Signed Tree Models

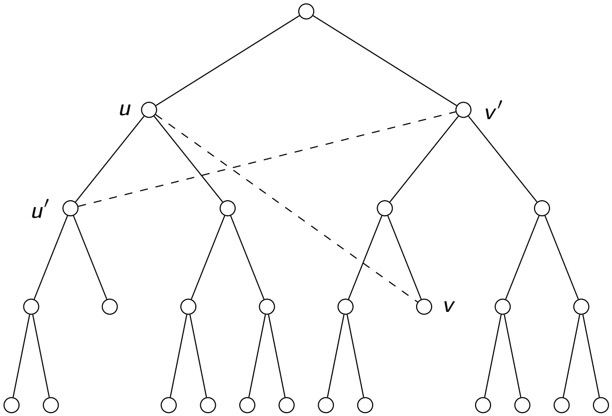
Édouard Bonnet
joint works with Julien Duron, John Sylvester, Viktor Zamaraev,
and with Colin Geniet, Eunjung Kim, and Sungmin Moon

ENS Lyon, LIP

Warsaw Seminar, March 27, 2026

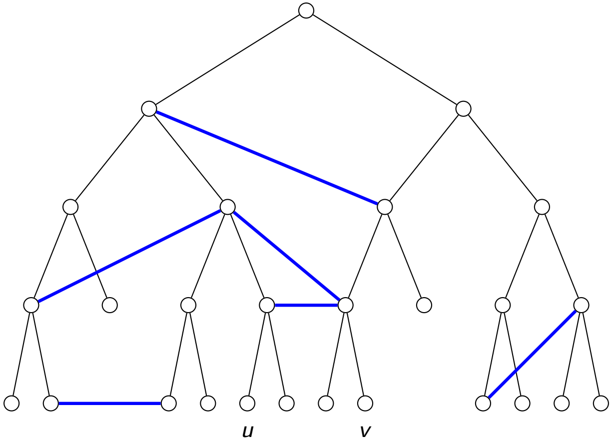
Crossing pairs in a tree

$uv, u'v'$ such that $u \prec u'$ and $v' \prec v$



Unique cover edge of a pair

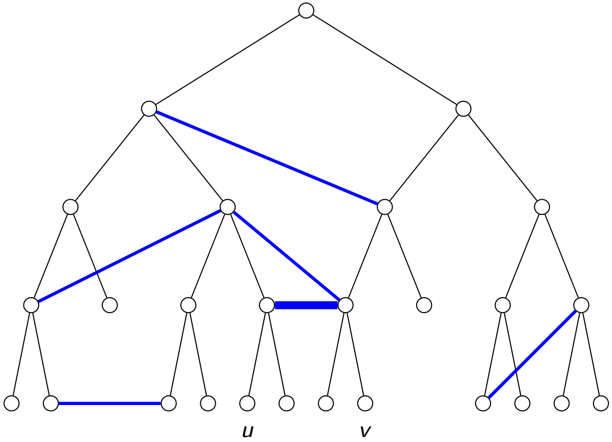
$u'v' \in E$ covers uv if $u' \preceq u$, $v' \preceq v$,
and $\neg\exists u''v'' \in E \setminus \{u'v'\}$ with $u' \preceq u'' \preceq u$ and $v' \preceq v'' \preceq v$



Tree + set E of non-crossing transversal edges

Unique cover edge of a pair

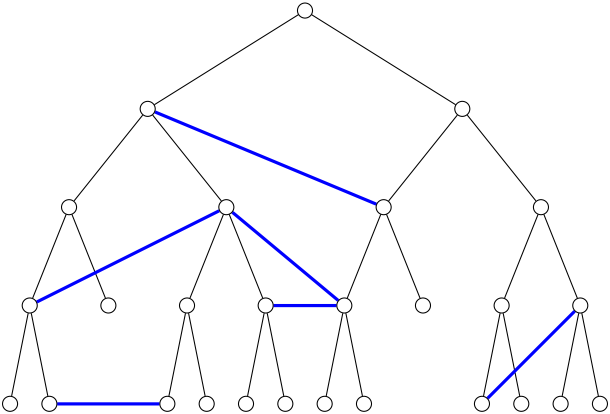
$u'v' \in E$ covers uv if $u' \preceq u$, $v' \preceq v$,
and $\neg\exists u''v'' \in E \setminus \{u'v'\}$ with $u' \preceq u'' \preceq u$ and $v' \preceq v'' \preceq v$



Tree + set E of non-crossing transversal edges

Unique cover edge of a pair

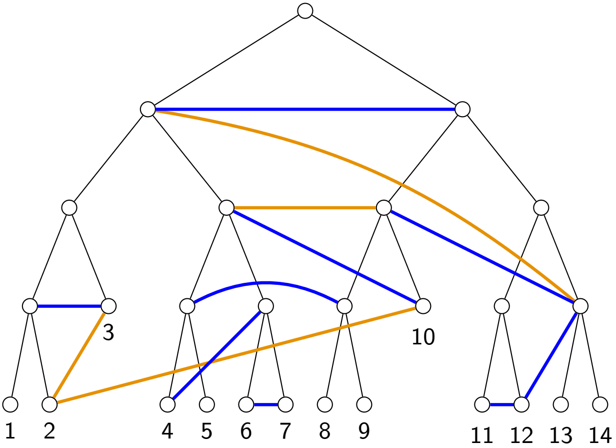
$u'v' \in E$ covers uv if $u' \preceq u$, $v' \preceq v$,
and $\neg\exists u''v'' \in E \setminus \{u'v'\}$ with $u' \preceq u'' \preceq u$ and $v' \preceq v'' \preceq v$



Tree + set E of non-crossing transversal edges

Signed tree model

Tree T + transversal anti-edges A + transversal edges B
No crossing pairs in $A \cup B$

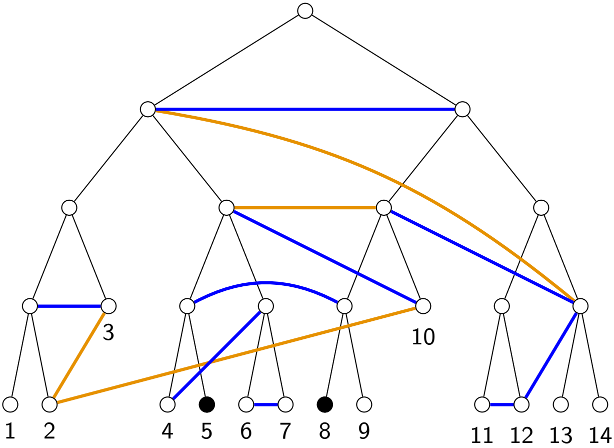


graph G at the leaves with $uv \in E(G) \Leftrightarrow uv$ covered by $u'v' \in B$

Signed tree model

Tree T + transversal anti-edges A + transversal edges B

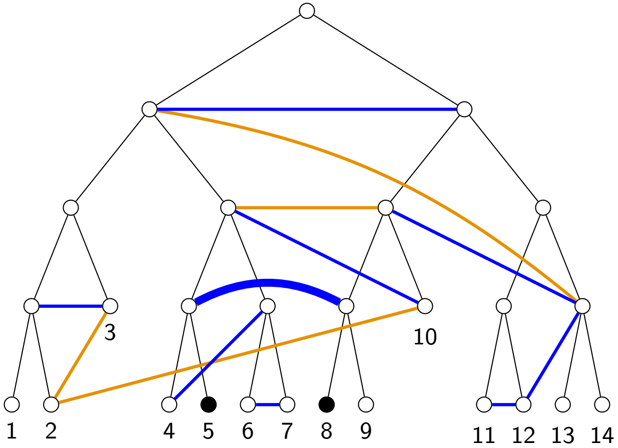
No crossing pairs in $A \cup B$



graph G at the leaves with $uv \in E(G) \Leftrightarrow uv$ covered by $u'v' \in B$

Signed tree model

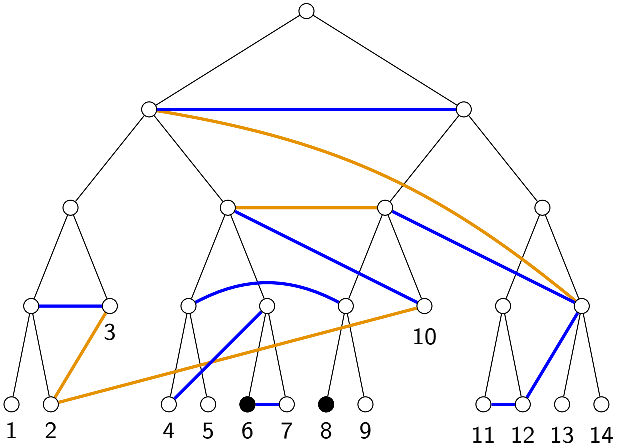
Tree T + transversal anti-edges A + transversal edges B
No crossing pairs in $A \cup B$



graph G at the leaves with $uv \in E(G) \Leftrightarrow uv$ covered by $u'v' \in B$

Signed tree model

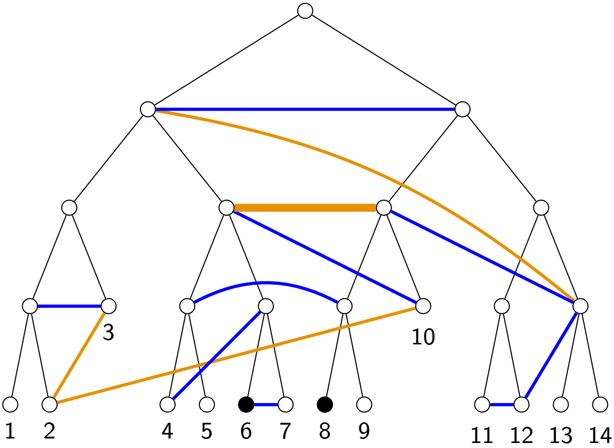
Tree T + transversal anti-edges A + transversal edges B
No crossing pairs in $A \cup B$



graph G at the leaves with $uv \in E(G) \Leftrightarrow uv$ covered by $u'v' \in B$

Signed tree model

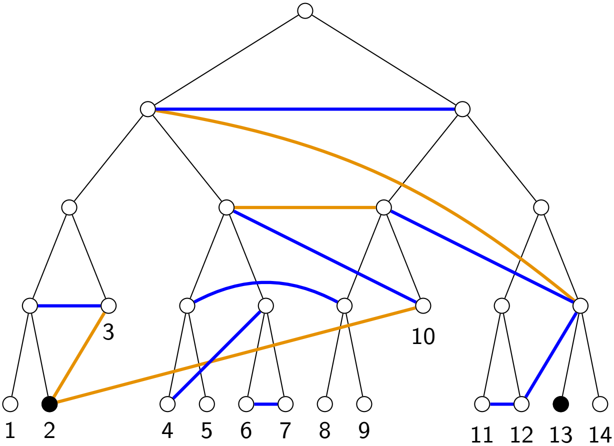
Tree T + transversal anti-edges A + transversal edges B
No crossing pairs in $A \cup B$



graph G at the leaves with $uv \in E(G) \Leftrightarrow uv$ covered by $u'v' \in B$

Signed tree model

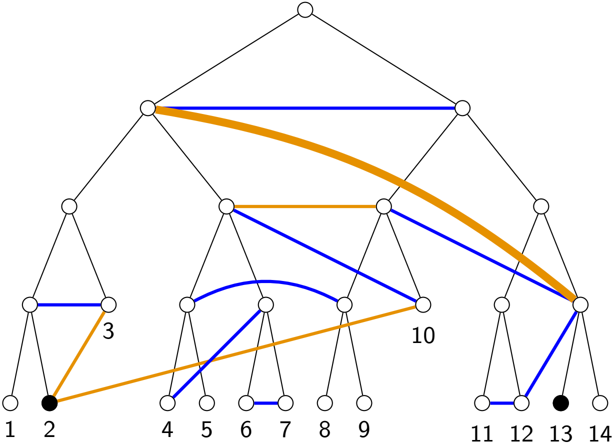
Tree T + transversal anti-edges A + transversal edges B
No crossing pairs in $A \cup B$



graph G at the leaves with $uv \in E(G) \Leftrightarrow uv$ covered by $u'v' \in B$

Signed tree model

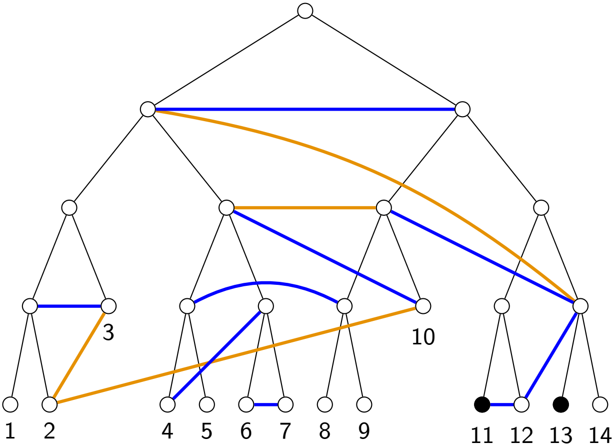
Tree T + transversal anti-edges A + transversal edges B
No crossing pairs in $A \cup B$



graph G at the leaves with $uv \in E(G) \Leftrightarrow uv$ covered by $u'v' \in B$

Signed tree model

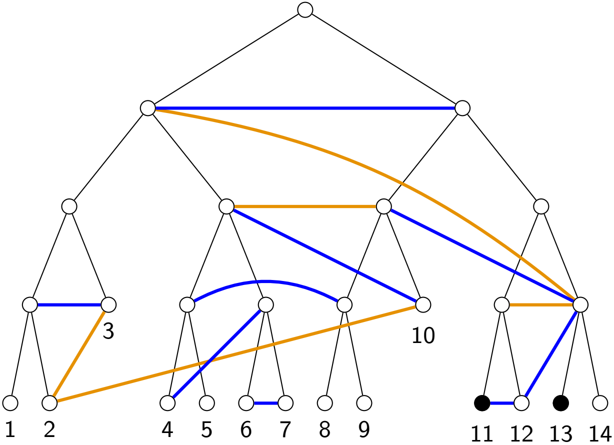
Tree T + transversal anti-edges A + transversal edges B
No crossing pairs in $A \cup B$



graph G at the leaves with $uv \in E(G) \Leftrightarrow uv$ covered by $u'v' \in B$

Signed tree model

Tree T + transversal anti-edges A + transversal edges B
No crossing pairs in $A \cup B$



graph G at the leaves with $uv \in E(G) \Leftrightarrow uv$ covered by $u'v' \in B$

Sparse signed tree models

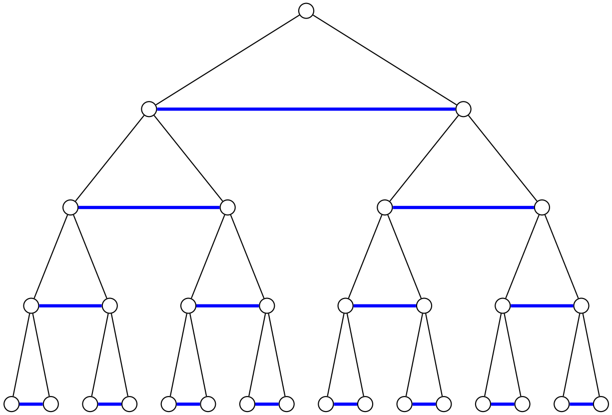
Every graph G has a tree model with $|E(G)|$ transversal edges

Sparse signed tree models

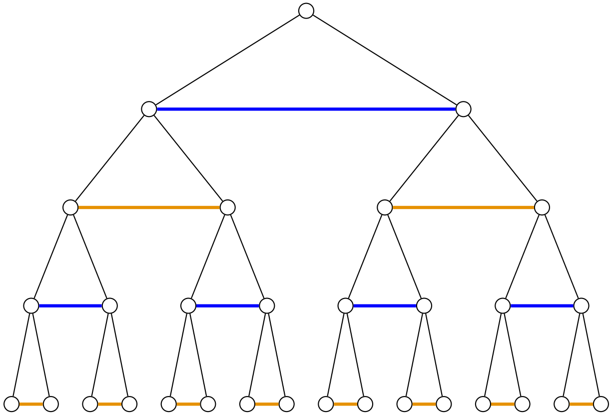
Every graph G has a tree model with $|E(G)|$ transversal edges

Sparse signed tree model: $\boldsymbol{O(n)}$ transversal pairs
Almost sparse signed tree model $\boldsymbol{n^{1+o(1)}}$ transversal pairs

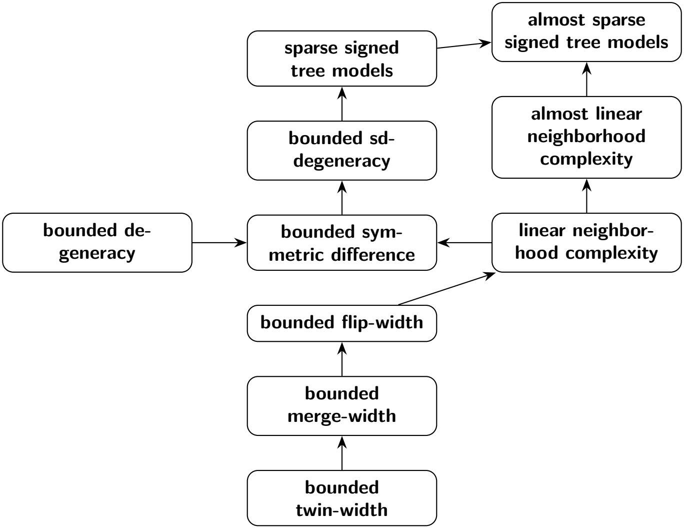
Examples of sparse signed tree models



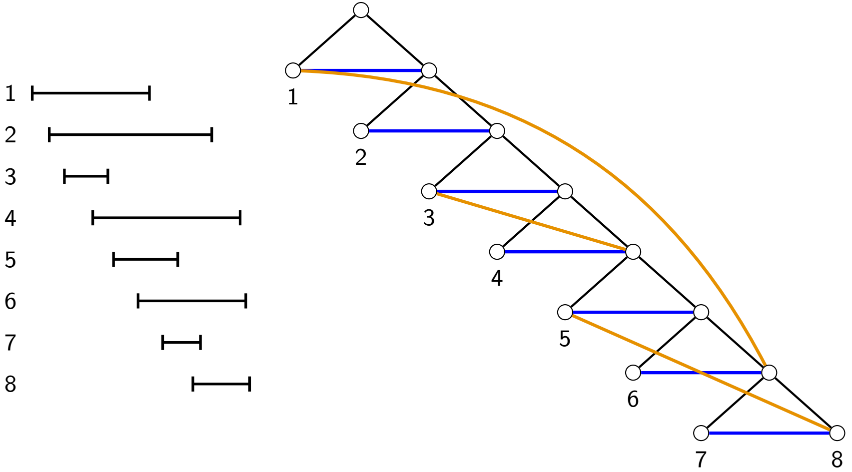
Examples of sparse signed tree models



Hasse diagram of our families of interest



Interval graphs admit sparse signed tree models



sort by left endpoints; each interval represents its *right* neighborhood by stopping adjacencices with an negative edge

Symmetric difference and sd-degeneracy

Dense counterparts of degeneracy based on

$$\mathsf{sd}_G(u, v) := |N_G(u) \setminus N_G[v]| + |N_G(v) \setminus N_G[u]|$$

Symmetric difference and sd-degeneracy

Dense counterparts of degeneracy based on

$$\mathrm{sd}_G(u,v) := |N_G(u) \setminus N_G[v]| + |N_G(v) \setminus N_G[u]|$$

Symmetric difference:

$$\mathrm{sd}(G) := \max_{H \text{ induced subgraph of } G} \ \min_{u \neq v \in V(H)} \mathrm{sd}_H(u,v)$$

Symmetric difference and sd-degeneracy

Dense counterparts of degeneracy based on

$$\mathrm{sd}_G(u,v) := |N_G(u) \setminus N_G[v]| + |N_G(v) \setminus N_G[u]|$$

Symmetric difference:

$$\mathrm{sd}(G) := \max_{H \text{ induced subgraph of } G} \; \min_{u \neq v \in V(H)} \mathrm{sd}_H(u,v)$$

Sd-degeneracy:

$$\mathrm{sdd}(G) := \min_{v_1,\dots,v_n \text{ ordering of } V(G)} \; \max_{i<n} \; \min_{j>i} \; \mathrm{sd}_{G-\{v_1,\dots,v_{i-1}\}}(v_i, v_j)$$

Symmetric difference and sd-degeneracy

Dense counterparts of degeneracy based on

$$\mathrm{sd}_G(u,v) := |N_G(u) \setminus N_G[v]| + |N_G(v) \setminus N_G[u]|$$

Symmetric difference:

$$\mathrm{sd}(G) := \max_{H \text{ induced subgraph of } G} \; \min_{u \neq v \in V(H)} \mathrm{sd}_H(u,v)$$

Sd-degeneracy:

$$\mathrm{sdd}(G) := \min_{v_1,\dots,v_n \text{ ordering of } V(G)} \; \max_{i<n} \; \min_{j>i} \; \mathrm{sd}_{G-\{v_1,\dots,v_{i-1}\}}(v_i, v_j)$$

Sd-degeneracy sequence of width d: $(u_1, v_1), \dots, (u_{n-1}, v_{n-1})$ s.t.

$$\mathrm{sd}_{G-\{u_1,u_2,\dots,u_{i-1}\}}(u_i, v_i) \leqslant d,$$

and $V(G) = \{u_1, u_2, \dots, u_{n-1}, v_{n-1}\}$

Computing symmetric difference and sd-degeneracy

DEGENERACY is easy whereas:

- SYMMETRIC DIFFERENCE is co-NP-complete
- SD-DEGENERACY is NP-complete

Computing symmetric difference and sd-degeneracy

DEGENERACY is easy whereas:

- SYMMETRIC DIFFERENCE is co-NP-complete
- SD-DEGENERACY is NP-complete

But computing an sd-degeneracy sequence of width $\mathrm{sd}(G)$ is easy

Linear neighborhood complexity to sd-degeneracy sequence

Theorem

Let $\mathcal{C}$ be a class of linear neighborhood complexity. One can compute an sd-degeneracy sequence of width $O(\log n)$ of an n-vertex m-edge graph G of $\mathcal{C}$ in time $O(m \log n)$.

Linear neighborhood complexity to sd-degeneracy sequence

Theorem

Let $\mathcal{C}$ be a class of linear neighborhood complexity. One can compute an sd-degeneracy sequence of width $O(\log n)$ of an n-vertex m-edge graph G of $\mathcal{C}$ in time $O(m \log n)$.

- Sample a linear-sized set X
- Partition by neighborhood classes w.r.t. X
- Find many pairs with $\text{sd} = O(\log n)$ within a same part

Sd-degeneracy sequence to signed tree model

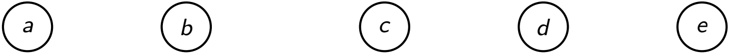
Lemma

There is an $O(dn + m)$-time algorithm that, given an sd-degeneracy sequence of width d of an n-vertex m-edge graph G, outputs a signed tree model of G with at most $O(dn)$ transversal pairs.

Sd-degeneracy sequence to signed tree model

Lemma

There is an $O(dn + m)$-time algorithm that, given an sd-degeneracy sequence of width d of an n-vertex m-edge graph G, outputs a signed tree model of G with at most $O(dn)$ transversal pairs.

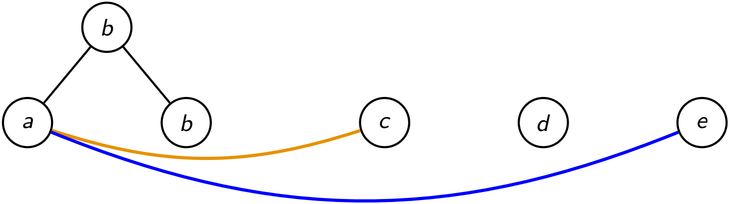


sd-degeneracy sequence: (a,b), (c,d), (d,e), (b,e)

Sd-degeneracy sequence to signed tree model

Lemma

There is an $O(dn + m)$-time algorithm that, given an sd-degeneracy sequence of width d of an n-vertex m-edge graph G, outputs a signed tree model of G with at most $O(dn)$ transversal pairs.

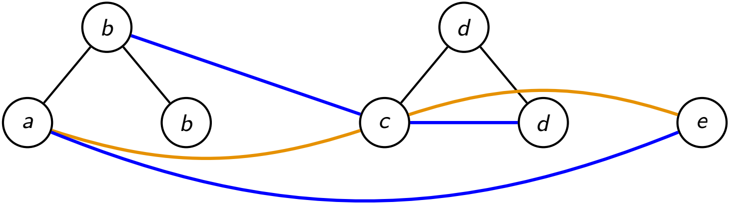


sd-degeneracy sequence: **(a,b)**, (c,d), (d,e), (b,e)

Sd-degeneracy sequence to signed tree model

Lemma

There is an $O(dn+m)$-time algorithm that, given an sd-degeneracy sequence of width d of an n-vertex m-edge graph G, outputs a signed tree model of G with at most $O(dn)$ transversal pairs.

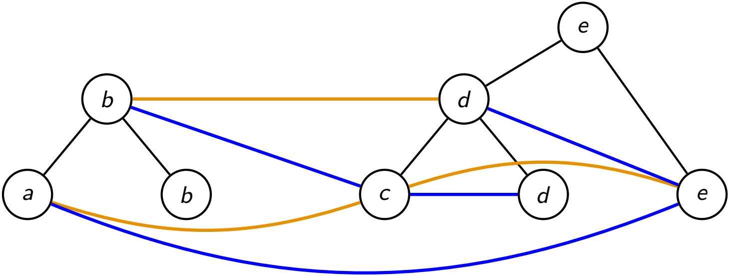


sd-degeneracy sequence: (a,b), **(c,d)**, (d,e), (b,e)

Sd-degeneracy sequence to signed tree model

Lemma

There is an $O(dn + m)$-time algorithm that, given an sd-degeneracy sequence of width d of an n-vertex m-edge graph G, outputs a signed tree model of G with at most $O(dn)$ transversal pairs.



sd-degeneracy sequence: (a,b), (c,d), **(d,e)**, (b,e)

Shortest Path for graphs with a signed tree model

Our goal now:

Theorem

There is an $O(p \log n)$-time algorithm that, given a signed tree model with p transversal pairs of an n-vertex graph G and $v \in V(G)$, outputs a shortest-path tree of G rooted at v.

Digraph compression

of a graph G: edge-weighted digraph D such that

- $V(G) \subseteq V(D)$, and
- for every $u, v \in V(G)$, $\text{dist}_D(u, v) = \text{dist}_G(u, v)$.

Size: $|E(D)|$

Digraph compression

of a graph G: edge-weighted digraph D such that

- $V(G) \subseteq V(D)$, and
- for every $u, v \in V(G)$, $\text{dist}_D(u, v) = \text{dist}_G(u, v)$.

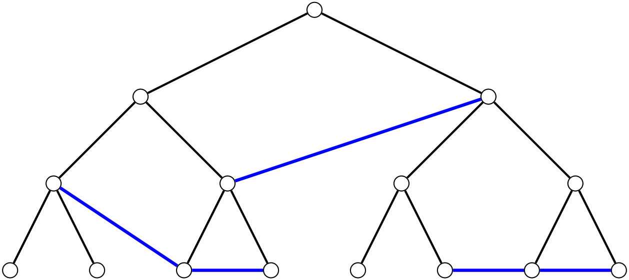
Size: $|E(D)|$

Observation

Given a digraph compression of G of size m, one can solve Single-Source Shortest Path *in G in time $O(m)$.*

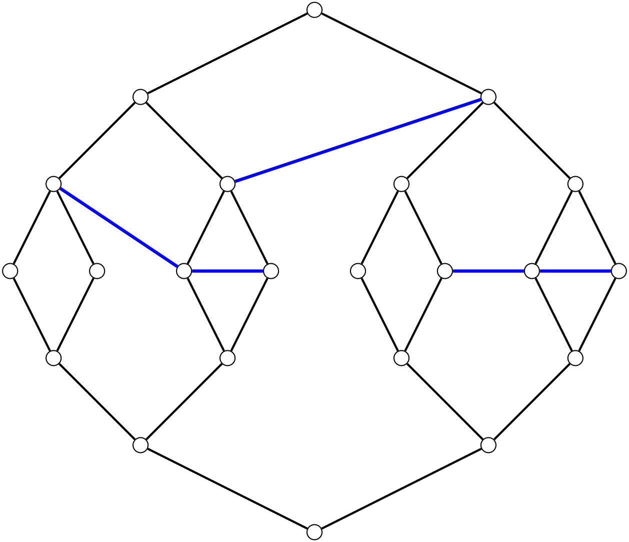
Digraph compression of a graph with a positive tree model

Bannach, Marwitz, Tantau '24



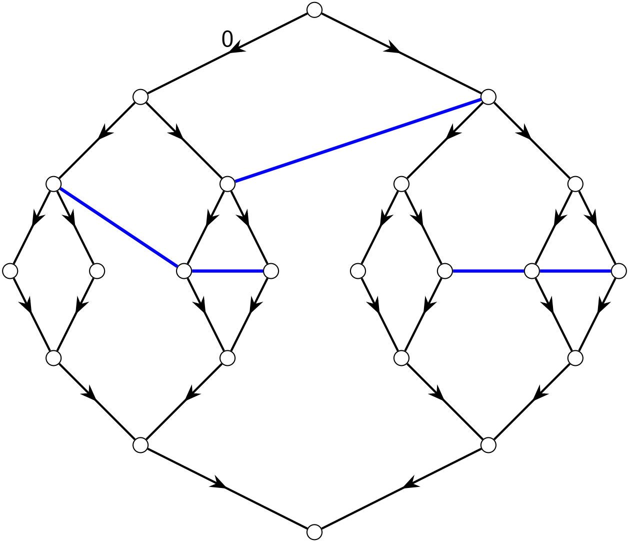
Digraph compression of a graph with a positive tree model

Bannach, Marwitz, Tantau '24



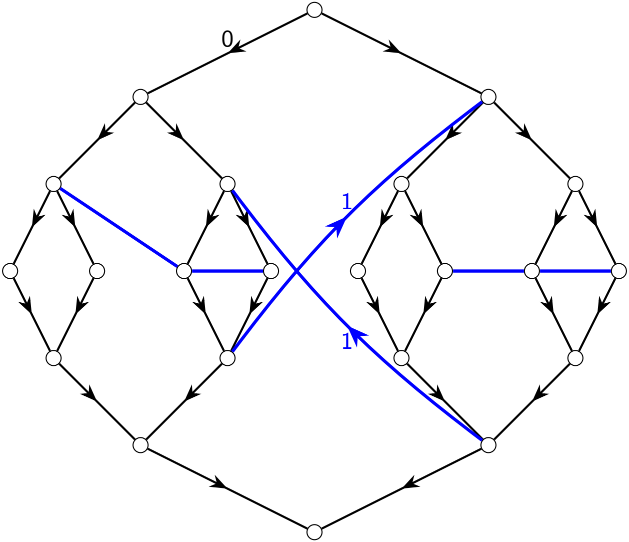
Digraph compression of a graph with a positive tree model

Bannach, Marwitz, Tantau '24



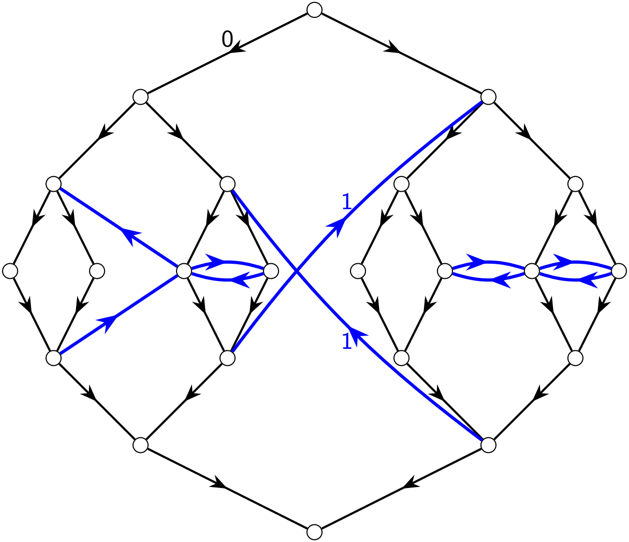
Digraph compression of a graph with a positive tree model

Bannach, Marwitz, Tantau ’24



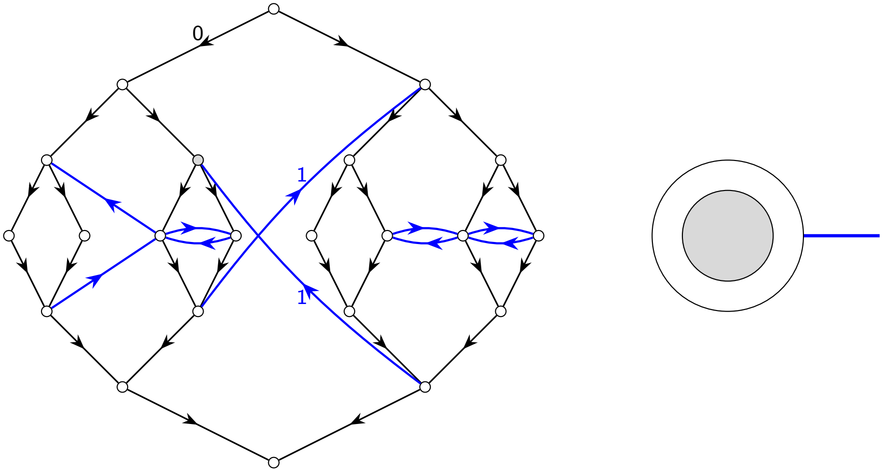
Digraph compression of a graph with a positive tree model

Bannach, Marwitz, Tantau '24



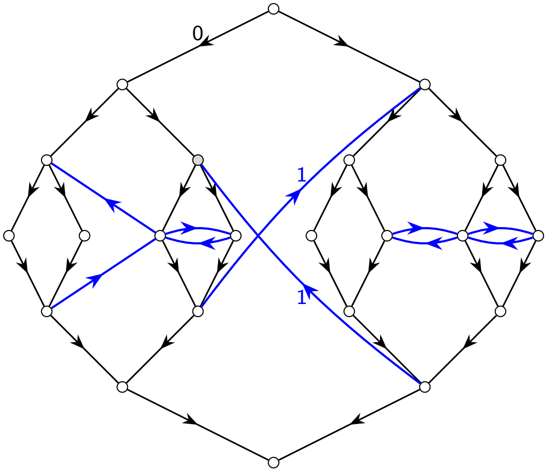
Digraph compression of a graph with a positive tree model

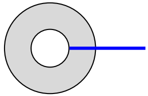
Bannach, Marwitz, Tantau '24



Digraph compression of a graph with a positive tree model

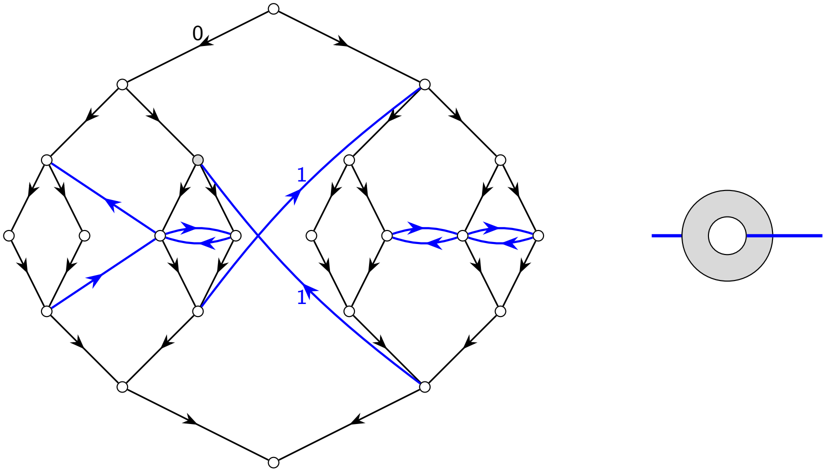
Bannach, Marwitz, Tantau '24





Digraph compression of a graph with a positive tree model

Bannach, Marwitz, Tantau '24



Interval Biclique Partition (IBP)

of a graph G: pair $(<, \mathcal{B})$ where

- $<$ is a total order of $V(G)$, and
- $\mathcal{B}$ is a set of bicliques every side of which is an $<$-interval.

Interval Biclique Partition (IBP)

of a graph G: pair $(<, \mathcal{B})$ where

- $<$ is a total order of $V(G)$, and
- $\mathcal{B}$ is a set of bicliques every side of which is an $<$-interval.

Theorem (B., Geniet, Kim, Thomassé, Watrigant '21)

Given $(<, \mathcal{B})$ an IBP of an n-vertex graph G, one can solve Single-Source Shortest Path *in G in time $O((n + |\mathcal{B}|) \log n)$.*

Interval Biclique Partition (IBP)

of a graph G: pair $(<, \mathcal{B})$ where

- $<$ is a total order of $V(G)$, and
- $\mathcal{B}$ is a set of bicliques every side of which is an $<$-interval.

Theorem (B., Geniet, Kim, Thomassé, Watrigant '21)

Given $(<, \mathcal{B})$ an IBP of an n-vertex graph G, one can solve Single-Source Shortest Path *in G in time $O((n + |\mathcal{B}|) \log n)$.*

Now improved to $O(n + |\mathcal{B}| \log^* n)$ via digraph compression

Positivizing a signed tree model

Theorem

There is an $O(p \log n)$-time algorithm that converts a signed tree model with p transversal pairs of an n-vertex graph G into any of:

1. *an interval biclique partition of G with $O(p)$ bicliques;*
2. *a digraph compression of size $O(p \log n)$ for G;*
3. *with additional $O(p \log^2 n)$ time, a tree model of G with $O(p \log^2 n)$ transversal edges.*

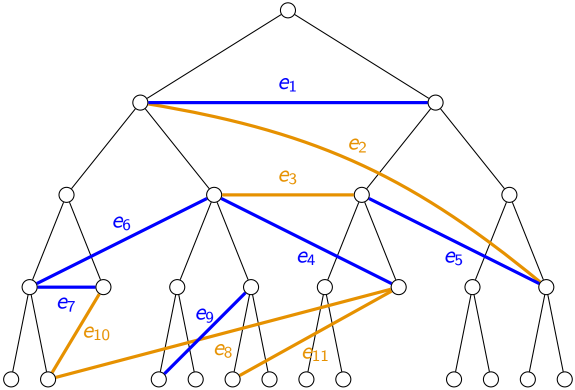
Positivizing a signed tree model

Theorem

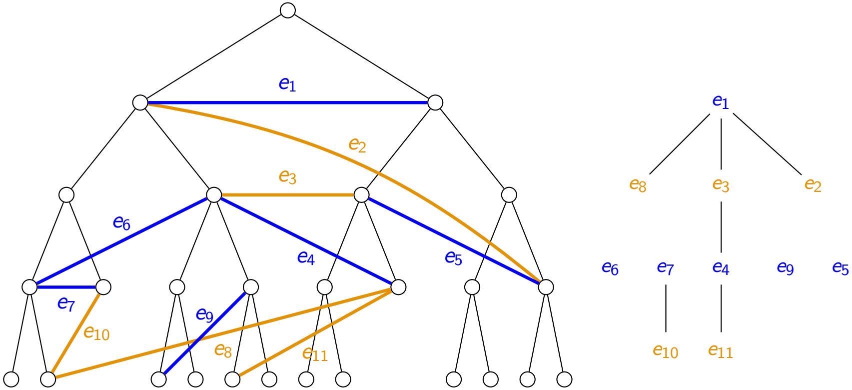
There is an $O(p \log n)$-time algorithm that converts a signed tree model with p transversal pairs of an n-vertex graph G into any of:

1. *an interval biclique partition of G with $O(p)$ bicliques;*
2. *a digraph compression of size $O(p \log^* n)$ for G;*
3. *with additional $O(p \log^2 n)$ time, a tree model of G with $O(p \log^2 n)$ transversal edges.*

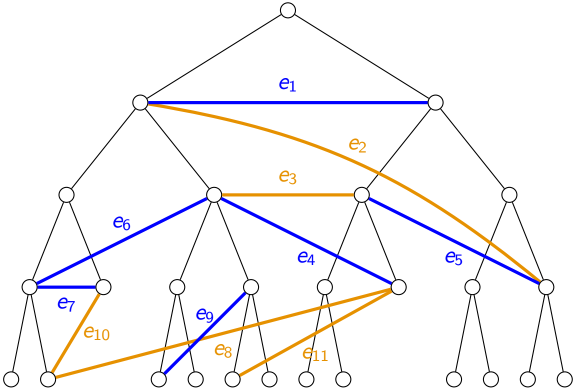
The cover relation on transversal pairs is a forest

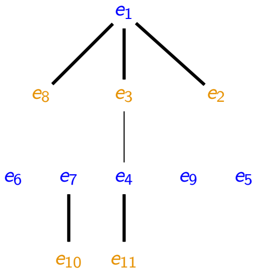


The cover relation on transversal pairs is a forest



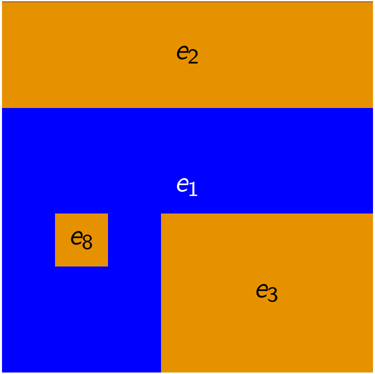
The cover relation on transversal pairs is a forest





parrallel complementation of bibliques within a biclique

Complementing n disjoint rectangles with $O(n)$ rectangles



Theorem (Rezende, Lee, Wu '89)

One can compute in $O(n \log n)$ time the complement of n axis-parallel ***disjoint*** *rectangles into $O(n)$ rectangles.*

Computing the cover forest

Theorem (Mortensen '06)

Given a point set P, a data structure can be computed in time $O(|P| \log |P|)$ that supports:

1. *adding or deleting a point in P in time $O(\log |P|)$, and*
2. *given a rectangle R, computing the points in $P \cap R$ in time $O(\log |P| + |P \cap R|)$.*

Computing the cover forest

Theorem (Mortensen '06)

Given a point set P, a data structure can be computed in time $O(|P| \log |P|)$ that supports:

1. *adding or deleting a point in P in time $O(\log |P|)$, and*
2. *given a rectangle R, computing the points in $P \cap R$ in time $O(\log |P| + |P \cap R|)$.*

- Sort the bicliques/rectangles $R_1, \ldots, R_h$ by increasing area
- Pick $p_i \in R_i$, and start with $P = \emptyset$
- For i from 1 to h, R_i is the parent of R_j for every $p_j \in P \cap R_i$
- Remove those contained p_j's from P, and add p_i to P

Positivizing a signed tree model

Theorem

There is an $O(p \log n)$-time algorithm that converts a signed tree model with p transversal pairs of an n-vertex graph G into any of:

1. *an interval biclique partition of G with $O(p)$ bicliques;*
2. *a digraph compression of size $O(p \log^* n)$ for G;*
3. *with additional $O(p \log^2 n)$ time, a tree model of G with $O(p \log^2 n)$ transversal edges.*

Positivizing a signed tree model

Theorem

There is an $O(p \log n)$-time algorithm that converts a signed tree model with p transversal pairs of an n-vertex graph G into any of:

1. *an interval biclique partition of G with $O(p)$ bicliques;*
2. *a digraph compression of size $O(p \log^* n)$ for G;*
3. *with additional $O(p \log^2 n)$ time, a tree model of G with $O(p \log^2 n)$ transversal edges.*

Theorem

All-Pairs Shortest Path *can be solved in $O(n^2 \log^* n)$ time on any n-vertex graph represented by a sparse signed tree model.*

Matrix Multiplication

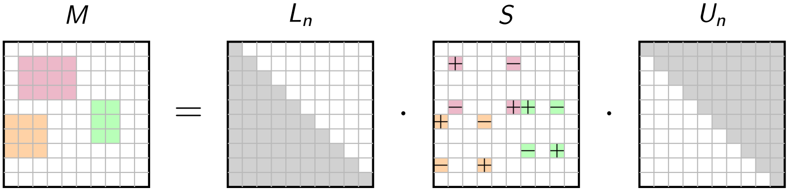
Theorem

Given $(<, \mathcal{B})$ *an IBP of an n-vertex graph G, one can multiply* $M := Adj_{<}(G)$ *with a vector of size n in time* $O(n + |\mathcal{B}|)$.

Matrix Multiplication

Theorem

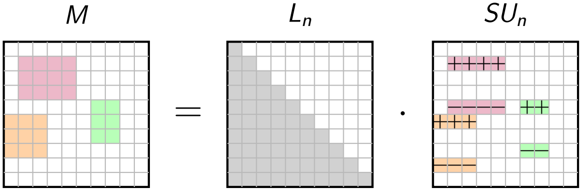
Given $(<, \mathcal{B})$ an IBP of an n-vertex graph G, one can multiply $M := Adj_<(G)$ with a vector of size n in time $O(n + |\mathcal{B}|)$.



Matrix Multiplication

Theorem

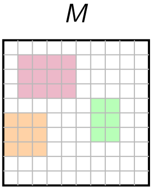
Given $(<, \mathcal{B})$ an IBP of an n-vertex graph G, one can multiply $M := Adj_<(G)$ with a vector of size n in time $O(n + |\mathcal{B}|)$.



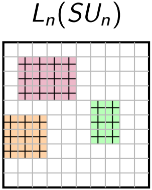
Matrix Multiplication

Theorem

Given $(<, \mathcal{B})$ *an IBP of an n-vertex graph G, one can multiply* $M := Adj_<(G)$ *with a vector of size n in time* $O(n + |\mathcal{B}|)$.



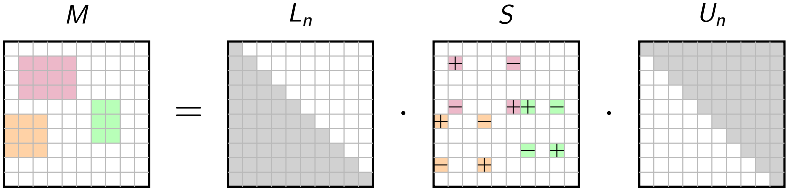
$=$



Matrix Multiplication

Theorem

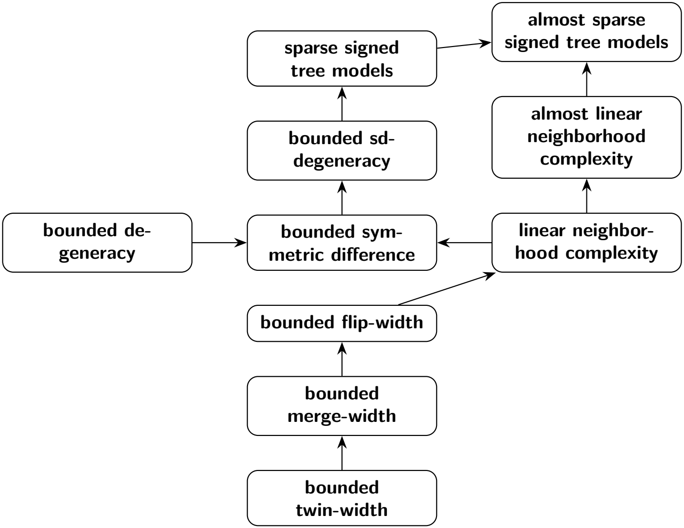
Given $(<, \mathcal{B})$ an IBP of an n-vertex graph G, one can multiply $M := Adj_<(G)$ with a vector of size n in time $O(n + |\mathcal{B}|)$.



Theorem

MN can be computed in $O(n^2)$ for any adjacency matrix M of an n-vertex graph with a sparse signed tree model and $n \times n$ matrix N.

Hasse diagram of our families of interest



Speed-ups in classes of linear neighborhood complexity

Let $\mathcal{C}$ be any class of linear neighborhood complexity

Theorem

All-Pairs Shortest Path can be solved in time $O(n^2 \log n \log^* n)$ in n-vertex graphs of $\mathcal{C}$.

Theorem

MN can be computed in time $O(n^2 \log n)$ for M the adjacency matrix of an n-vertex graph of $\mathcal{C}$ and N any $n \times n$ matrix.

Speed-ups in classes of linear neighborhood complexity

Let $\mathcal{C}$ be any class of linear neighborhood complexity

Conjecture

All-Pairs Shortest Path *can be solved in time* $O(n^2)$ *in* n*-vertex graphs of* $\mathcal{C}$.

Conjecture

MN *can be computed in time* $O(n^2)$ *for* M *the adjacency matrix of an* n*-vertex graph of* $\mathcal{C}$ *and* N *any* $n \times n$ *matrix.*

Questions

- Is there a factorial class *without* sparse signed tree models?
- Near-quadratic algorithms in graphs of bounded symmetric difference?
- Improving the $O(n^2 \log\log n)$ computation of signed tree models with $O(n \log n)$ transversal pairs in classes of lin. nc?
- Other (algorithmic) uses of signed tree models?
  Dynamic algorithms?
- Comparing bounded merge-width and signed tree models whose tree-minors have bounded degeneracy (expansion)?

Questions

- Is there a factorial class *without* sparse signed tree models?
- Near-quadratic algorithms in graphs of bounded symmetric difference?
- Improving the $O(n^2 \log \log n)$ computation of signed tree models with $O(n \log n)$ transversal pairs in classes of lin. nc?
- Other (algorithmic) uses of signed tree models?
  Dynamic algorithms?
- Comparing bounded merge-width and signed tree models whose tree-minors have bounded degeneracy (expansion)?

Thank you for your attention!