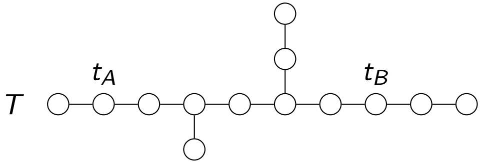
Helly property for subtrees in a tree

Tree-decompositions of co-bipartite graphs



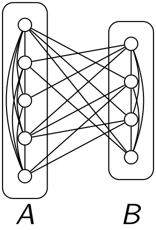
Fact

For every co-bipartite graph G, $tw(G) = pw(G)$.



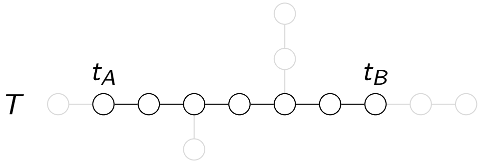
Let t_A, t_B be such that $A \subseteq \beta(t_A), B \subseteq \beta(t_B)$

Tree-decompositions of co-bipartite graphs



Fact

For every co-bipartite graph G, $tw(G) = pw(G)$.



Bags that are not on the t_A–t_B path are useless

NP-hardness of Treewidth

3-SAT $\to_\ell$ Max Cut $\to_\ell$ Cutwidth $\to_P$ Path/Treewidth

No known linear reduction for the last leg

Theorem (Arnborg, Corneil, Proskurowski '87)

Polytime reduction from Cutwidth *on n-vertex graphs of max degree Δ to* Treewidth *on $O(n\Delta)$-vertex co-bipartite graphs.*

NP-hardness of Treewidth

3-SAT $\to_\ell$ Max Cut $\to_\ell$ Cutwidth $\to_P$ Path/Treewidth

No known linear reduction for the last leg

Theorem (Arnborg, Corneil, Proskurowski '87)

Polytime reduction from Cutwidth *on n-vertex graphs of max degree* Δ *to* Treewidth *on* $O(n\Delta)$*-vertex co-bipartite graphs.*

?

3-SAT $\to_\ell$ Cutwidth
on bounded-degree graphs

NP-hardness of TREEWIDTH

3-SAT $\to_\ell$ MAX CUT $\to_\ell$ CUTWIDTH $\to_P$ PATH/TREEWIDTH

No known linear reduction for the last leg

Theorem (Arnborg, Corneil, Proskurowski '87)

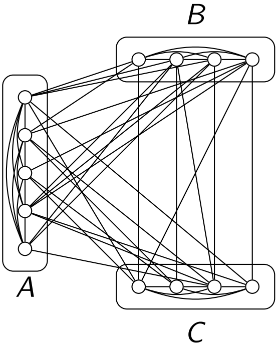
Polytime reduction from CUTWIDTH *on n-vertex graphs of max degree Δ to* TREEWIDTH *on $O(n\Delta)$-vertex co-bipartite graphs.*

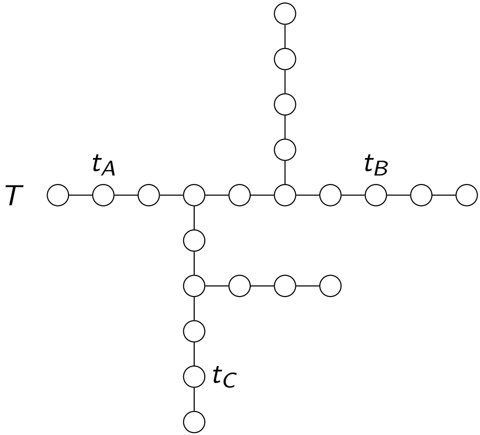
?

3-SAT $\to_\ell$ CUTWIDTH on bounded-degree graphs

co-tripartite graphs

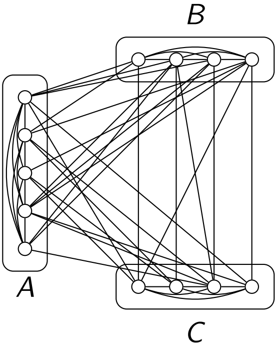
Tree-decompositions of co-tripartite graphs

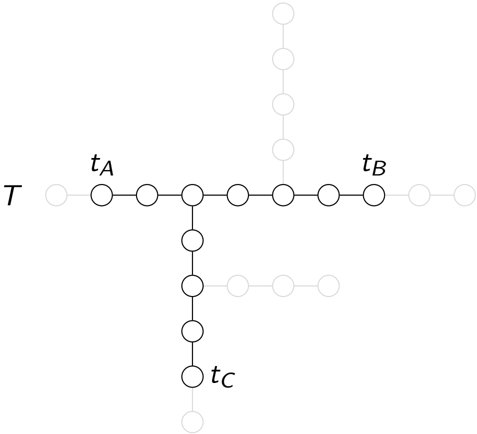




Let t_A, t_B, t_C be such that $A \subseteq \beta(t_A), B \subseteq \beta(t_B), C \subseteq \beta(t_C)$

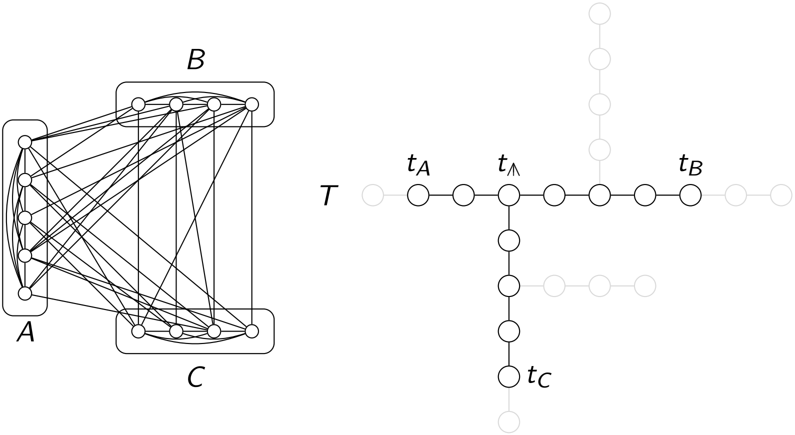
Tree-decompositions of co-tripartite graphs





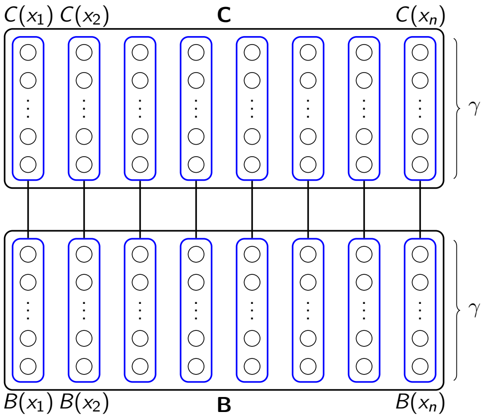
Bags that are not on the t_A–t_B, t_A–t_C, t_B–t_C paths are useless

Tree-decompositions of co-tripartite graphs



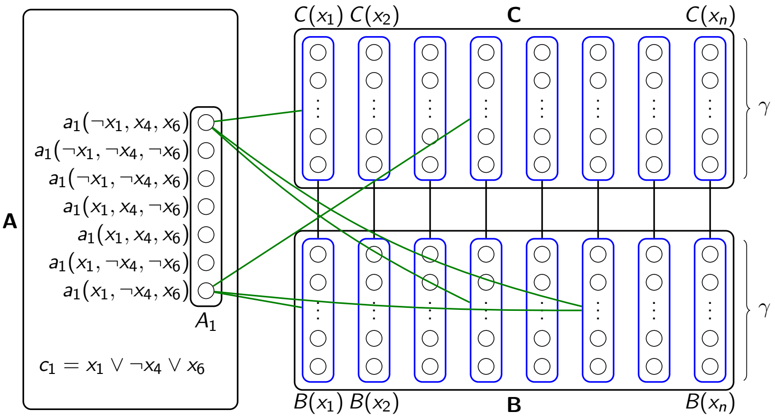
$\beta(t_{\Lsh})$ has to be a vertex cover of $E(A, B) \cup E(A, C) \cup E(B, C)$

3-SAT $\to_\ell$ TREEWIDTH on co-tripartite graphs



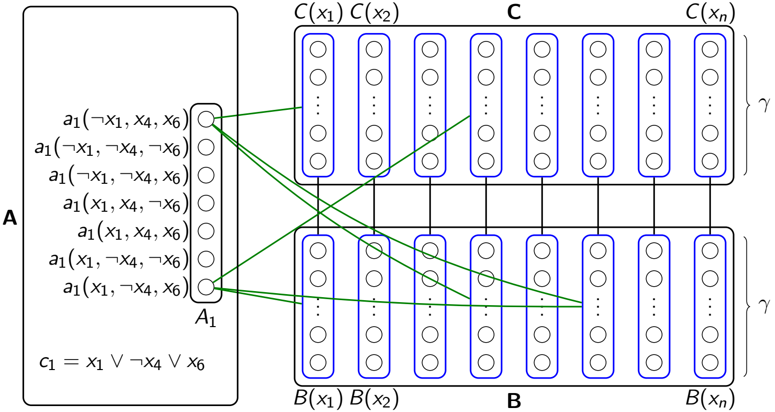
We encode variables in two cliques B (true) and C (false):
each variable is a $K_{\gamma,\gamma}$ biclique, γ is 4 times the max occurrence

3-SAT $\rightarrow_\ell$ Treewidth on co-tripartite graphs



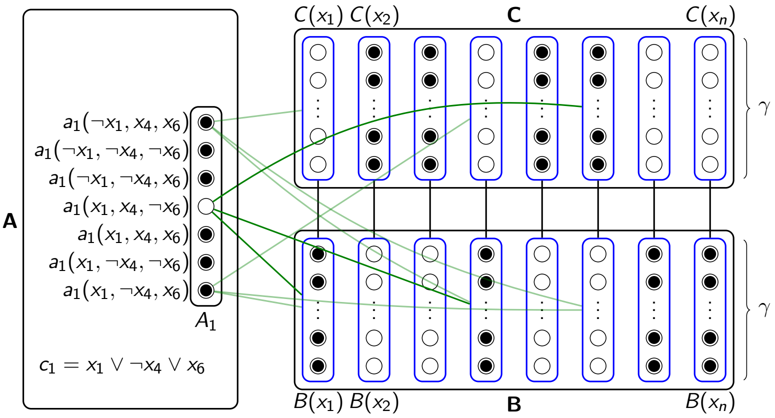
The *clause* vertices are in a third clique A: for each 3-clause, add one vertex per satisfying assignment linked to its literals

$3\text{-SAT} \rightarrow_\ell$ TREEWIDTH on co-tripartite graphs



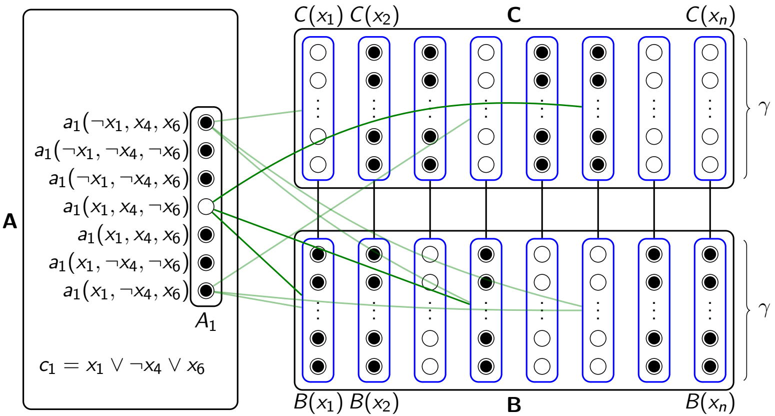
From an n-variable m-clause 3-SAT formula φ, builds a graph $G := G(\varphi)$ with $7m + 2\gamma n$ vertices

If the formula is satisfiable, $\mathrm{tw}(G) \leqslant \gamma n + 6m + \gamma - 1$



$\beta(t_\wedge)$ comprises the *variable* vertices of a satisfying assignment $\mathcal{A}$, and all the *clause* vertices but the m corresponding to $\mathcal{A}$

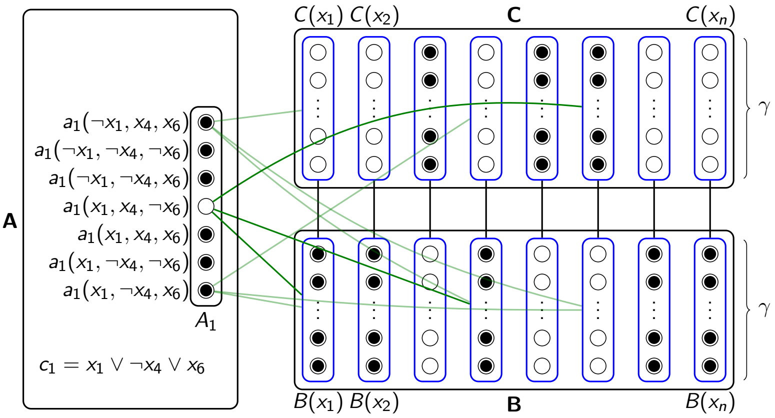
If the formula is satisfiable, $\mathrm{tw}(G) \leqslant \gamma n + 6m + \gamma - 1$



$$|\beta(t_\wedge)| = \gamma n + 6m$$

toward t_B: add $B(x_i)$ and remove $C(x_i)$ for each x_i set to false

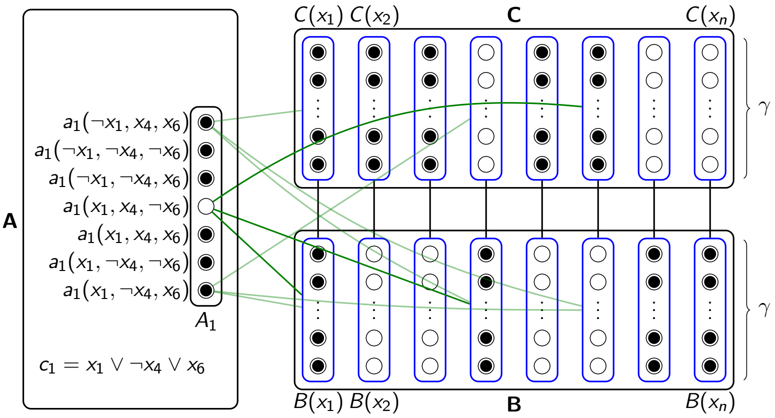
If the formula is satisfiable, $\mathrm{tw}(G) \leqslant \gamma n + 6m + \gamma - 1$



$$|\beta(t_\wedge)| = \gamma n + 6m$$

toward t_B: add $B(x_i)$ and remove $C(x_i)$ for each x_i set to false

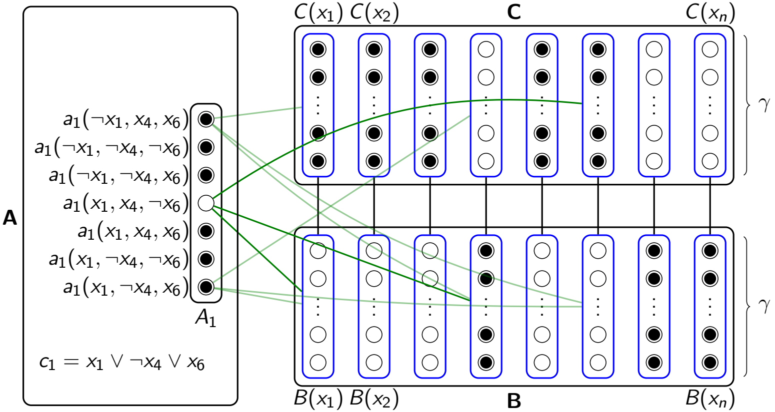
If the formula is satisfiable, $\text{tw}(G) \leqslant \gamma n + 6m + \gamma - 1$



$$|\beta(t_\wedge)| = \gamma n + 6m$$

toward t_C: add $C(x_i)$ and remove $B(x_i)$ for each x_i set to true

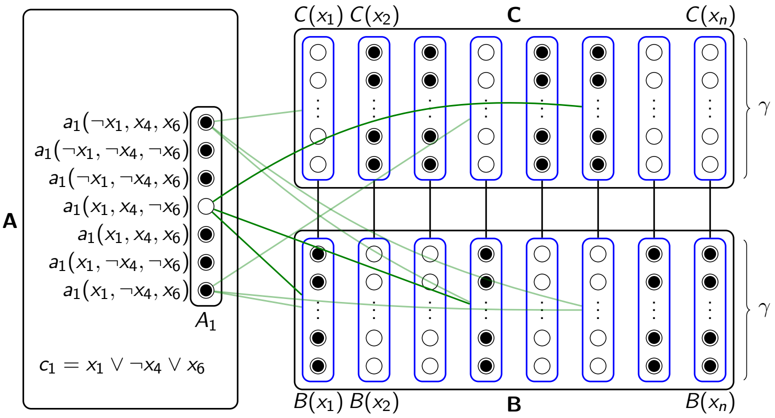
If the formula is satisfiable, $\text{tw}(G) \leqslant \gamma n + 6m + \gamma - 1$



$$|\beta(t_\wedge)| = \gamma n + 6m$$

toward t_C: add $C(x_i)$ and remove $B(x_i)$ for each x_i set to true

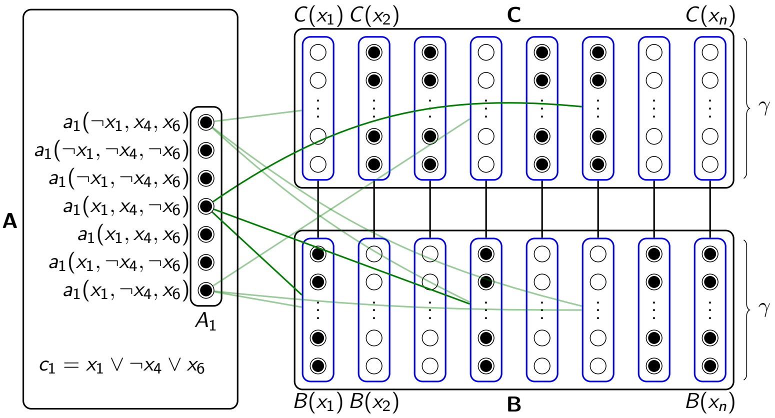
If the formula is satisfiable, $\text{tw}(G) \leqslant \gamma n + 6m + \gamma - 1$



$$|\beta(t_\wedge)| = \gamma n + 6m$$

toward t_A: add occurrences of x_i in A and remove $B(x_i)$ or $C(x_i)$

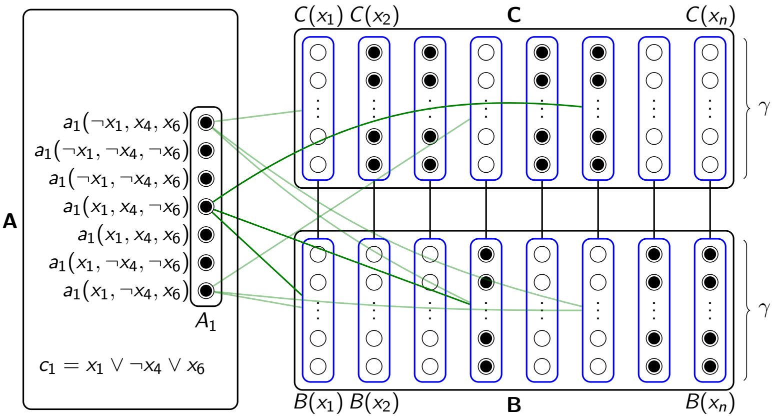
If the formula is satisfiable, $\mathrm{tw}(G) \leqslant \gamma n + 6m + \gamma - 1$



$$|\beta(t_\wedge)| = \gamma n + 6m$$

toward t_A: add occurrences of x_i in A and remove $B(x_i)$ or $C(x_i)$

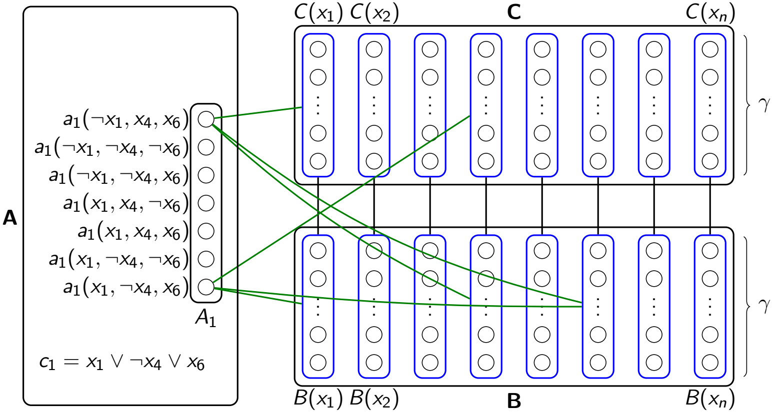
If the formula is satisfiable, $\text{tw}(G) \leqslant \gamma n + 6m + \gamma - 1$



$$|\beta(t_\wedge)| = \gamma n + 6m$$

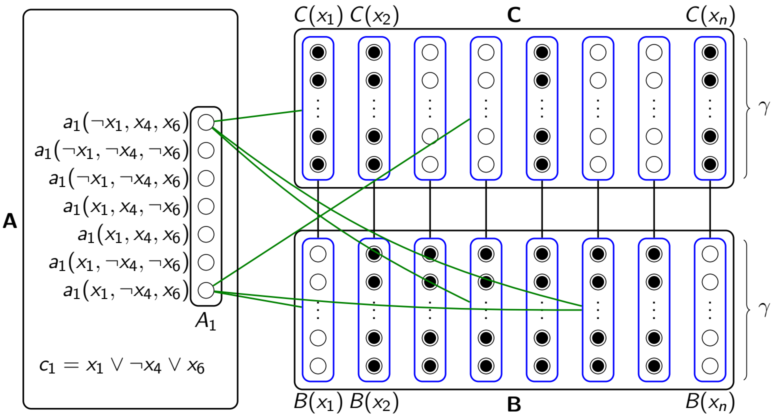
toward t_A: add occurrences of x_i in A and remove $B(x_i)$ or $C(x_i)$

If $\leqslant m'$ clauses are satisfiable, $\text{tw}(G) \geqslant \gamma n + 7m - m' - 1$



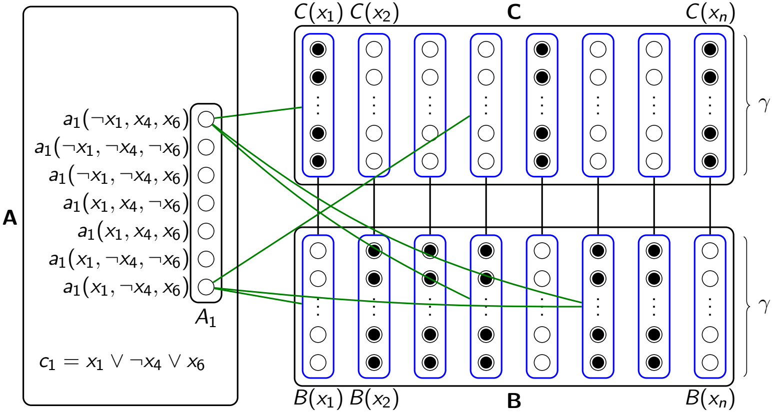
$\beta(t_{\pitchfork})$ is a vertex cover of $E(A, B) \cup E(A, C) \cup E(B, C)$

If $\leqslant m'$ clauses are satisfiable, $\text{tw}(G) \geqslant \gamma n + 7m - m' - 1$



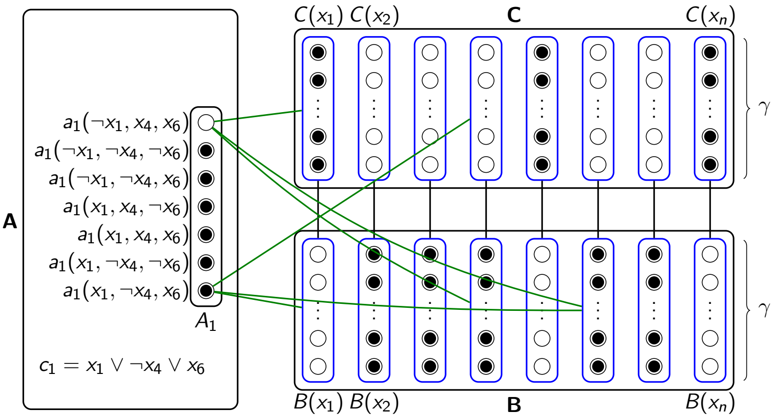
No point in partially intersecting some $B(x_i)$ or $C(x_i)$

If $\leqslant m'$ clauses are satisfiable, $\mathrm{tw}(G) \geqslant \gamma n + 7m - m' - 1$



By def of γ, a min vertex cover contains exactly one of $B(x_i), C(x_i)$

If $\leqslant m'$ clauses are satisfiable, $\text{tw}(G) \geqslant \gamma n + 7m - m' - 1$



Then, one *clause* vertex can be spared per satisfied clause

TREEWIDTH is APX-hard

Theorem
1.00005-approximating TREEWIDTH *is NP-hard.*

TREEWIDTH is APX-hard

Theorem

1.00005-approximating TREEWIDTH *is NP-hard.*

Theorem (Berman, Karpinski, Scott '03)

For any $\varepsilon > 0$, it is NP-hard to distinguish m-clause 4-OCC 3-SAT *instances where*

- *at least $(1-\varepsilon)m$ clauses are satisfiable, or*
- *at most $(\frac{1015}{1016}+\varepsilon)m$ clauses are satisfiable.*

TREEWIDTH is APX-hard

Theorem

1.00005-approximating TREEWIDTH *is NP-hard.*

Theorem (Berman, Karpinski, Scott '03)

For any $\varepsilon > 0$, it is NP-hard to distinguish m-clause 4-OCC 3-SAT *instances where*

- *at least $(1 - \varepsilon)m$ clauses are satisfiable, or*
- *at most $(\frac{1015}{1016} + \varepsilon)m$ clauses are satisfiable.*

Hence to distinguish between treewidth

- at most $\gamma\frac{3m}{4} + (6 + \varepsilon)m + \gamma - 1$, or
- at least $\gamma\frac{3m}{4} + (7 - \frac{1015}{1016} - \varepsilon)m - 1$.

$\gamma := 14$, and we get the claimed inapproximability ratio

Treewidth is ETH-hard

Theorem

Under the ETH, Treewidth *needs* $2^{\Omega(n)}$ *time on n-vertex graphs.*

TREEWIDTH is ETH-hard

Theorem

Under the ETH, TREEWIDTH *needs* $2^{\Omega(n)}$ *time on n-vertex graphs.*

Theorem (Impagliazzo, Paturi, Zane '01)

*There is a constant B such that n-variable B-*OCC 3-SAT *requires* $2^{\Omega(n)}$ *time, unless the ETH fails.*

TREEWIDTH is ETH-hard

Theorem

Under the ETH, TREEWIDTH *needs* $2^{\Omega(n)}$ *time on n-vertex graphs.*

Theorem (Impagliazzo, Paturi, Zane '01)

*There is a constant B such that n-variable B-*OCC 3-SAT *requires* $2^{\Omega(n)}$ *time, unless the ETH fails.*

$\gamma := 4B$, get m-clause φ by $\gamma + 1$ duplications

$G := G(\varphi)$ has at most $7B(\gamma + 1)n + 2\gamma(\gamma + 1)n = O(n)$ vertices

- φ satisfiable $\Rightarrow \text{tw}(G) \leqslant \gamma(\gamma + 1)n + 6m + \gamma - 1$
- φ unsatisfiable $\Rightarrow \text{tw}(G) \geqslant \gamma(\gamma + 1)n + 6m + \gamma$.

Treewidth has no subexponential approximation scheme

Theorem

Unless the ETH fails, there are $\delta > 1$ and $c > 0$ such that δ-approximating Treewidth *requires $2^{\Omega(n/\log^c n)}$ time.*

$\textsc{Treewidth}$ has no subexponential approximation scheme

Theorem

Unless the ETH fails, there are $\delta > 1$ and $c > 0$ such that δ-approximating $\textsc{Treewidth}$ requires $2^{\Omega(n/\log^c n)}$ time.

Theorem (Bafna, Minzer, Vyas, Yun '25)

For any $r \in (\frac{7}{8}, 1)$, there is a constant $c := c(r)$ such that any r-approximation algorithm for m-clause $3\text{-SAT-}\log^c m$ requires $2^{\Omega(m/\log^c m)}$ time, unless the ETH fails.

Treewidth has no subexponential approximation scheme

Theorem
Unless the ETH fails, there are $\delta > 1$ and $c > 0$ such that δ-approximating Treewidth *requires $2^{\Omega(n/\log^c n)}$ time.*

Theorem (Bafna, Minzer, Vyas, Yun '25)
For any $r \in (\frac{7}{8}, 1)$, there is a constant $c := c(r)$ such that any r-approximation algorithm for m-clause $3\text{-SAT-}\log^c m$ requires $2^{\Omega(m/\log^c m)}$ time, unless the ETH fails.

Hence the following algorithm is essentially best possible

Theorem (Korhonen, Lokshtanov '23)
For any $\varepsilon > 0$, a tree-decomposition of G of width $(1+\varepsilon)tw(G)$ can be computed in $2^{O(\frac{tw(G)\log tw(G)}{\varepsilon})}n^4$ time.

Perspectives

From the same hardness of VERTEX COVER in tripartite graphs:

Theorem (B., Neuen, Sokołowski '25)

1.0003-approximating TREEDEPTH *is NP-hard. Under the ETH,* TREEDEPTH *needs* $2^{\Omega(n)}$ *time on n-vertex graphs.*

- Tighten the gap between 1.00005 and $O(\sqrt{\log \mathrm{tw}})$.
- Improved parameterized lower bounds?

Perspectives

From the same hardness of VERTEX COVER in tripartite graphs:

Theorem (B., Neuen, Sokołowski '25)

1.0003-approximating TREEDEPTH *is NP-hard. Under the ETH,* TREEDEPTH *needs* $2^{\Omega(n)}$ *time on n-vertex graphs.*

- Tighten the gap between 1.00005 and $O(\sqrt{\log \mathrm{tw}})$.
- Improved parameterized lower bounds?

Thank you for your attention!