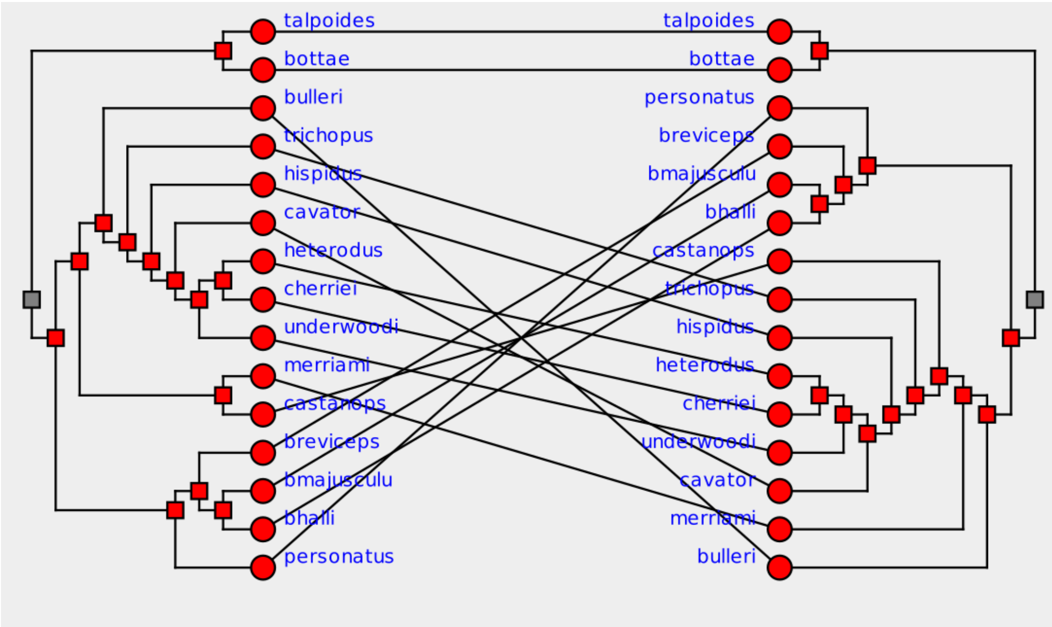
Block Crossings in One-Sided Tanglegrams

Alexander Dobler, Martin Nöllenburg

July 31, 2023 · WADS 2023

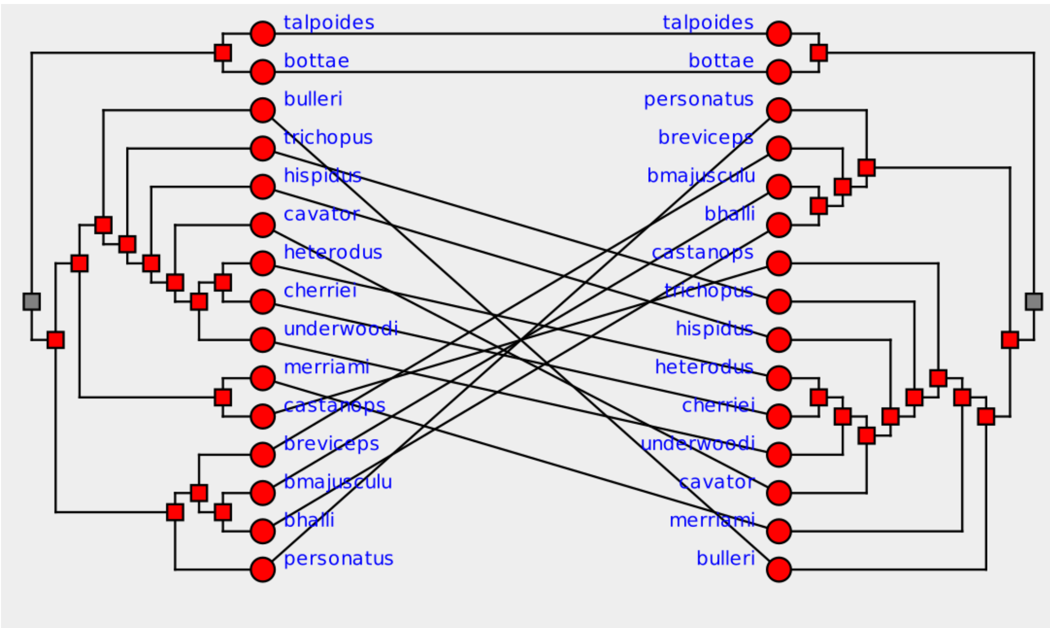
ALGORITHMS AND
COMPLEXITY GROUP

Tanglegrams



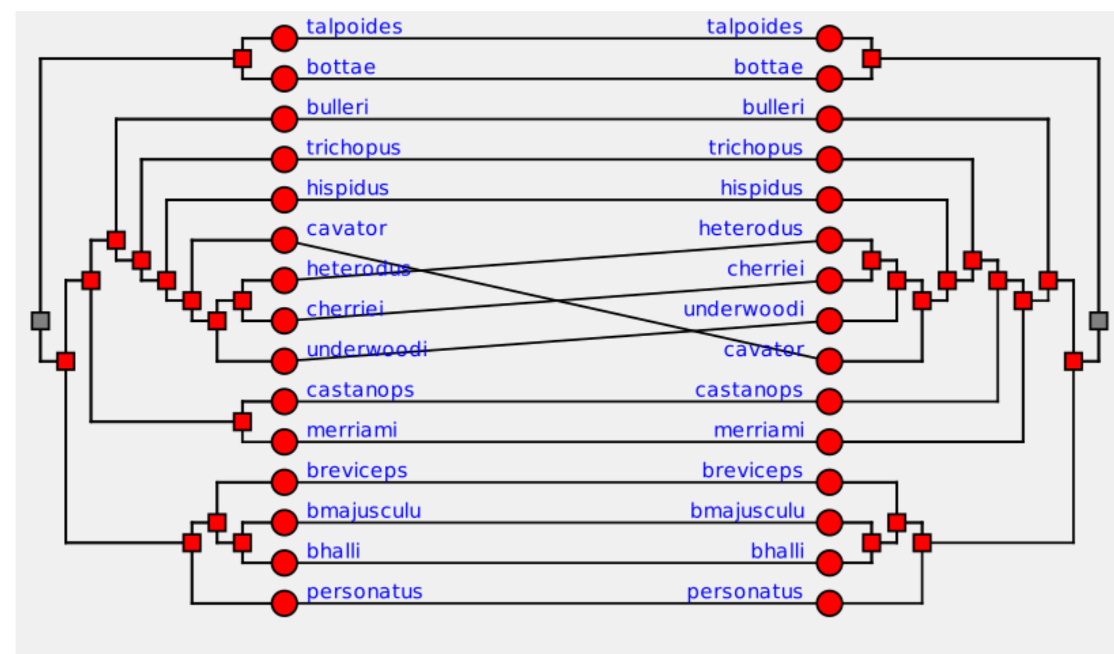
Comparison of species trees of **same leaf set**

Tanglegrams



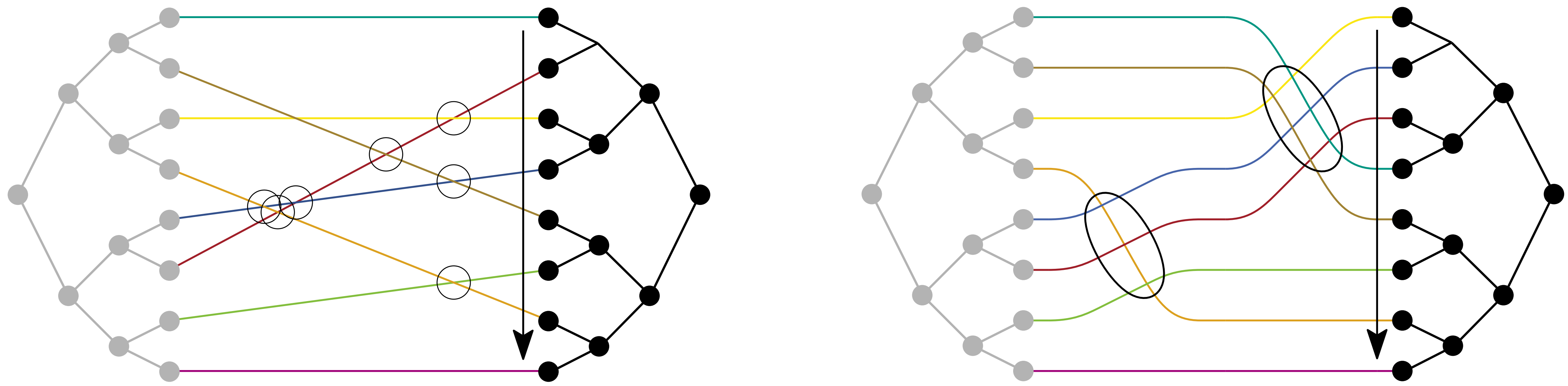
Comparison of species trees of **same leaf set**

Known **combinatorial problem**: reorder leaves of one/both trees to **minimize crossings**



Edge Crossings vs. Block Crossings

Same instance, left leaf order fixed

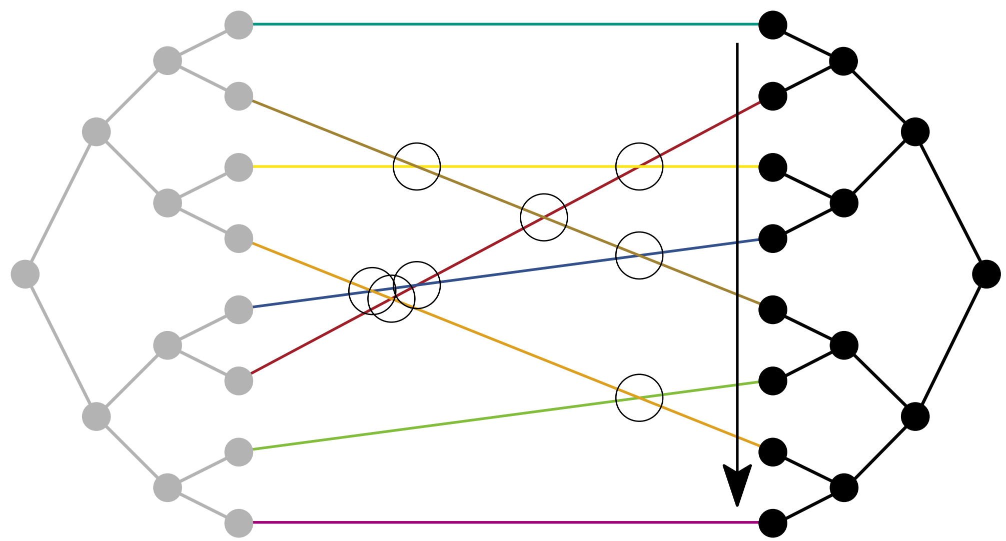


Optimal: 8 edge crossings

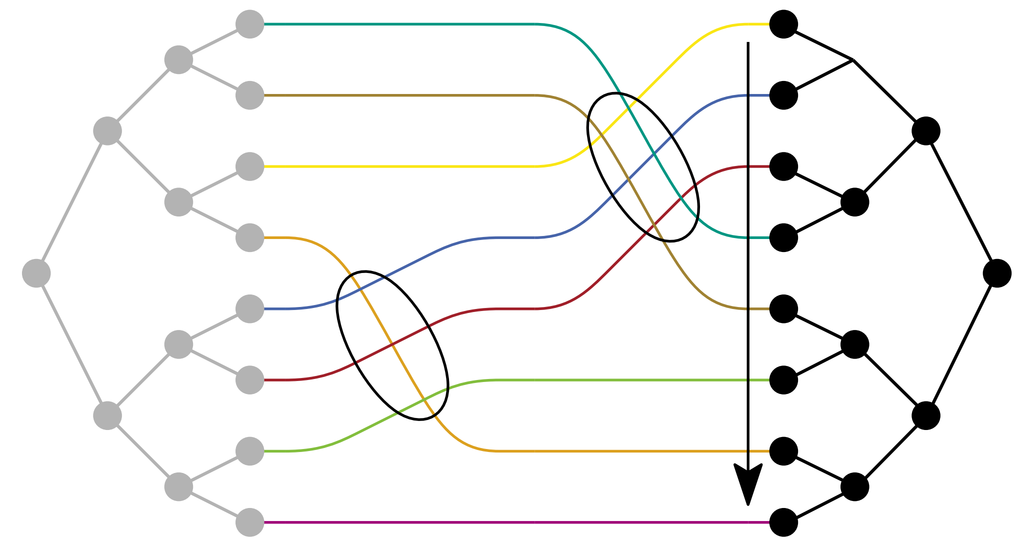
Optimal: 2 block crossings
(but 9 edge crossings)

Edge Crossings vs. Block Crossings

Same instance, left leaf order fixed



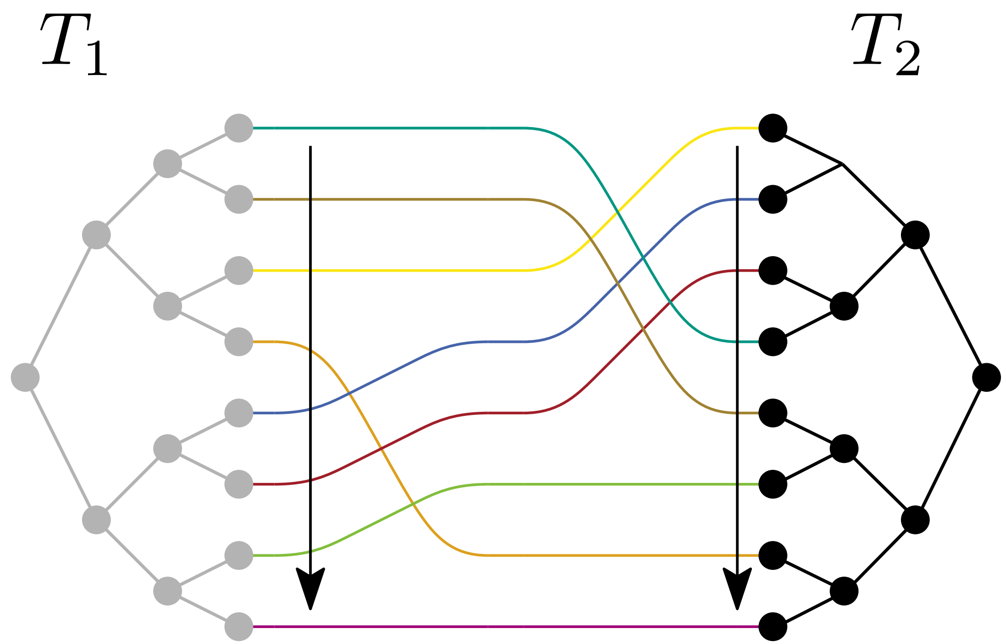
Optimal: 8 edge crossings



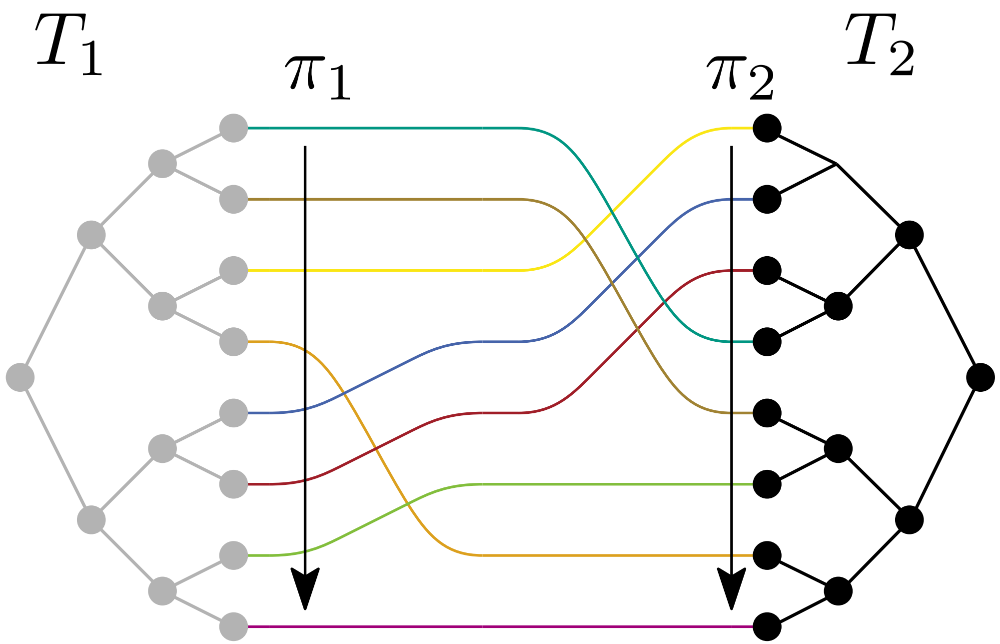
Optimal: 2 block crossings (but 9 edge crossings)

Reduces **visual clutter**, but edges **non-straight-line**

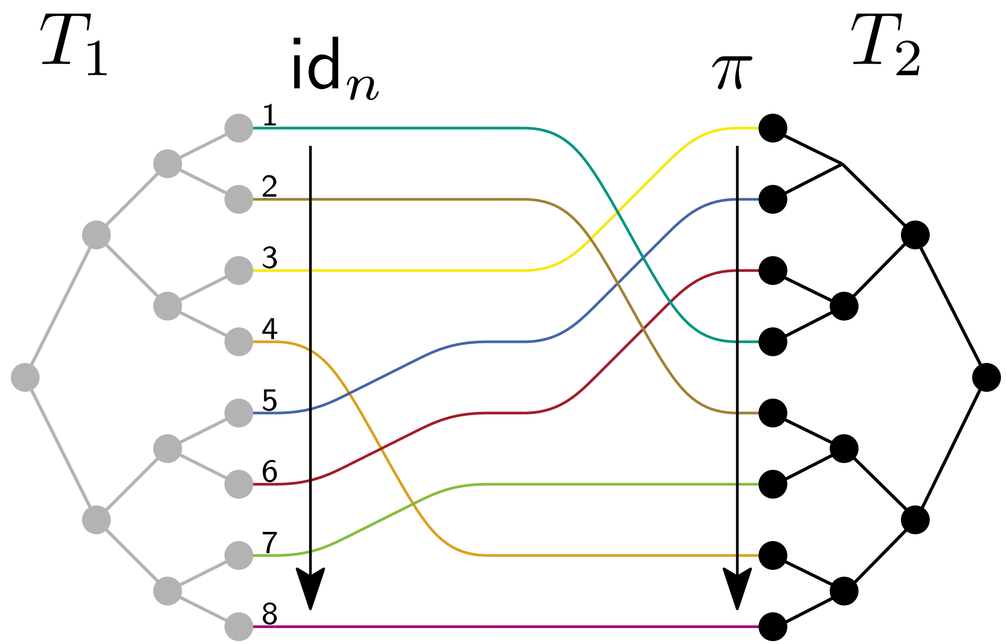
One-Sided Block Crossing Minimization Problem



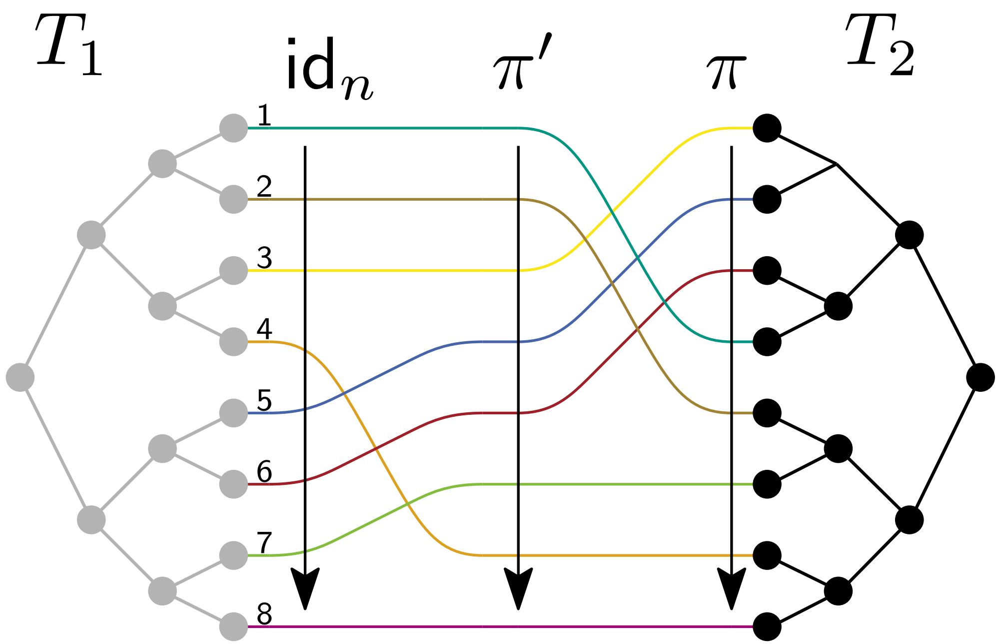
One-Sided Block Crossing Minimization Problem



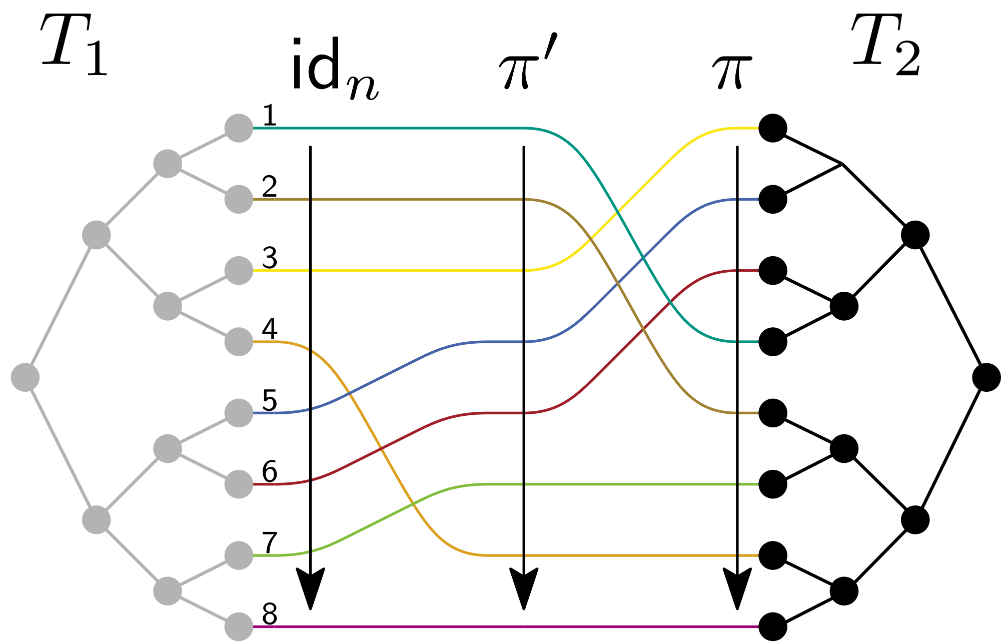
One-Sided Block Crossing Minimization Problem



One-Sided Block Crossing Minimization Problem

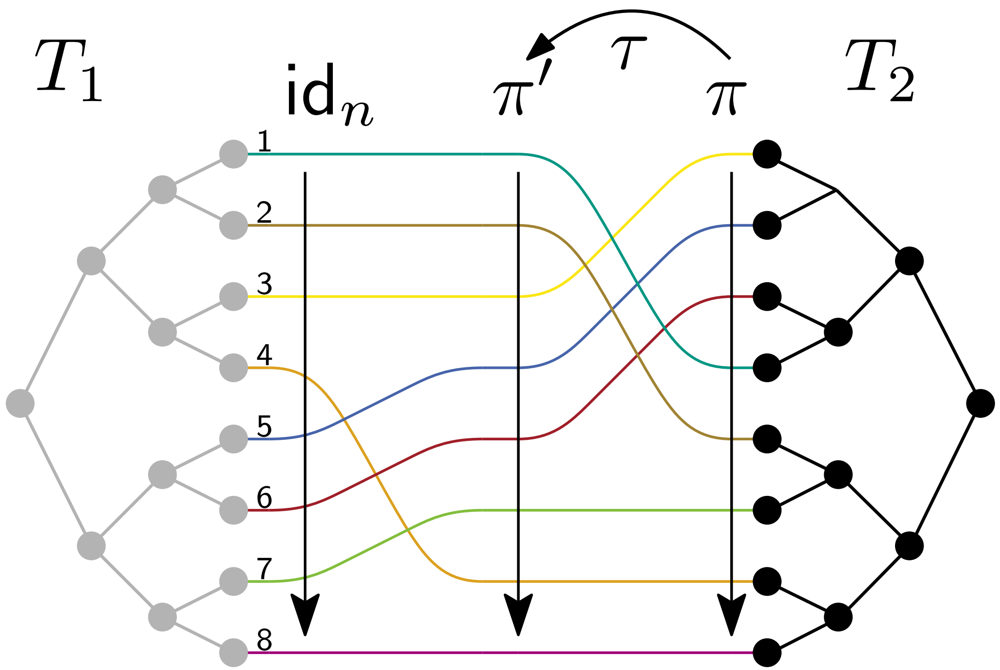


One-Sided Block Crossing Minimization Problem



Between π and π' two bundles of edges exchange positions

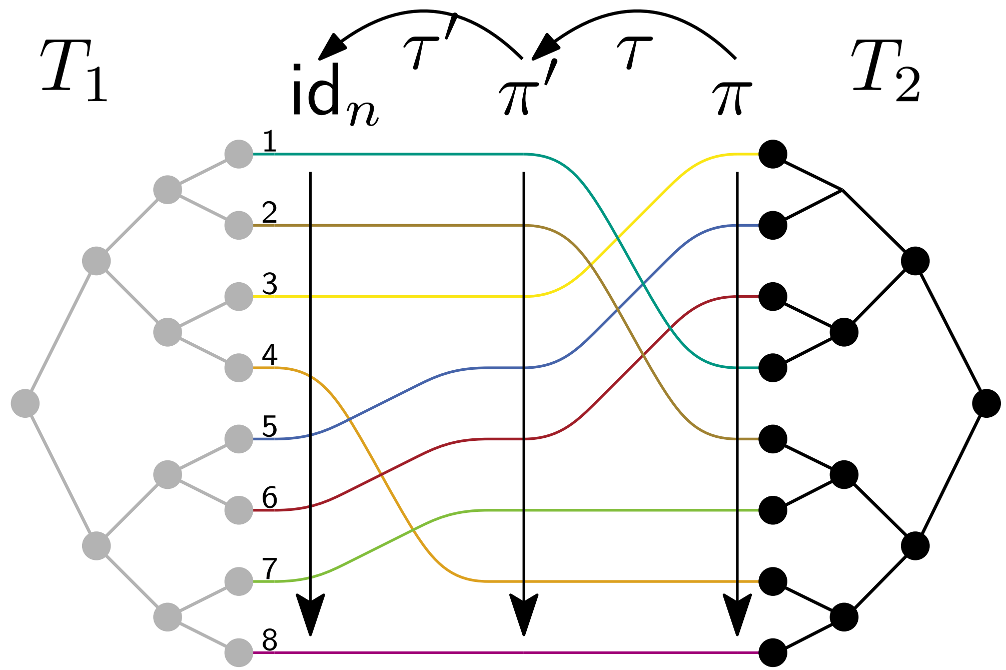
One-Sided Block Crossing Minimization Problem



Between π and π' two bundles of edges exchange positions

$\pi' = \pi \circ \tau$ for **transposition** τ

One-Sided Block Crossing Minimization Problem



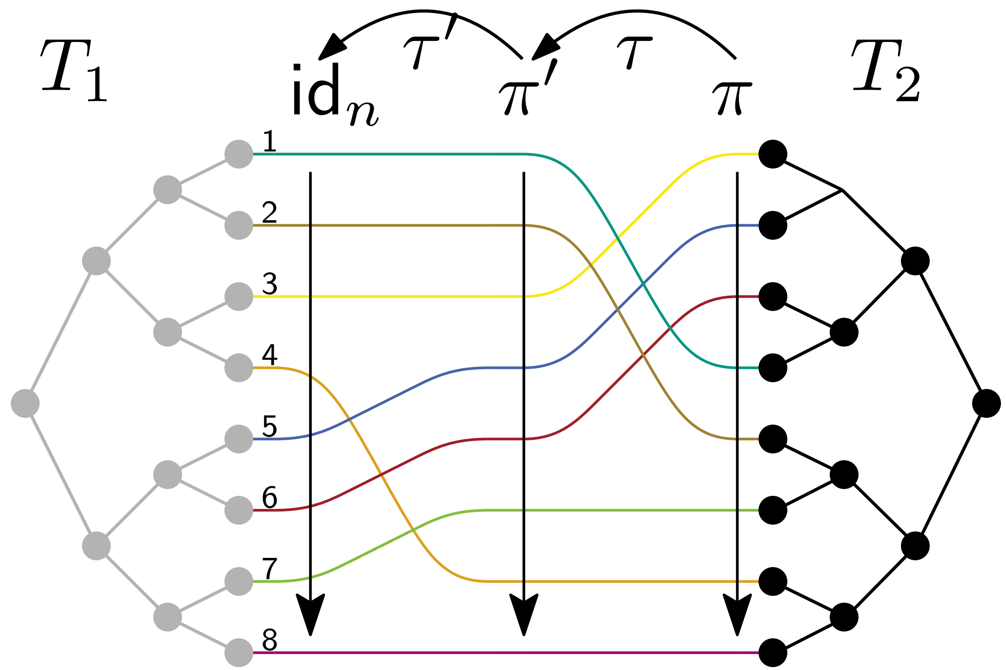
Between π and π' two bundles of edges exchange positions

$\pi' = \pi \circ \tau$ for **transposition** τ

$\mathrm{id}_n = \pi \circ \tau \circ \tau'$ for **transpositions** τ, τ'

Transposition = Block Crossing

One-Sided Block Crossing Minimization Problem



Between π and π' two bundles of edges exchange positions

$\pi' = \pi \circ \tau$ for **transposition** τ

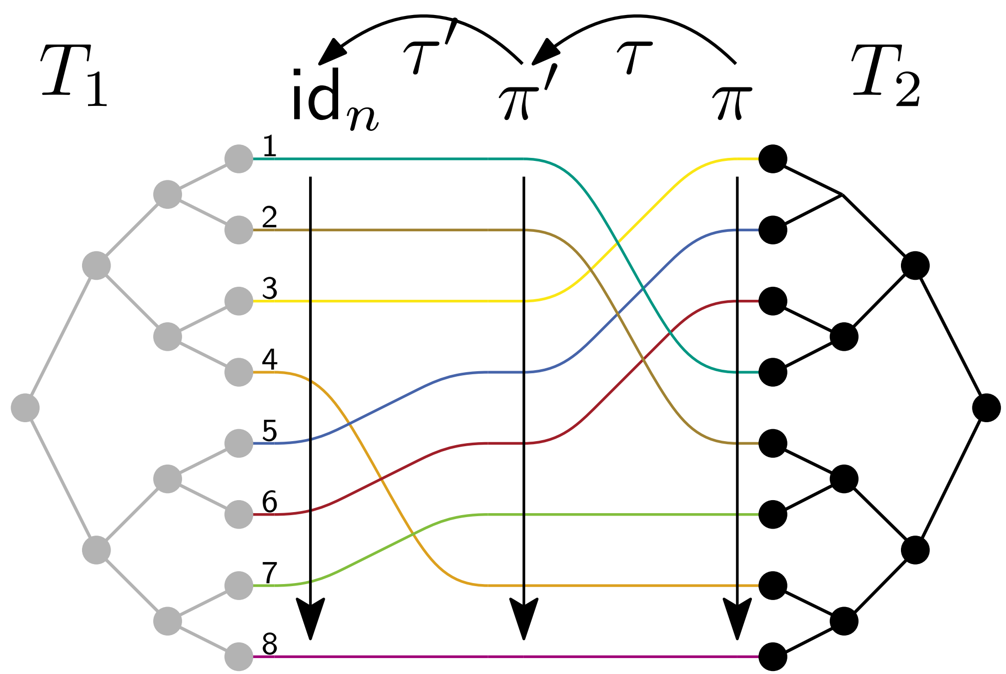
$\mathrm{id}_n = \pi \circ \tau \circ \tau'$ for **transpositions** τ, τ'

Transposition = Block Crossing

A **transposition** $\tau = \tau(i, j, k) \in \Pi_n$ with $1 \leq i < j < k \leq n + 1$ is the permutation

$$(1, \ldots, i-1, j, \ldots, k-1, i, \ldots, j-1, k, \ldots, n).$$

One-Sided Block Crossing Minimization Problem



Between π and π' two bundles of edges exchange positions

$\pi' = \pi \circ \tau$ for **transposition** τ

$\text{id}_n = \pi \circ \tau \circ \tau'$ for **transpositions** τ, τ'

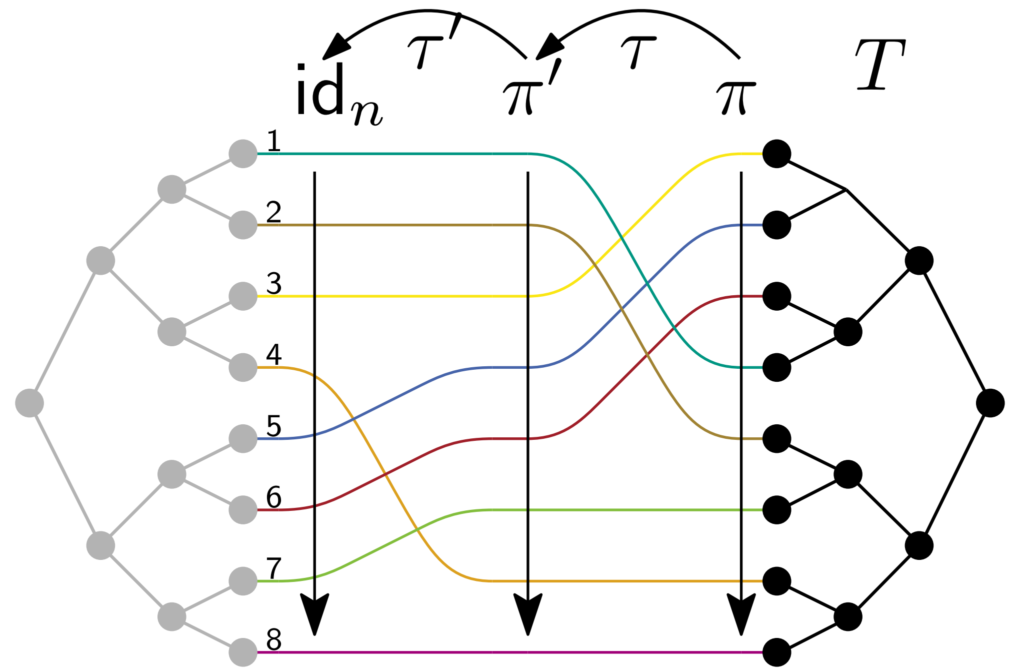
Transposition = Block Crossing

A **transposition** $\tau = \tau(i, j, k) \in \Pi_n$ with $1 \leq i < j < k \leq n + 1$ is the permutation

$$(1, \ldots, i-1, j, \ldots, k-1, i, \ldots, j-1, k, \ldots, n).$$

The **transposition distance** $d_t(\pi)$ of $\pi \in \Pi_n$ is the min. number $k \in \mathbb{N}$ s.t. there exist transpositions $\tau_1, \ldots, \tau_k$ with $\pi \circ \tau_1 \circ \tau_2, \cdots \circ \tau_k = \text{id}_n$

One-Sided Block Crossing Minimization Problem



One-Tree Block Crossing Minimization (OTBCM)

Instance: A rooted tree T with leaves$(T) = [n]$ and a positive integer k.

Question: Is there a permutation $\pi \in \Pi_n$ consistent with T such that there exist transpositions $\tau_1, \ldots, \tau_k$ with $\pi \circ \tau_1 \circ \tau_2 \circ \cdots \circ \tau_k = \mathsf{id}_n$?

Results

Complexity results for One-Tree Block Crossing Minimization:

Restr. on T	Block Crossing Min.	Crossing Min.
Complete Binary	**NP-complete** $\mathcal{O}(n^2)$ **2.25-approximation** FPT-algorithm in k	P [Dwyer and Schreiber 2004]
Binary	NP-complete $\mathcal{O}(n^3)$ **2.25-approximation** **FPT-algorithm in** k	P [Dwyer and Schreiber 2004]
Non-binary	NP-complete **Approx. does not extend**	NP-complete [Bulteau et al. 2022]

Results

Complexity results for One-Tree Block Crossing Minimization:

Restr. on T	Block Crossing Min.	Crossing Min.
Complete Binary	**NP-complete** $\mathcal{O}(n^2)$ **2.25-approximation** FPT-algorithm in k	P [Dwyer and Schreiber 2004]
Binary	NP-complete $\mathcal{O}(n^3)$ **2.25-approximation** **FPT-algorithm in** k	P [Dwyer and Schreiber 2004]
Non-binary	NP-complete **Approx. does not extend**	NP-complete [Bulteau et al. 2022]

Note: Crossing Min. also NP-hard if leaves of both trees can be permuted. Approximations and FPT-algorithms exist. [Buchin et al. 2012]

Results

Complexity results for One-Tree Block Crossing Minimization:

Restr. on T	Block Crossing Min.	Crossing Min.
Complete Binary	**NP-complete** $\mathcal{O}(n^2)$ **2.25-approximation** FPT-algorithm in k	P [Dwyer and Schreiber 2004]
Binary	NP-complete $\mathcal{O}(n^3)$ **2.25-approximation** **FPT-algorithm in k**	P [Dwyer and Schreiber 2004]
Non-binary	NP-complete **Approx. does not extend**	NP-complete [Bulteau et al. 2022]

Note: Crossing Min. also NP-hard if leaves of both trees can be permuted. Approximations and FPT-algorithms exist. [Buchin et al. 2012]

NP-completeness

NP-membership:

NP-completeness

NP-membership: Combinatorial problem $\square$

NP-completeness

NP-membership: Combinatorial problem □

NP-hardness:

NP-completeness

NP-membership: Combinatorial problem $\square$

NP-hardness:
Index $i \in \{0, 1, \ldots, n\}$ is a **breakpoint** in $\pi \in \Pi_n$ if

- $i = 0$ and $\pi_1 \neq 1$,
- $1 \leq i \leq n-1$ and $\pi_i + 1 \neq \pi_{i+1}$, or
- $i = n$ and $\pi_n \neq n$.

$$\pi = (2, 1, 3, 4, 6, 7, 5, 8)$$

NP-completeness

NP-membership: Combinatorial problem □

NP-hardness:

Index $i \in \{0, 1, \ldots, n\}$ is a **breakpoint** in $\pi \in \Pi_n$ if

- $i = 0$ and $\pi_1 \neq 1$,
- $1 \leq i \leq n - 1$ and $\pi_i + 1 \neq \pi_{i+1}$, or
- $i = n$ and $\pi_n \neq n$.

$$\pi = (2, 1, 3, 4, 6, 7, 5, 8) \qquad \text{bp}(\pi) = 5$$

bp(π) is the **number of breakpoints** of π

NP-completeness

NP-membership: Combinatorial problem □

NP-hardness:
Index $i \in \{0, 1, \ldots, n\}$ is a **breakpoint** in $\pi \in \Pi_n$ if

- $i = 0$ and $\pi_1 \neq 1$,
- $1 \leq i \leq n-1$ and $\pi_i + 1 \neq \pi_{i+1}$, or
- $i = n$ and $\pi_n \neq n$.

$$\pi = (2, 1, 3, 4, 6, 7, 5, 8) \qquad \text{bp}(\pi) = 5$$

bp(π) is the **number of breakpoints** of π

Lemma. (Bafna and Pevzner 1998) $d_t(\pi) \geq \lfloor \frac{\text{bp}(\pi)}{3} \rfloor$.

NP-completeness

NP-membership: Combinatorial problem □

NP-hardness:

Index $i \in \{0, 1, \ldots, n\}$ is a **breakpoint** in $\pi \in \Pi_n$ if

- $i = 0$ and $\pi_1 \neq 1$,
- $1 \leq i \leq n - 1$ and $\pi_i + 1 \neq \pi_{i+1}$, or
- $i = n$ and $\pi_n \neq n$.

$$\pi = (2, 1, 3, 4, 6, 7, 5, 8) \qquad \text{bp}(\pi) = 5$$

bp(π) is the **number of breakpoints** of π

Lemma. (Bafna and Pevzner 1998) $d_t(\pi) \geq \lfloor \frac{\text{bp}(\pi)}{3} \rfloor$.

Theorem. (Bulteau et al. 2012) It is NP-hard to decide whether $d_t(\pi) = \lfloor \frac{\text{bp}(\pi)}{3} \rfloor$.

NP-hardness cont.

Lemma. (Bafna and Pevzner 1998) $d_t(\pi) \geq \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$.

Theorem. (Bulteau et al. 2012) It is NP-hard to decide whether $d_t(\pi) = \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$.

Idea: Decide $d_t(\pi) = \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$ with our problem.

NP-hardness cont.

Lemma. (Bafna and Pevzner 1998) $d_t(\pi) \geq \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$.

Theorem. (Bulteau et al. 2012) It is NP-hard to decide whether $d_t(\pi) = \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$.

Idea: Decide $d_t(\pi) = \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$ with our problem.

Given π

$\pi = (2, 1, 3, 5, 4, 6)$

NP-hardness cont.

Lemma. (Bafna and Pevzner 1998) $d_t(\pi) \geq \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$.

Theorem. (Bulteau et al. 2012) It is NP-hard to decide whether $d_t(\pi) = \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$.

Idea: Decide $d_t(\pi) = \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$ with our problem.

Given π

$\pi = (2, 1, 3, 5, 4, 6)$

$\pi' = (3, 1, 2, 4, 6, 5, 7)$

Step 1: create π' of size $2^p - 1$ with $d_t(\pi) = d_t(\pi')$ and $\mathsf{bp}(\pi) = \mathsf{bp}(\pi')$

NP-hardness cont.

Lemma. (Bafna and Pevzner 1998) $d_t(\pi) \geq \lfloor \frac{\text{bp}(\pi)}{3} \rfloor$.

Theorem. (Bulteau et al. 2012) It is NP-hard to decide whether $d_t(\pi) = \lfloor \frac{\text{bp}(\pi)}{3} \rfloor$.

Idea: Decide $d_t(\pi) = \lfloor \frac{\text{bp}(\pi)}{3} \rfloor$ with our problem.

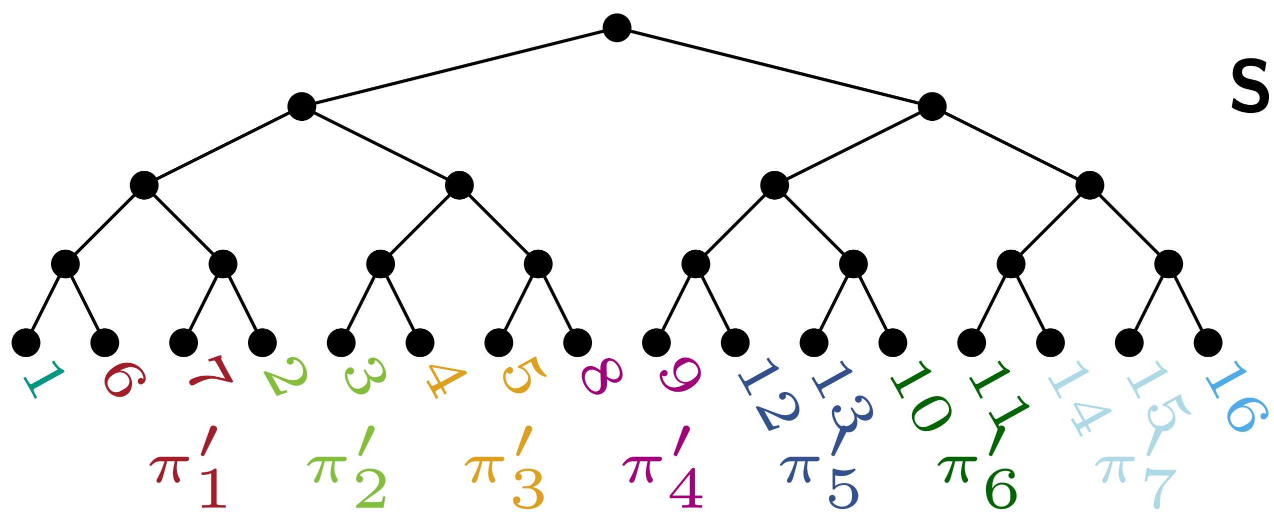
Given π

$\pi = (2, 1, 3, 5, 4, 6)$

$\pi' = (3, 1, 2, 4, 6, 5, 7)$

Step 1: create π' of size $2^p - 1$ with $d_t(\pi) = d_t(\pi')$ and $\text{bp}(\pi) = \text{bp}(\pi')$

Step 2: create tree T.



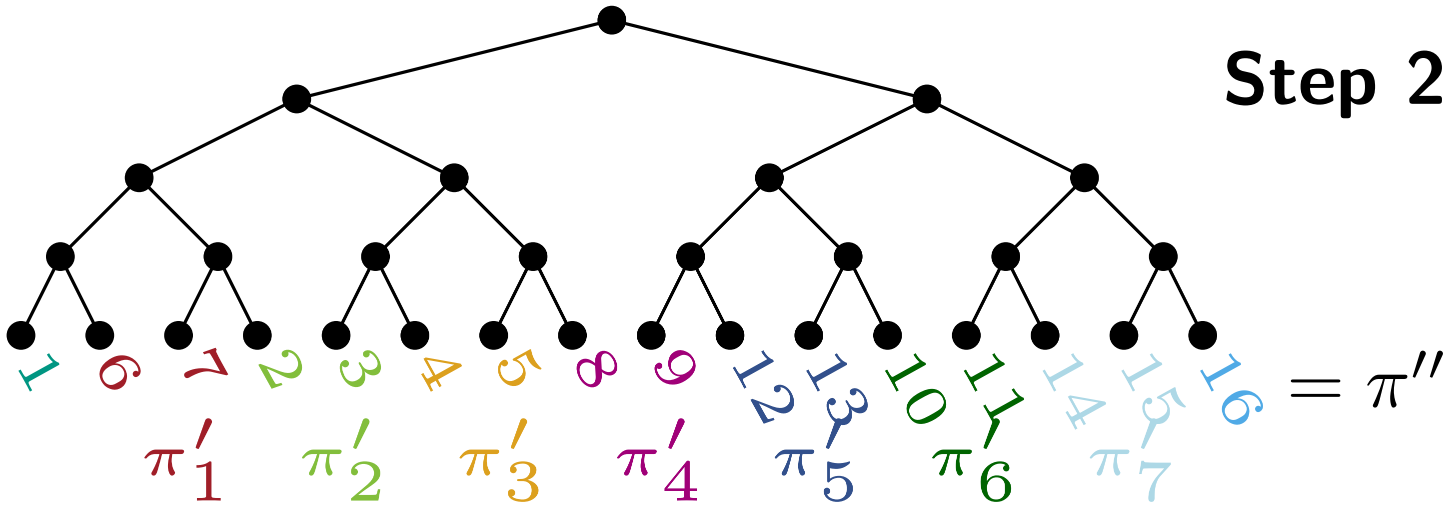
NP-hardness cont.

$\pi = (3, 1, 4, 6, 5, 7)$

$\pi' = (3, 1, 2, 4, 6, 5, 7)$

Step 1: create π' of size $2^p - 1$ with $d_t(\pi) = d_t(\pi')$ and $\text{bp}(\pi) = \text{bp}(\pi')$

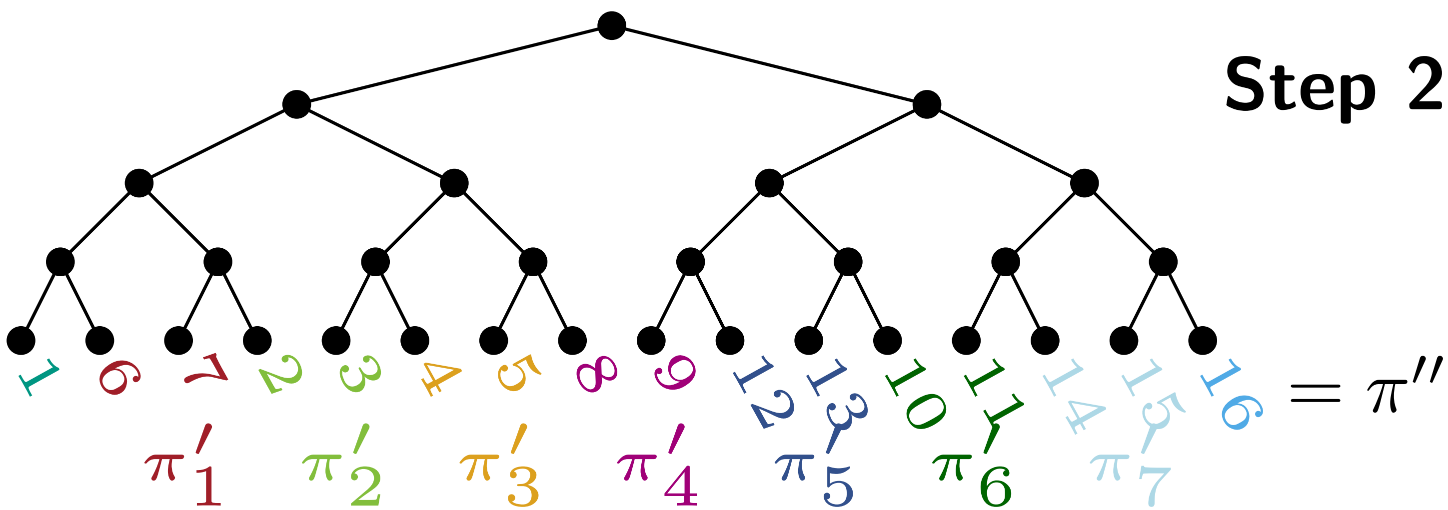
Step 2: create tree T.



NP-hardness cont.

$\pi = (3, 1, 4, 6, 5, 7)$

$\pi' = (3, 1, 2, 4, 6, 5, 7)$



Step 1: create π' of size $2^p - 1$ with $d_t(\pi) = d_t(\pi')$ and $\text{bp}(\pi) = \text{bp}(\pi')$

Step 2: create tree T.

Lemma. $d_t(\pi'') = d_t(\pi)$, $\text{bp}(\pi'') = \text{bp}(\pi)$.

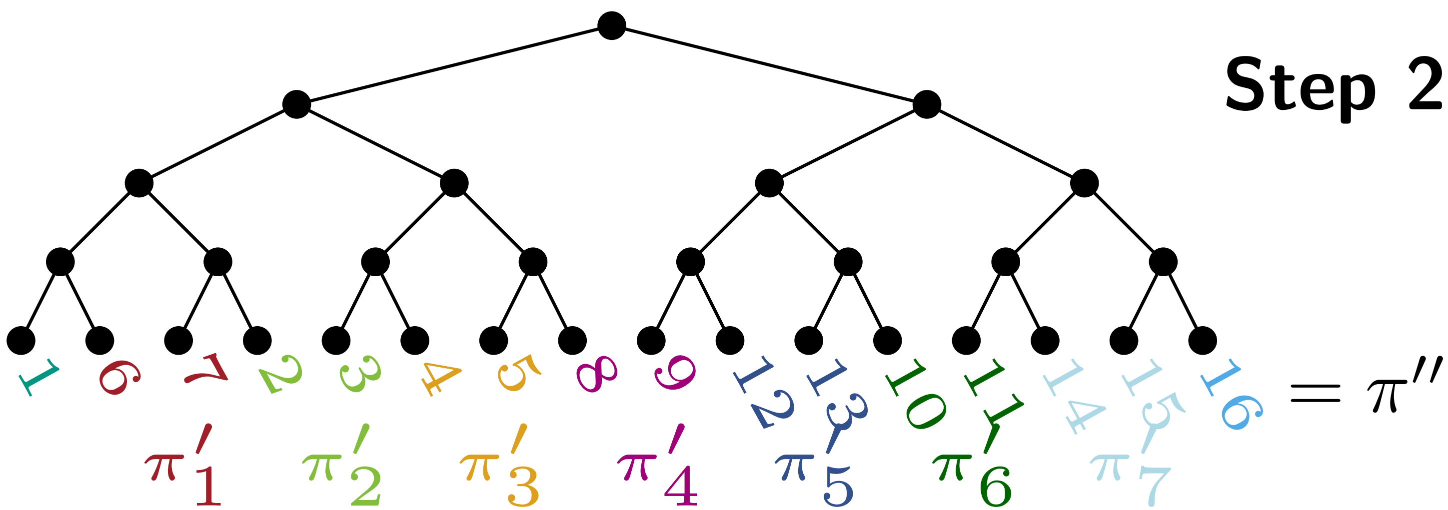
NP-hardness cont.

$\pi = (3, 1, 4, 6, 5, 7)$

$\pi' = (3, 1, 2, 4, 6, 5, 7)$

Step 1: create π' of size $2^p - 1$ with $d_t(\pi) = d_t(\pi')$ and $\text{bp}(\pi) = \text{bp}(\pi')$

Step 2: create tree T.



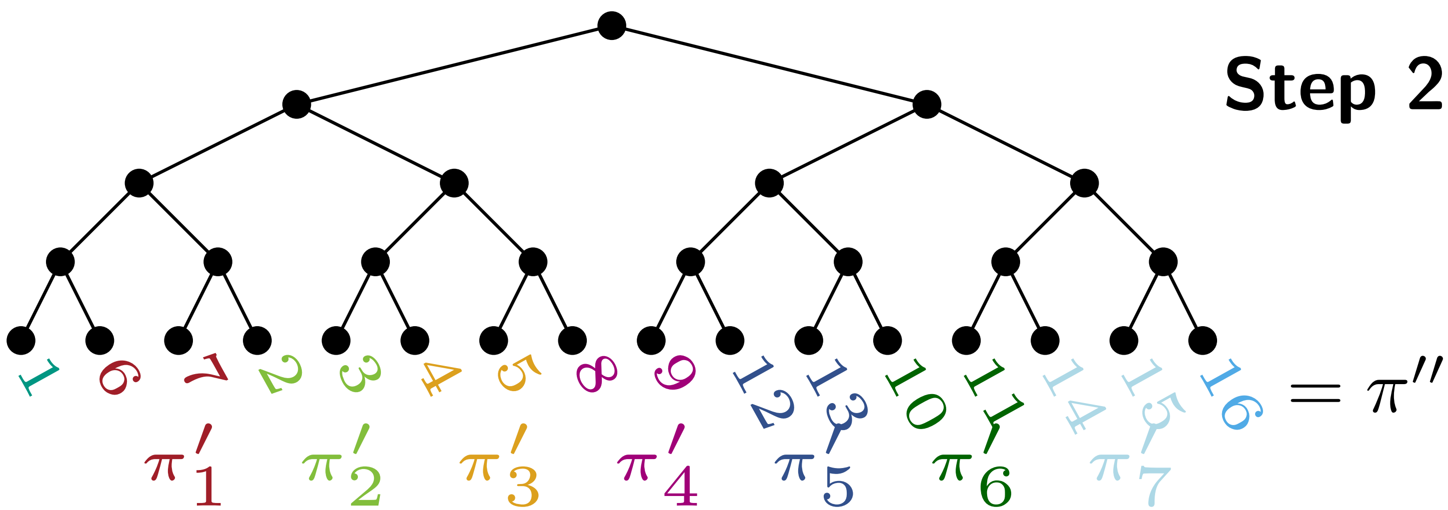
Lemma. $d_t(\pi'') = d_t(\pi)$, $\text{bp}(\pi'') = \text{bp}(\pi)$.

Lemma. π'' is the only permutation consistent with T that has $\text{bp}(\pi'') \leq \text{bp}(\pi)$.

NP-hardness cont.

$\pi = (3, 1, 4, 6, 5, 7)$

$\pi' = (3, 1, 2, 4, 6, 5, 7)$



Step 1: create π' of size $2^p - 1$ with $d_t(\pi) = d_t(\pi')$ and $\mathsf{bp}(\pi) = \mathsf{bp}(\pi')$

Step 2: create tree T.

Lemma. $d_t(\pi'') = d_t(\pi)$, $\mathsf{bp}(\pi'') = \mathsf{bp}(\pi)$.

Lemma. π'' is the only permutation consistent with T that has $\mathsf{bp}(\pi'') \leq \mathsf{bp}(\pi)$.

$\Rightarrow d_t(\pi) = \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$ iff. (T, k) is a yes instance for OTBCM with $k = \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$. $\square$

FPT-Algorithm

Instance: A rooted tree T with leaves$(T) = [n]$ and a positive integer k.

Question: Is there a permutation $\pi \in \Pi_n$ consistent T such that there exist transpositions $\tau_1, \ldots, \tau_k$ with $\pi \circ \tau_1 \circ \tau_2 \circ \cdots \circ \tau_k = \mathrm{id}_n$?

FPT-Algorithm

Instance: A rooted tree T with leaves$(T) = [n]$ and a positive integer k.

Question: Is there a permutation $\pi \in \Pi_n$ consistent T such that there exist transpositions $\tau_1, \ldots, \tau_k$ with $\pi \circ \tau_1 \circ \tau_2 \circ \cdots \circ \tau_k = \mathrm{id}_n$?

Theorem. For binary T, we can find π and $\tau_1, \ldots, \tau_k$ in time $f(k) \cdot n^c$ if they exist, and report NO otherwise in the same time.

FPT-Algorithm

Instance: A rooted tree T with leaves$(T) = [n]$ and a positive integer k.
Question: Is there a permutation $\pi \in \Pi_n$ consistent T such that there exist transpositions $\tau_1, \ldots, \tau_k$ with $\pi \circ \tau_1 \circ \tau_2 \circ \cdots \circ \tau_k = \mathrm{id}_n$?

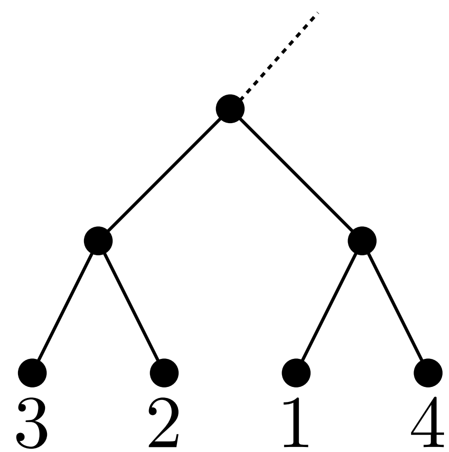
Theorem. For binary T, we can find π and $\tau_1, \ldots, \tau_k$ in time $f(k) \cdot n^c$ if they exist, and report NO otherwise in the same time.

Ingredients:

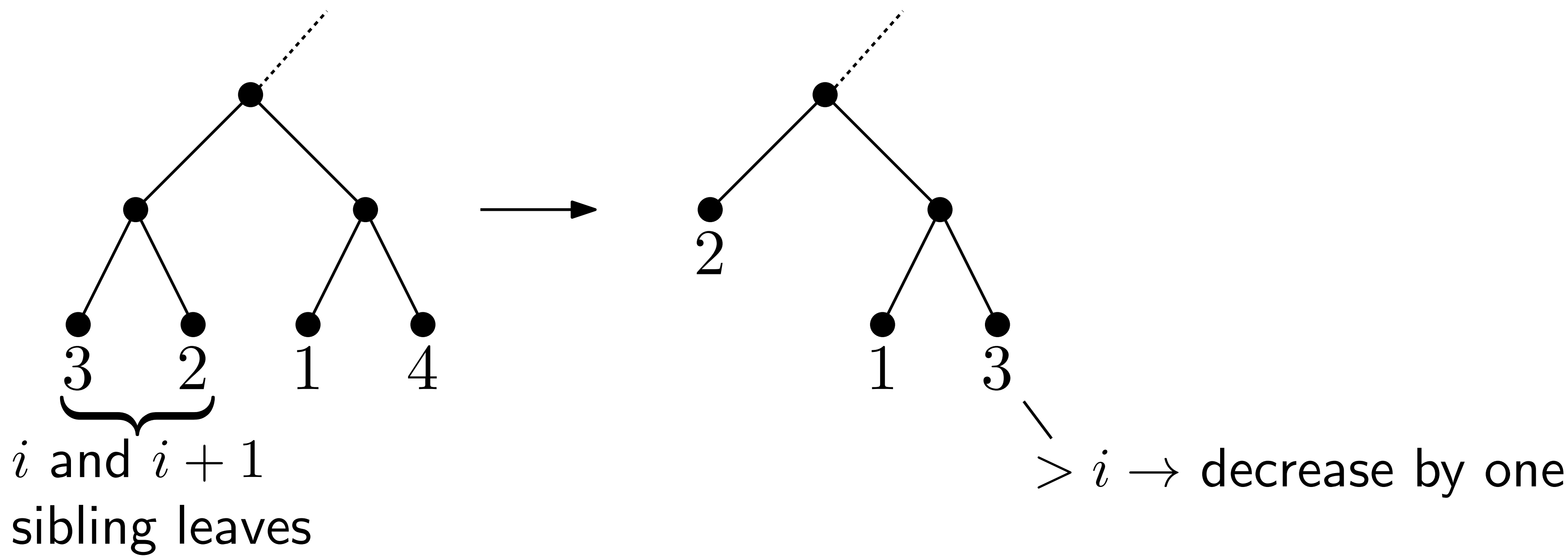
- Two reduction rules
- Search tree algorithm fixing order of children of an inner node
- Leaf ordering π fixed by search tree $\rightarrow$ FPT-algorithm for finding $\tau_1, \ldots, \tau_k$

FPT-Algorithm

Reduction Rule 1. (Preprocessing)



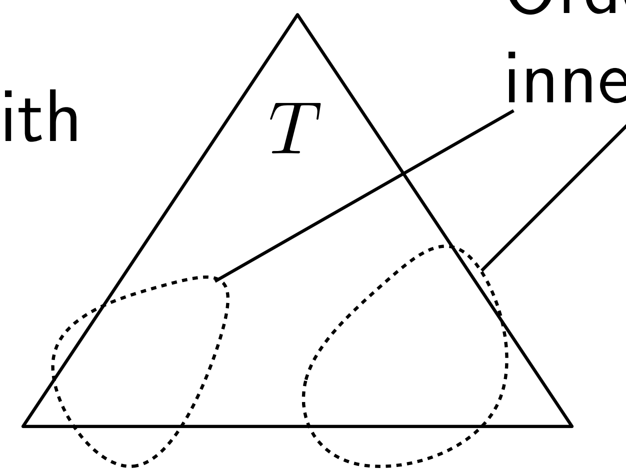
Reduction Rule 1. (Preprocessing)



FPT-Algorithm

Search Tree.

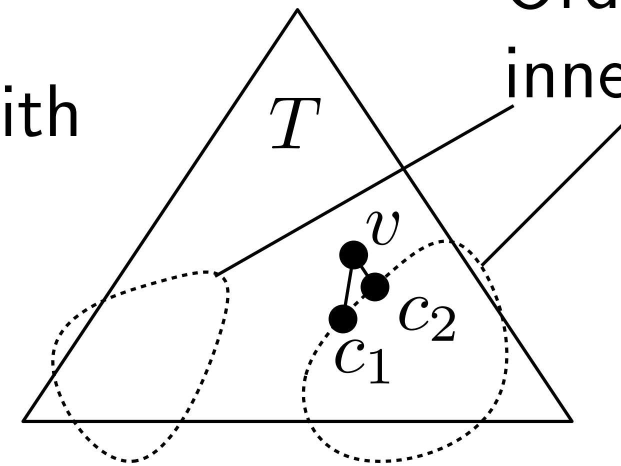
Problem assoc. with search tree node:



FPT-Algorithm

Search Tree.

Problem assoc. with search tree node:

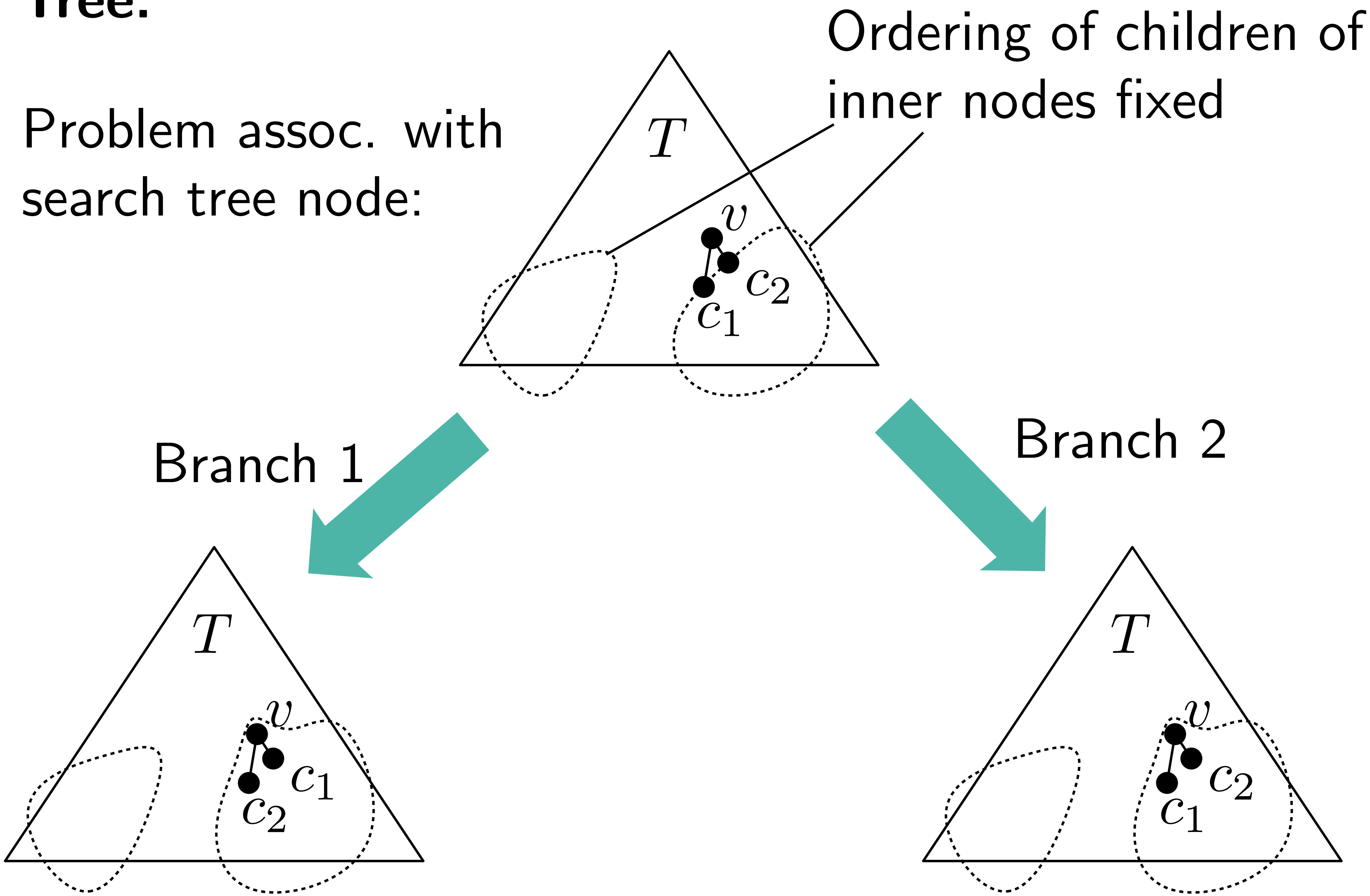


Ordering of children of inner nodes fixed

FPT-Algorithm

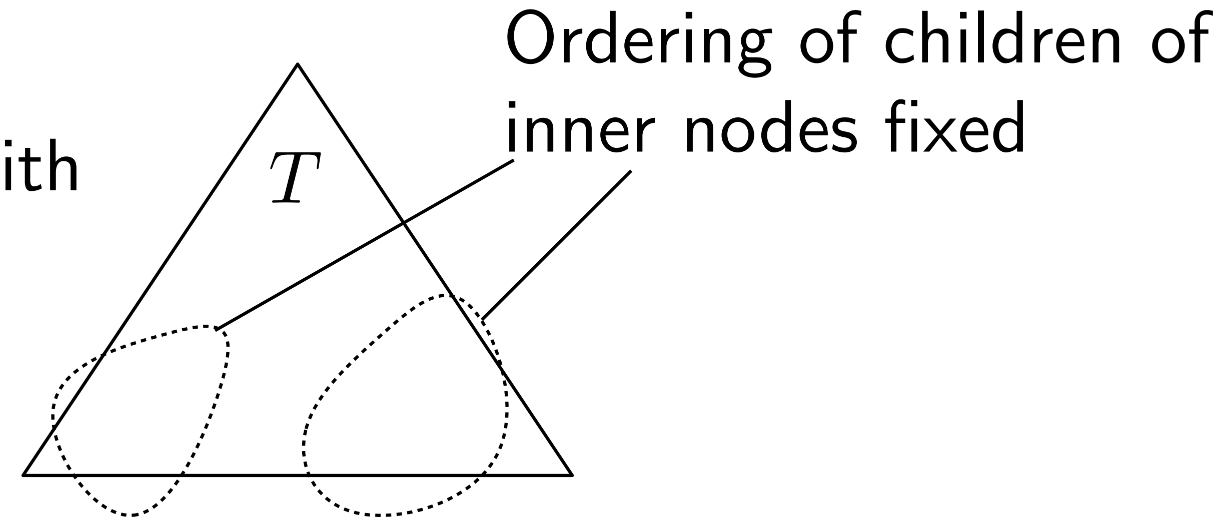
Search Tree.

Problem assoc. with search tree node:



FPT-Algorithm

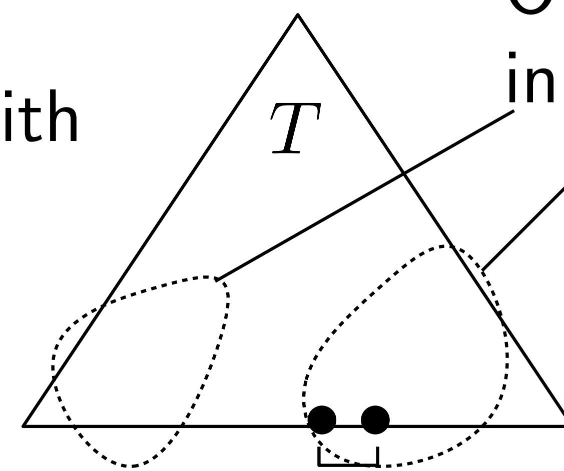
Problem assoc. with search tree node:



FPT-Algorithm

Problem assoc. with search tree node:

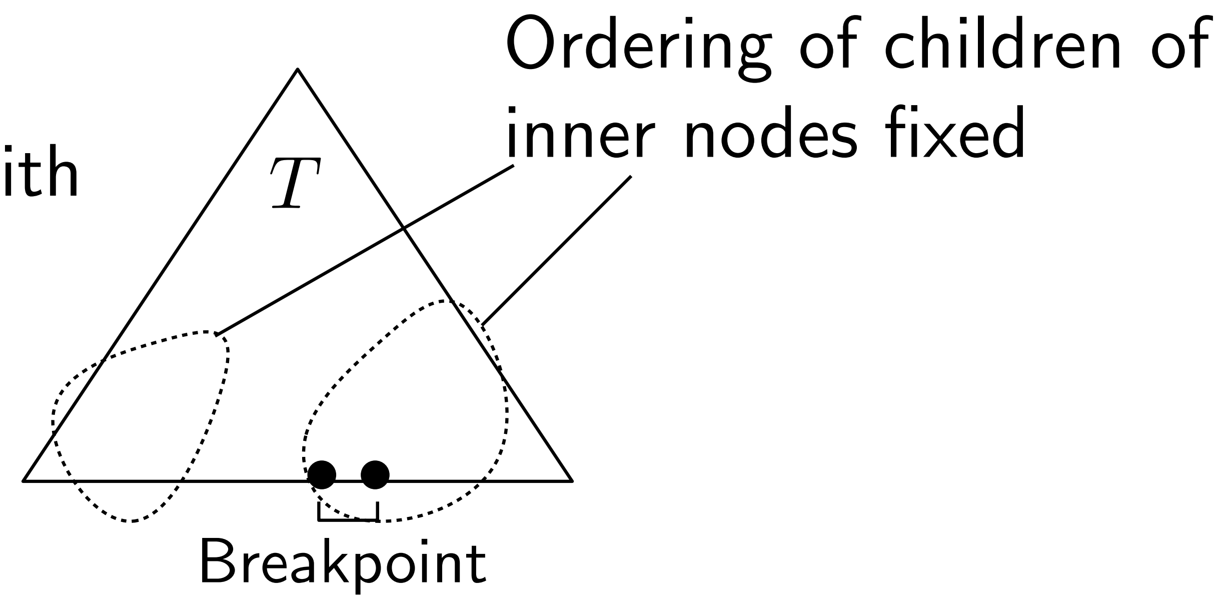
Ordering of children of inner nodes fixed



Breakpoint

FPT-Algorithm

Problem assoc. with search tree node:

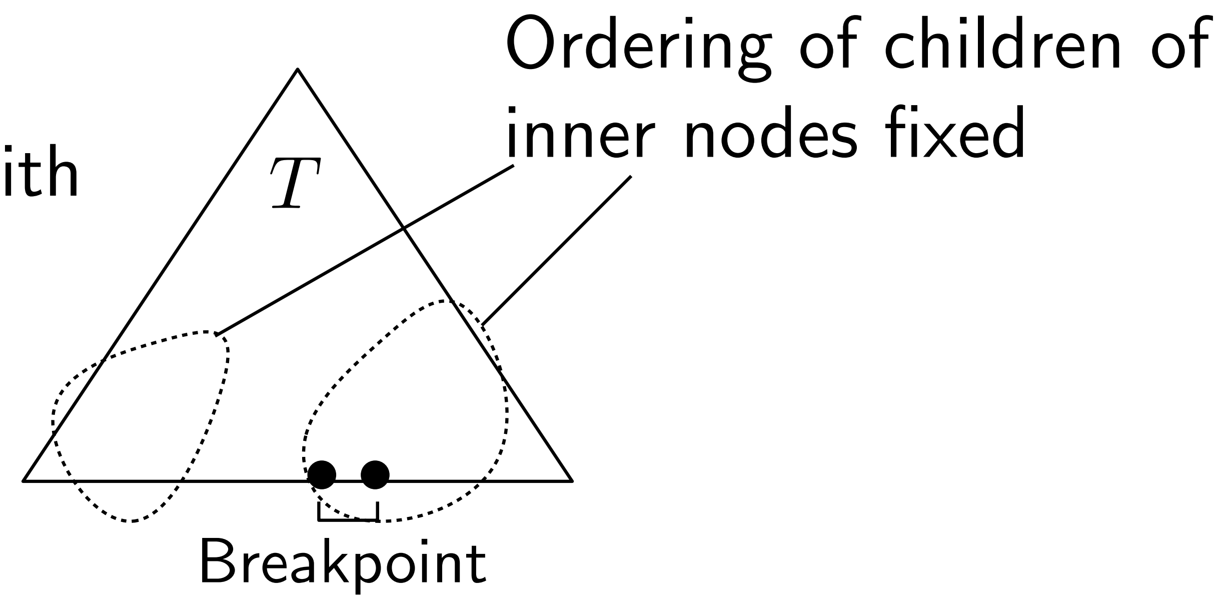


Lemma. (Bafna and Pevzner 1998) $d_t(\pi) \geq \lfloor \frac{\mathsf{bp}(\pi)}{3} \rfloor$.

$> 3k$ breakpoints: Report **failure** for current search tree branch

FPT-Algorithm

Problem assoc. with search tree node:



Lemma. (Bafna and Pevzner 1998) $d_t(\pi) \geq \lfloor \frac{\text{bp}(\pi)}{3} \rfloor$.

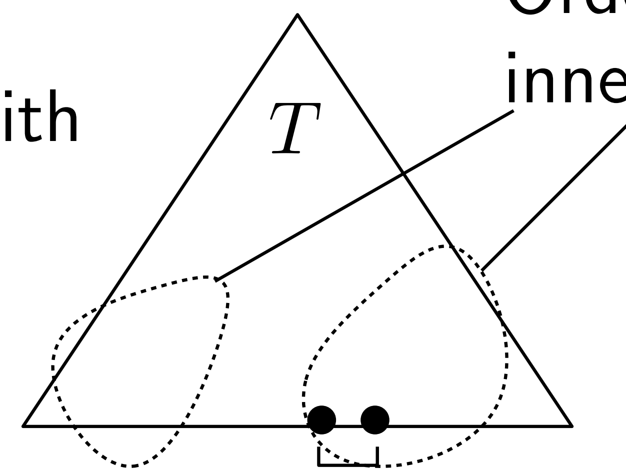
$> 3k$ breakpoints: Report **failure** for current search tree branch

Leaves of search tree: $\leq 3k$ breakpoints, apply FPT-Algorithm for finding $\tau_1, \ldots, \tau_k$.

FPT-Algorithm

Problem assoc. with search tree node:

Ordering of children of inner nodes fixed



Breakpoint

Lemma. (Bafna and Pevzner 1998) $d_t(\pi) \geq \lfloor \frac{\text{bp}(\pi)}{3} \rfloor$.

$> 3k$ breakpoints: Report **failure** for current search tree branch

Leaves of search tree: $\leq 3k$ breakpoints, apply FPT-Algorithm for finding $\tau_1, \ldots, \tau_k$.

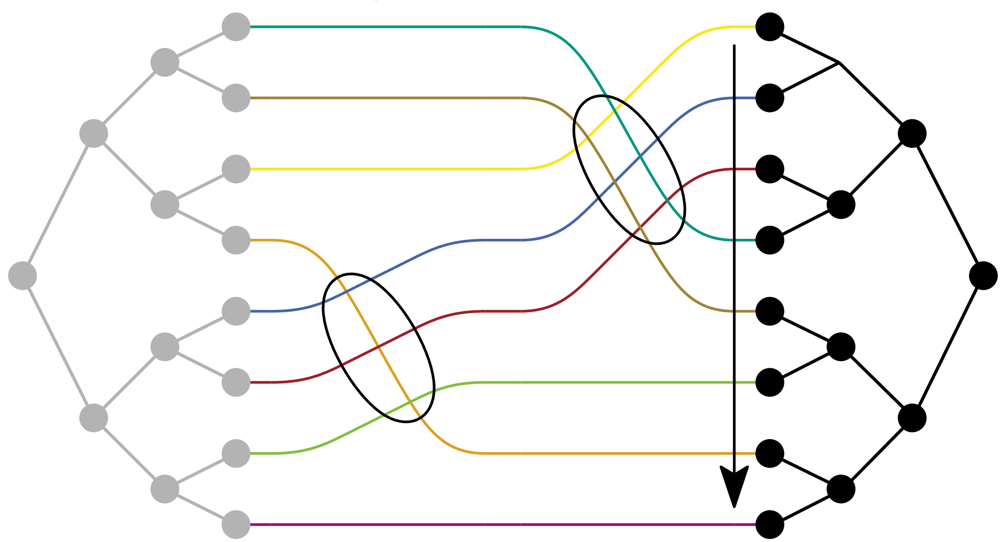
Not enough for bounding runtime $\rightarrow$ **reduction rule 2**
Similar to reduction rule 1, fixes order of children of an inner node

FPT-Algorithm

Instance: A rooted tree T with leaves$(T) = [n]$ and a positive integer k.
Question: Is there a permutation $\pi \in \Pi_n$ consistent with the leaves of T such that there exist transpositions $\tau_1, \ldots, \tau_k$ with $\pi \circ \tau_1 \circ \tau_2 \circ \cdots \circ \tau_k = \mathrm{id}_n$?

Theorem. For binary T, we can find π and $\tau_1, \ldots, \tau_n$ in time $\mathcal{O}(2^{6k} \cdot (3k)^{3k} \cdot n^c)$ if they exist, and report NO otherwise in the same time.

Restr. on T	Block Crossing Min.	Crossing Min.
Complete Binary	**NP-complete** $\mathcal{O}(n^2)$ **2.25-approximation** FPT-algorithm in k	P [Dwyer and Schreiber 2004]
Binary	NP-complete $\mathcal{O}(n^3)$ **2.25-approximation** **FPT-algorithm in k**	P [Dwyer and Schreiber 2004]
Non-binary	NP-complete **Approx. does not extend**	NP-complete [Bulteau et al. 2022]



Open problems:

- Permute leaves of both trees?
- Better approximations?
- Edges can cross multiple times: disallow this?
- Experiments