UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Ph. D. in Computing

Theory, Algorithms and Applications of Arrangements of Trees: Generation, Expectation and Optimization

Lluís Alemany Puig

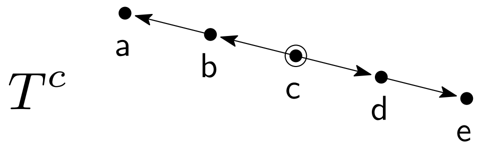
Advisor: Ramon Ferrer i Cancho

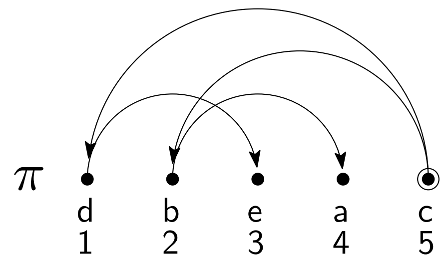
Computer Science Department
Universitat Politèncica de Catalunya – BarcelonaTech

September 26, 2024

The basics

A linear arrangement π of a graph $G = (V, E)$ of n vertices is a permutation of the vertices, a bijection $\pi : V \to [n]$.

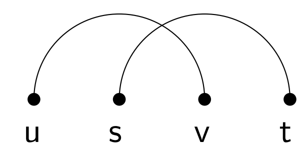




$\pi(d) = 1$
$\pi(b) = 2$
$\pi(e) = 3$
$\pi(a) = 4$
$\pi(c) = 5$

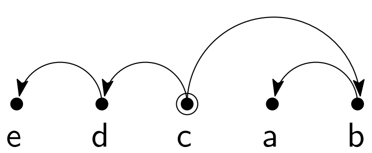
Edge crossing: two edges uv, st cross in π if, w.l.o.g.,

$$\pi(u) < \pi(s) < \pi(v) < \pi(t)$$

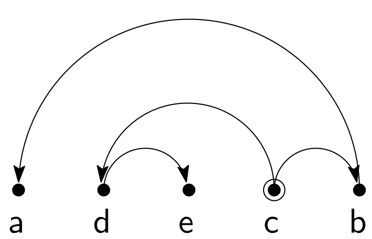


Classes of arrangements

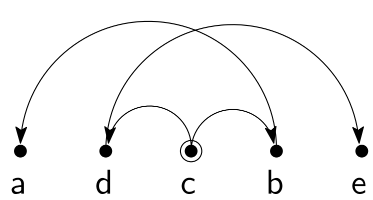
- *Projective* arrangements (of rooted trees): no edge crossings + root not covered
- *Planar* arrangements: no edge crossings
- *Unconstrained* arrangements



Projective arrangement



Planar (non-projective) arrangement

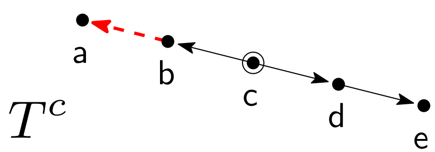


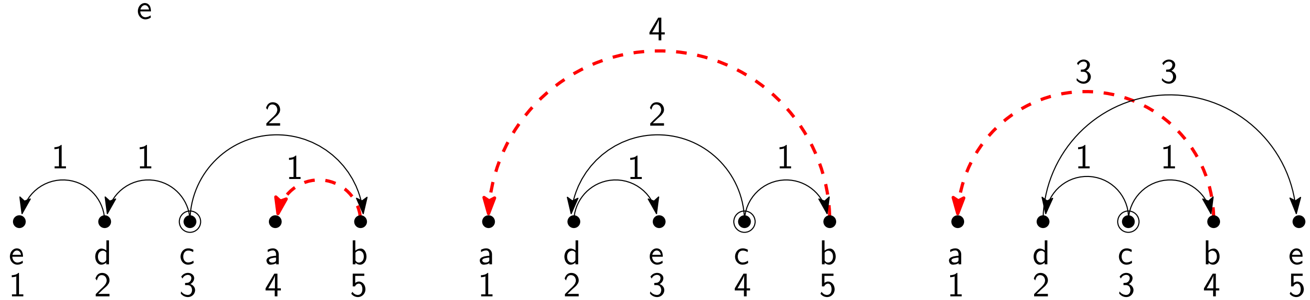
Unconstrained arrangement

Cost function

We focus on problems involving *the sum of edge lengths*. The length of an edge is denoted with δ.

$$D_\pi(G) := \sum_{uv \in E} \delta_\pi(uv), \qquad \delta_\pi(uv) := |\pi(u) - \pi(v)|$$





Projective arrangement	**Planar** (non-projective) arrangement	**Unconstrained** arrangement
$\delta_\pi(ba) = 1$	$\delta_\pi(ba) = 4$	$\delta_\pi(ba) = 3$
$D_\pi(T^c) = 5$	$D_\pi(T^c) = 8$	$D_\pi(T^c) = 8$

Relevant computational problems

We study the range of variation of D

- Optimization problems: minimum/Maximum Linear Arrangement problem ($\mathtt{minLA}$/$\mathtt{MaxLA}$)

$$m[D(G)] := \min_{\pi}\{D_\pi(G)\}, \qquad M[D(G)] := \max_{\pi}\{D_\pi(G)\}.$$

- Expected values: the expected value of the sum of edge lengths of a graph (in a uniformly random arrangement)

$$\mathbb{E}[D(G)] = \frac{1}{n!}\sum_{\pi} D_\pi(G)$$

We tackled three different variants as a function of the type of arrangements: unconstrained, planar and projective.

State of the art

- Minimum Linear Arrangement problem (`minLA`). When unconstrained:
  - **NP**-Hard in general graphs (Garey, Johnson, & Stockmeyer, 1976).
  - Polynomial-time solvable in trees (Goldberg & Klipker, 1976; Shiloach, 1979; Chung, 1984).

  When constrained to projective/planar arrangements of trees:
  - Known polynomial time solutions (Iordanskii, 1987; Hochberg & Stallmann, 2003; Gildea & Temperley, 2007).

State of the art

- Minimum Linear Arrangement problem (`minLA`). When unconstrained:
  - **NP**-Hard in general graphs (Garey, Johnson, & Stockmeyer, 1976).
  - Polynomial-time solvable in trees (Goldberg & Klipker, 1976; Shiloach, 1979; Chung, 1984).

  When constrained to projective/planar arrangements of trees:
  - Known polynomial time solutions (Iordanskii, 1987; Hochberg & Stallmann, 2003; Gildea & Temperley, 2007).
- Maximum Linear Arrangement problem (`MaxLA`). When unconstrained:
  - **NP**-Hard in general graphs (Even & Shiloach, 1975).
  - *Unknown* if it is polynomial-time solvable in trees, but some solutions were known for specific classes (Ferrer-i-Cancho, Gómez-Rodríguez, & Esteban, 2021).

  When constrained to projective/planar (or other types of) arrangements of trees:
  - No known polynomial time solutions.

State of the art

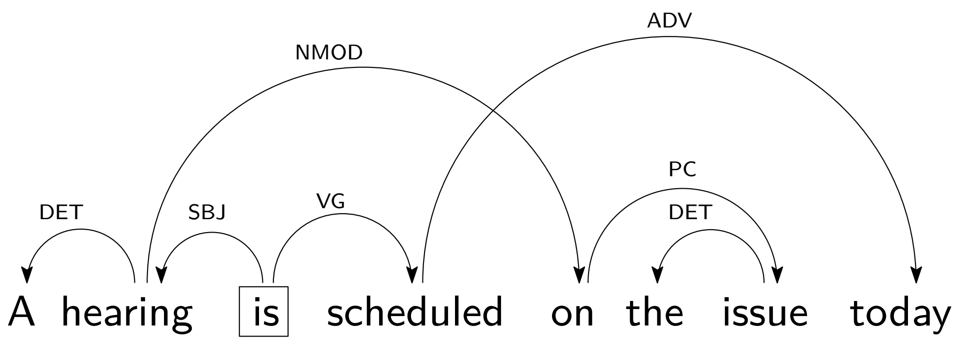
- Minimum Linear Arrangement problem (`minLA`). When unconstrained:
  - **NP**-Hard in general graphs (Garey, Johnson, & Stockmeyer, 1976).
  - Polynomial-time solvable in trees (Goldberg & Klipker, 1976; Shiloach, 1979; Chung, 1984).

  When constrained to projective/planar arrangements of trees:
  - Known polynomial time solutions (Iordanskii, 1987; Hochberg & Stallmann, 2003; Gildea & Temperley, 2007).
- Maximum Linear Arrangement problem (`MaxLA`). When unconstrained:
  - **NP**-Hard in general graphs (Even & Shiloach, 1975).
  - *Unknown* if it is polynomial-time solvable in trees, but some solutions were known for specific classes (Ferrer-i-Cancho, Gómez-Rodríguez, & Esteban, 2021).

  When constrained to projective/planar (or other types of) arrangements of trees:
  - No known polynomial time solutions.
- Expected values of D
  - Known formulas for unconstrained arrangements of graphs (and trees) (Ferrer-i-Cancho, 2004, 2019).
  - Unknown formulas for projective/planar arrangements of trees.

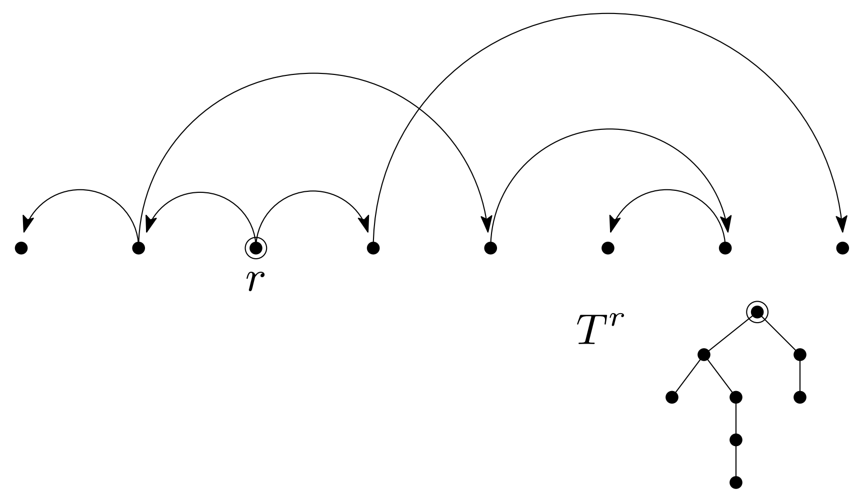
Application – Quantitative Dependency Syntax

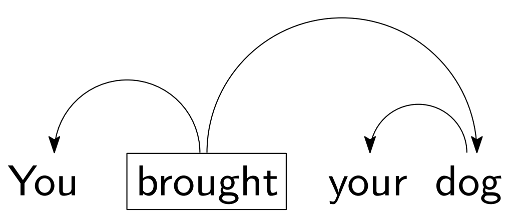
- **Quantitative Linguistics** concerned with statistical properties of language: length, frequency, ... of linguistic units (morphs, syllables, sentences, ...).
- **Dependency Syntax** studies syntactic dependency structures (rooted tree structures where the verb is the root vertex).



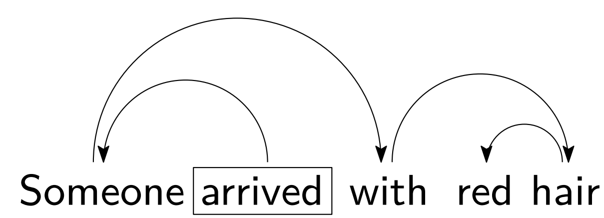
Unconstrained

Rooted tree T^r + linear arrangement π





Projective



Planar (non-projective)

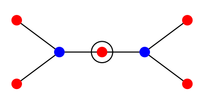
Articles and main contributions in this thesis

	Unconstrained	Planar	Projective
Maximum	(A.-P., Esteban, & Ferrer-i-Cancho, 2023)	(A.-P., Esteban, & Ferrer-i-Cancho, 2024) $O(n)$	$O(n)$
Expected	(Ferrer-i-Cancho, 2004, 2019) $O(1)$	(A.-P. & Ferrer-i-Cancho, 2024) $O(n)$	(A.-P. & Ferrer-i-Cancho, 2022) $O(n)$
Minimum	(Chung, 1984) $O(n^{1.58})$	(A.-P., Esteban, & Ferrer-i-Cancho, 2022) $O(n)$	$O(n)$

- A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400. https://doi.org/10.1016/j.ipl.2023.106400
- A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362
- A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442
- A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204
- A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487
- A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2021). The Linear Arrangement Library. A new tool for research on syntactic dependency structures. *Proceedings of the Second Workshop on Quantitative Syntax (Quasy, SyntaxFest 2021)*, 1–16. https://aclanthology.org/2021.quasy-1.1

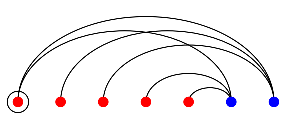
Articles and main contributions in this thesis

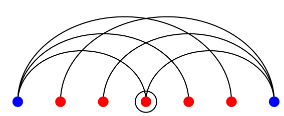
	Unconstrained	Planar	Projective
Maximum	(A.-P., Esteban, & Ferrer-i-Cancho, 2023)	(A.-P., Esteban, & Ferrer-i-Cancho, 2024) $O(n)$	$O(n)$
Expected	(Ferrer-i-Cancho, 2004, 2019) $O(1)$	(A.-P. & Ferrer-i-Cancho, 2024) $O(n)$	(A.-P. & Ferrer-i-Cancho, 2022) $O(n)$
Minimum	(Chung, 1984) $O(n^{1.58})$	(A.-P., Esteban, & Ferrer-i-Cancho, 2022) $O(n)$	$O(n)$



Bipartite arrangements

Arrangements with 1 thistle





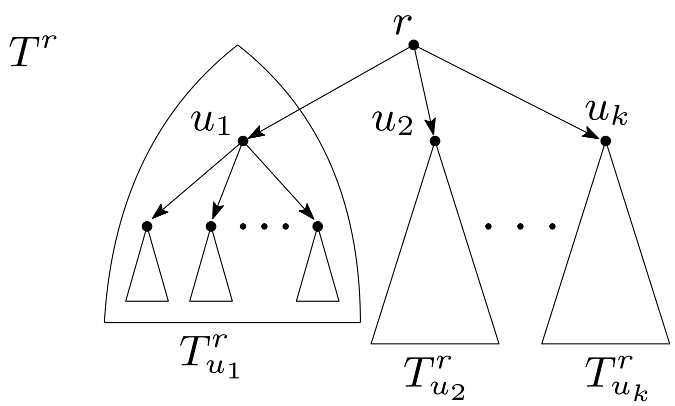
	1-thistle	Bipartite
Maximum	(A.-P., Esteban, & Ferrer-i-Cancho, 2023) $O(n^3 2^{\Delta})$	$O(n)$

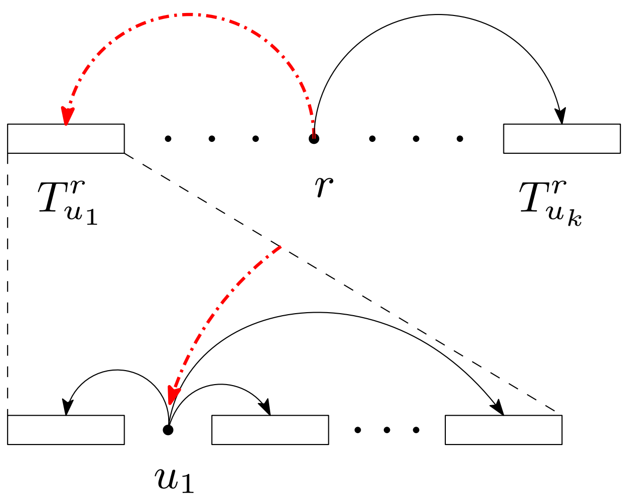
- A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442
- A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204
- A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487
- A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2021). The Linear Arrangement Library. A new tool for research on syntactic dependency structures. *Proceedings of the Second Workshop on Quantitative Syntax (Quasy, SyntaxFest 2021)*, 1–16. https://aclanthology.org/2021.quasy-1.1

Generation of arrangements – Projective case

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

Sampling projective arrangements uniformly at random in time $O(n)$ was described by Futrell, Mahowald, and Gibson (2015).

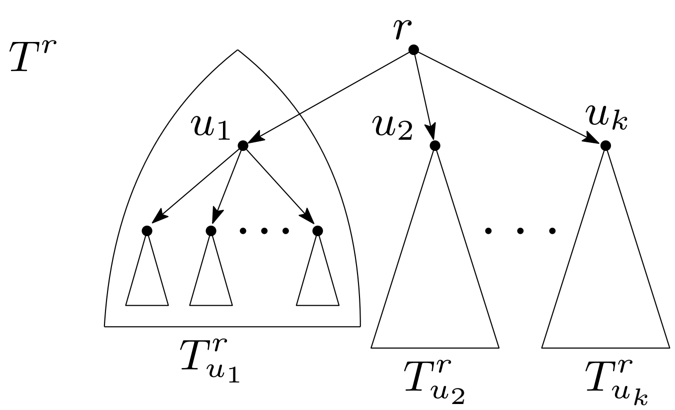


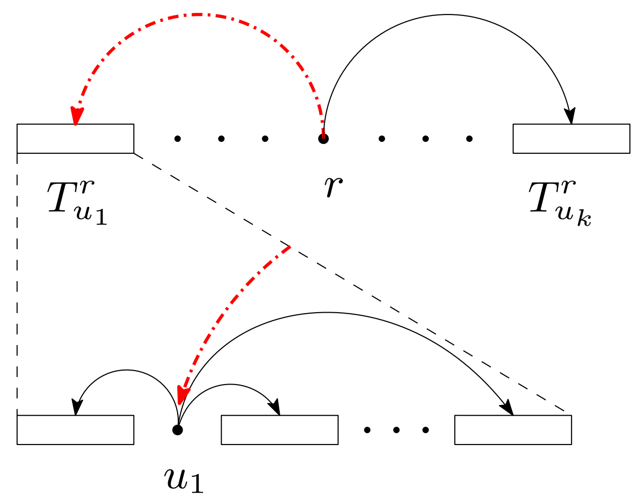


Generation of arrangements – Projective case

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

Sampling projective arrangements uniformly at random in time $O(n)$ was described by Futrell, Mahowald, and Gibson (2015).





Function $R_{\text{pr}}(T^r)$ **is**

In: T^r a rooted tree

Out: A projective arrangement of T^r selected u.a.r.

$[z_1, \dots, z_i, r, z_{i+1}, \dots, z_k] \leftarrow$ uniformly random permutation of $[u_1, \dots, u_k, r]$

return $R_{\text{pr}}(T^r_{z_1}) : \cdots : R_{\text{pr}}(T^r_{z_i}) : r : R_{\text{pr}}(T^r_{z_{i+1}}) : \cdots : R_{\text{pr}}(T^r_{z_k})$

Generation of arrangements – Planar case

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

A planar arrangement of a free tree T is a projective arrangement of T^u where u is the vertex at the leftmost position.

Function $R_{\text{pl}}(T)$ **is**

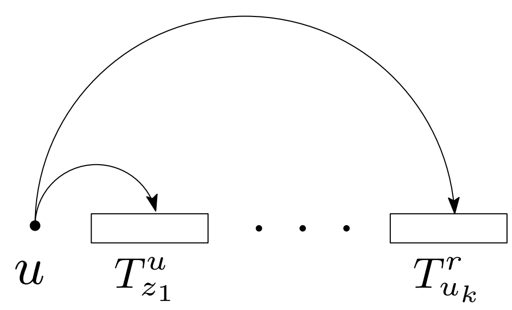
In: T a free tree

Out: A planar arrangement of T selected u.a.r.

$u \leftarrow$ choose a root u.a.r.

$[z_1, \ldots, z_k] \leftarrow$ uniformly random permutation of the children of u

return $u : R_{\text{pr}}(T^u_{z_1}) : \cdots : R_{\text{pr}}(T^u_{z_k})$



Expectation of D

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

How can we calculate the expectation of D over (*uniformly random*) projective, planar and unconstrained arrangements? With random sampling?

- High error due to undersampling. When approximating expected D over projective arrangements:
  - Kramer (2021): 10 samples (**relative error** $\approx 20\%$).
  - Futrell, Mahowald, and Gibson (2015) and Futrell, Levy, and Gibson (2020): 100 samples (**relative error**: $\approx 10\%$).
  - Gildea and Temperley (2007), Park and Levy (2009), Gildea and Temperley (2010), and Gulordava and Merlo (2015): not reported.
- Attempt to describe uniform sampling methods (Gildea & Temperley, 2007; Gildea & Temperley, 2010; Temperley & Gildea, 2018) that were not actually uniform.

Expectation of D

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

How can we calculate the expectation of D over (*uniformly random*) projective, planar and unconstrained arrangements? With random sampling?

- High error due to undersampling. When approximating expected D over projective arrangements:
  - Kramer (2021): 10 samples (**relative error** $\approx 20\%$).
  - Futrell, Mahowald, and Gibson (2015) and Futrell, Levy, and Gibson (2020): 100 samples (**relative error**: $\approx 10\%$).
  - Gildea and Temperley (2007), Park and Levy (2009), Gildea and Temperley (2010), and Gulordava and Merlo (2015): not reported.
- Attempt to describe uniform sampling methods (Gildea & Temperley, 2007; Gildea & Temperley, 2010; Temperley & Gildea, 2018) that were not actually uniform.

$$\mathbb{E}[D(G)] = \text{expected value of } D \text{ in a uniformly random arrangement of } G$$
$$= \frac{n+1}{3}m$$
$$\mathbb{E}_{\text{pr}}[D(T^r)] := \mathbb{E}[D(T^r) \mid \text{projective arrangements}]$$
$$\mathbb{E}_{\text{pl}}[D(T)] := \mathbb{E}[D(T) \mid \text{planar arrangements}]$$

Expectation of D

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516.
https://doi.org/10.1162/coli_a_00442

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

Main contributions

Formulas for the expected value of δ and D in projective and planar arrangements.

- For any rooted tree T^r

$$\mathbb{E}_{\mathsf{pr}}[\delta(uv)] = \frac{2s_r(u) + s_r(v) + 1}{6}$$

$$\mathbb{E}_{\mathsf{pr}}[D(T^r)] = \frac{1}{6}\left(-1 + \sum_{v \in V} s_r(v)(2d_r(v) + 1)\right)$$

Expectation of D

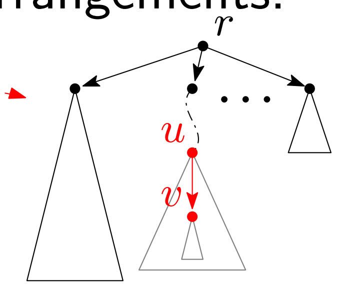
A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

Main contributions

Formulas for the expected value of δ and D in projective and planar arrangements.

- For any rooted tree T^r



$$\mathbb{E}_{\text{pr}}[\delta(uv)] = \frac{2s_r(u) + s_r(v) + 1}{6}$$

$$\mathbb{E}_{\text{pr}}[D(T^r)] = \frac{1}{6}\left(-1 + \sum_{v \in V} s_r(v)(2d_r(v) + 1)\right)$$

Expectation of D

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516.
https://doi.org/10.1162/coli_a_00442

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

Main contributions

Formulas for the expected value of δ and D in projective and planar arrangements.

- For any rooted tree T^r

$$\mathbb{E}_{\text{pr}}[\delta(uv)] = \frac{2s_r(u) + s_r(v) + 1}{6}$$

$$\mathbb{E}_{\text{pr}}[D(T^r)] = \frac{1}{6}\left(-1 + \sum_{v \in V} s_r(v)(2d_r(v) + 1)\right)$$

- For any free tree T

$$\mathbb{E}_{\text{pl}}[\delta(uv)] = 1 + \frac{1}{n} \sum_{r \in V \setminus \{u,v\}} \mathbb{E}_{\text{pr}}[\delta(uv) \mid r]$$

$$\mathbb{E}_{\text{pl}}[D(T)] = \frac{(n-1)(n-2)}{6n} + \frac{1}{n} \sum_{u \in V} \mathbb{E}_{\text{pr}}[D(T^u)].$$

Plus a $O(n)$-time algorithm for $\mathbb{E}_{\text{pl}}[D(T)]$.

Expectation of D

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

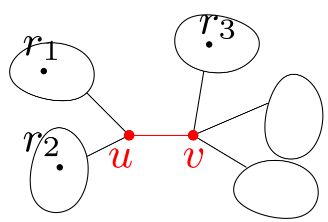
Main contributions

Formulas for the expected value of δ and D in projective and planar arrangements.

- For any rooted tree T^r

$$\mathbb{E}_{\text{pr}}[\delta(uv)] = \frac{2s_r(u) + s_r(v) + 1}{6}$$

$$\mathbb{E}_{\text{pr}}[D(T^r)] = \frac{1}{6}\left(-1 + \sum_{v \in V} s_r(v)(2d_r(v) + 1)\right)$$



- For any free tree T

$$\mathbb{E}_{\text{pl}}[\delta(uv)] = 1 + \frac{1}{n}\sum_{r \in V \setminus \{u,v\}} \mathbb{E}_{\text{pr}}[\delta(uv) \mid r]$$

$$\mathbb{E}_{\text{pl}}[D(T)] = \frac{(n-1)(n-2)}{6n} + \frac{1}{n}\sum_{u \in V} \mathbb{E}_{\text{pr}}[D(T^u)].$$

Plus a $O(n)$-time algorithm for $\mathbb{E}_{\text{pl}}[D(T)]$.

Minimum arrangements – Projective and planar

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

Review of the calculation of

$$m_{\text{pr}}[D(T^r)] := \min_{\pi \text{ projective}} \{D_\pi(T^r)\}, \qquad m_{\text{pl}}[D(T)] := \min_{\pi \text{ planar}} \{D_\pi(T)\}.$$

- Calculation of $m_{\text{pr}}[D(T^r)]$: Gildea and Temperley (2007).
- Calculation of $m_{\text{pl}}[D(T)]$: Hochberg and Stallmann (2003) and Iordanskii (1987).

The algorithms devised in each paper are fundamentally different from each other.

Minimum arrangements – Projective and planar

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

Review of the calculation of

$$m_{\text{pr}}[D(T^r)] := \min_{\pi \text{ projective}} \{D_\pi(T^r)\}, \qquad m_{\text{pl}}[D(T)] := \min_{\pi \text{ planar}} \{D_\pi(T)\}.$$

- Calculation of $m_{\text{pr}}[D(T^r)]$: Gildea and Temperley (2007).
- Calculation of $m_{\text{pl}}[D(T)]$: Hochberg and Stallmann (2003) and Iordanskii (1987).

The algorithms devised in each paper are fundamentally different from each other.

Gildea and Temperley (2007)

- Presented a sketch (they gave no pseudocode) of the algorithm, and made a minor mistake in its description.[1]
- They claimed their algorithm runs in time $O(n)$ but this depends on the choice of sorting algorithm and storage of the data before sorting, which was not specified.

Hochberg and Stallmann (2003) made a minor mistake in their algorithm.

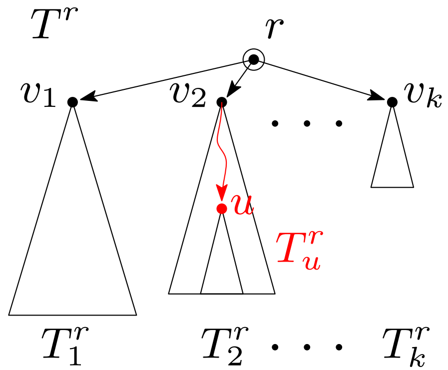
The work by Iordanskii (1987), in Russian, was not cited.

[1] Mistake pointed out by Bommasani (2020).

Minimum arrangements – Projective case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

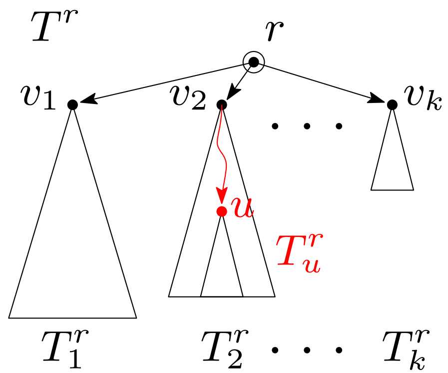
Gildea and Temperley (2007): distribute subtrees over disjoint, contiguous intervals.

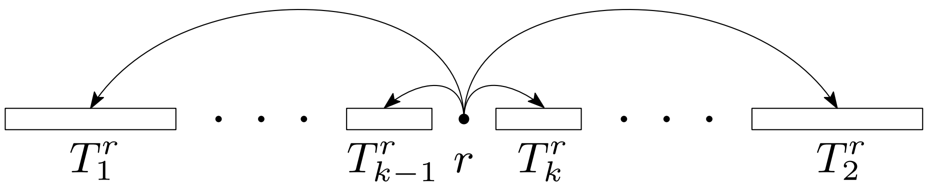


Minimum arrangements – Projective case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

Gildea and Temperley (2007): distribute subtrees over disjoint, contiguous intervals.

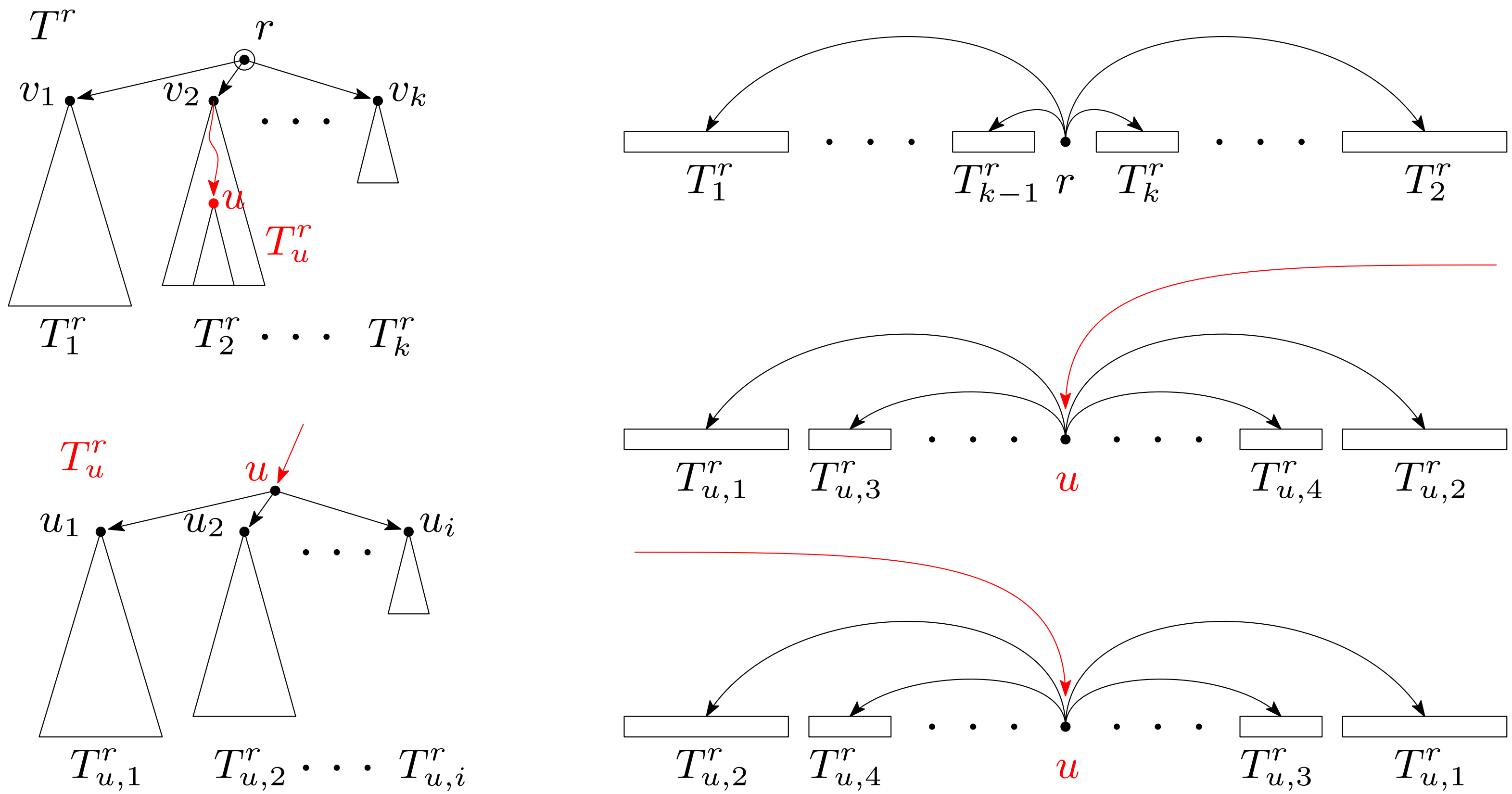




Minimum arrangements – Projective case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

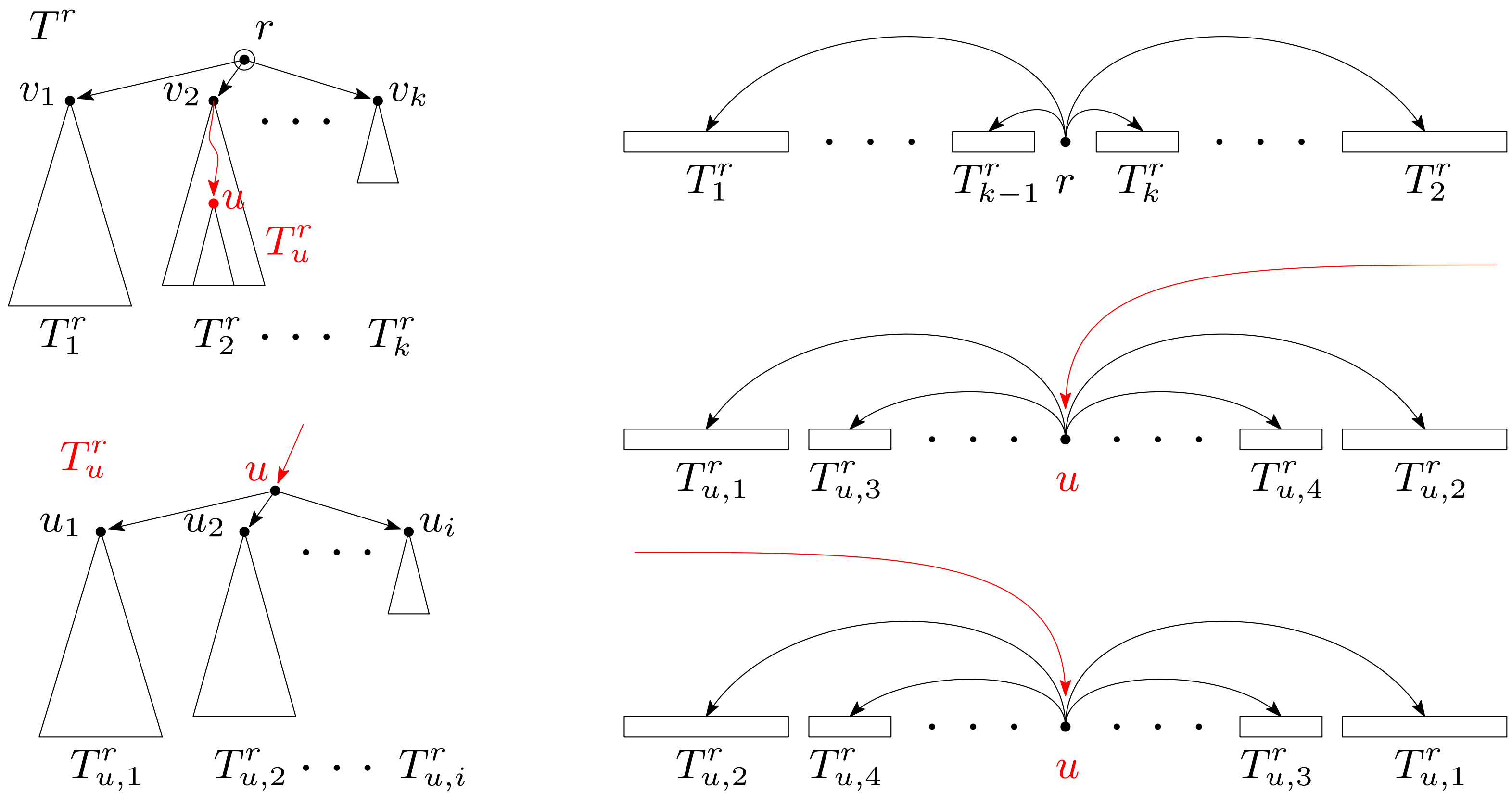
Gildea and Temperley (2007): distribute subtrees over disjoint, contiguous intervals.



Minimum arrangements – Projective case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

Gildea and Temperley (2007): distribute subtrees over disjoint, contiguous intervals.



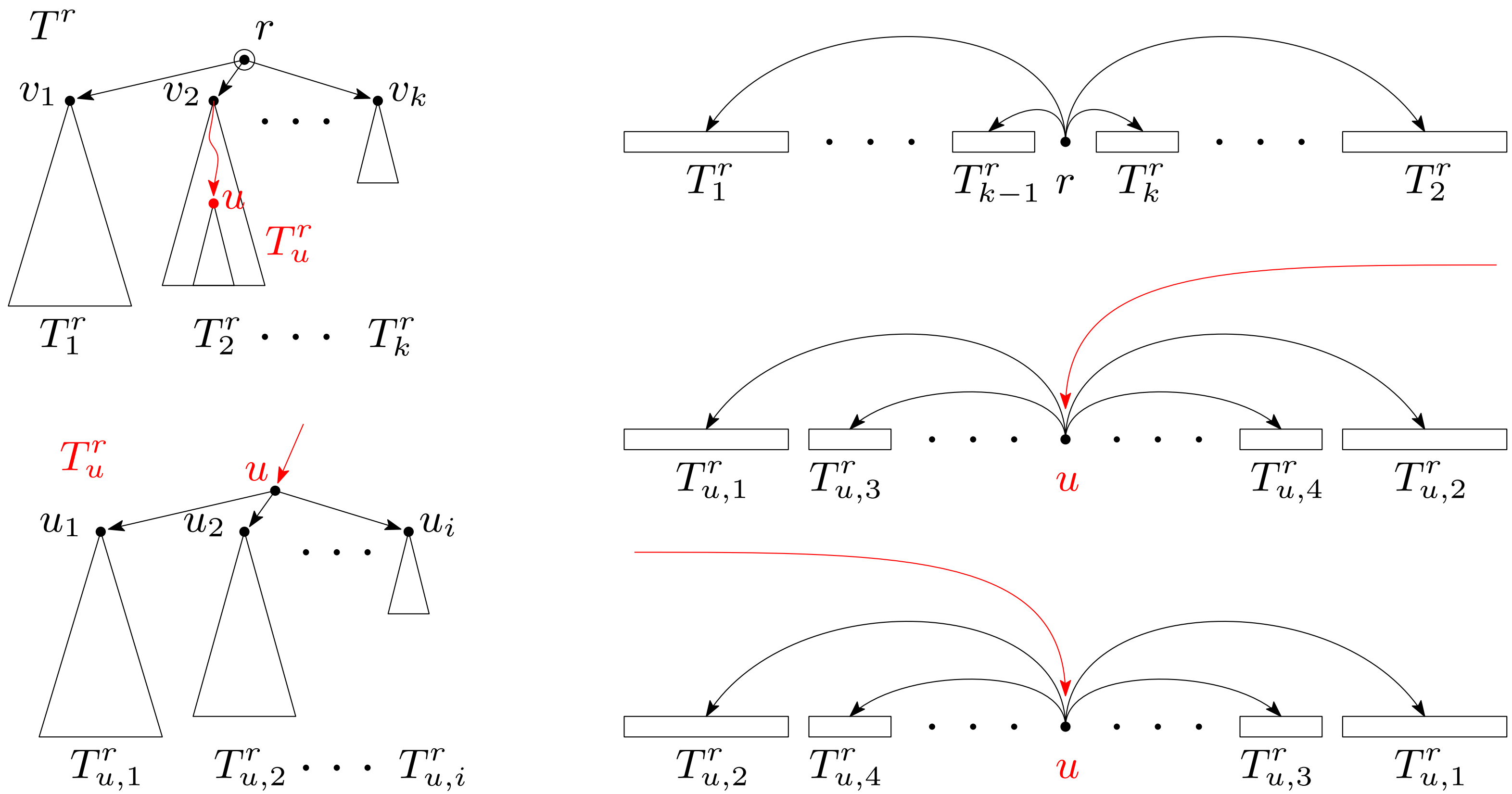
Gildea and Temperley (2007): "*If there are an **odd** number of children, the side of the final (smallest) child makes no difference, because the other children are evenly balanced on the two sides [...]*"[1]

[1] Mistake pointed out by Bommasani (2020).

Minimum arrangements – Projective case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

Gildea and Temperley (2007): distribute subtrees over disjoint, contiguous intervals.



We use *Counting sort* for this algorithm. To sort all the subtrees of a rooted tree T^r in time and space $O(n)$, build the list of tuples

$$(u, v, s_r(v)) \quad \forall (u, v) \in E, \qquad s_r(v) := |V(T^r_v)|,$$

sort it, and then store the required values in an adjacency list-like data structure.

Minimum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

Hochberg and Stallmann (2003): place a centroidal vertex c between the subtrees of T^c. The embedding method for subtrees calculates, for each vertex u, a displacement/offset $relPos[u]$ with respect to c.

$$\pi(u) \leftarrow \pi(c) + relPos[u]$$

We can use intervals to construct a minimum planar arrangement:

Function Minimum_Planar(T) **is**

- **In:** T a free tree.
- **Out:** A minimum planar arrangement of T.
- $c \leftarrow$ a centroidal vertex of T in time $O(n)$
- **return** Minimum_Projective(T^c)

Maximum arrangements – Projective case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400. https://doi.org/10.1016/j.ipl.2023.106400

The construction of a maximum projective arrangement is quite similar to the construction of a minimum projective arrangement.

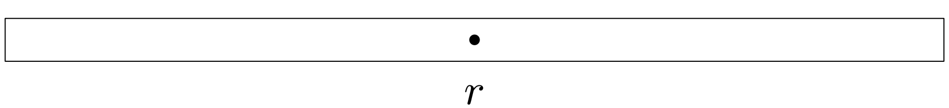
Maximum arrangements – Projective case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

The construction of a maximum projective arrangement is quite similar to the construction of a minimum projective arrangement.

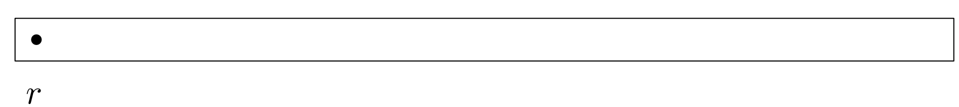
Minimum projective arrangements

- Arrange the subtrees of the tree to both sides of the root in a balanced manner.



Maximum projective arrangements

- Place the root at one end of the arrangement (left)



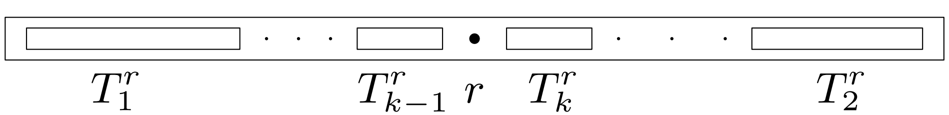
Maximum arrangements – Projective case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

The construction of a maximum projective arrangement is quite similar to the construction of a minimum projective arrangement.

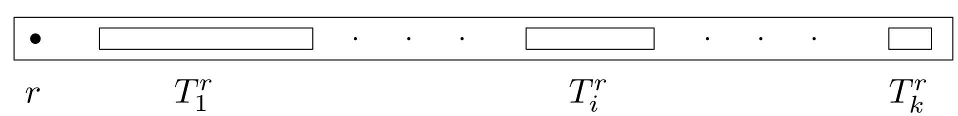
Minimum projective arrangements

- Arrange the subtrees of the tree to both sides of the root in a balanced manner.
- The subtrees are sorted from largest to smallest (inwards).



Maximum projective arrangements

- Place the root at one end of the arrangement (left)
- The subtrees of the root are sorted from largest to smallest (left to right).



Maximum arrangements – Projective case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

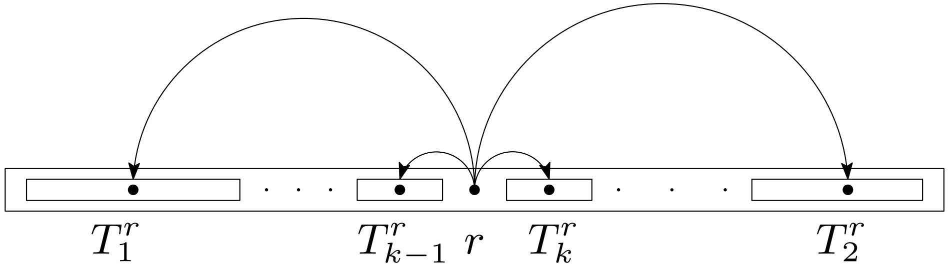
The construction of a maximum projective arrangement is quite similar to the construction of a minimum projective arrangement.

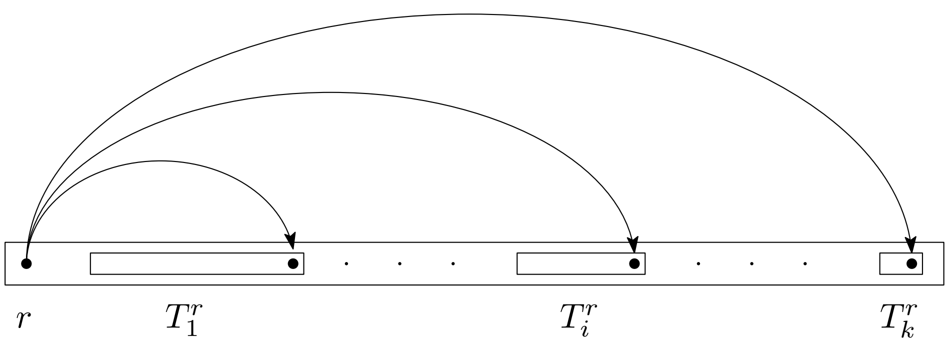
Minimum projective arrangements

- Arrange the subtrees of the tree to both sides of the root in a balanced manner.
- The subtrees are sorted from largest to smallest (inwards).
- The root of each subtree is always placed in between its subtrees in a balanced manner.

Maximum projective arrangements

- Place the root at one end of the arrangement (left)
- The subtrees of the root are sorted from largest to smallest (left to right).
- The root of each subtree is placed at the opposite end compared to its parent's.





Maximum arrangements – Projective case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

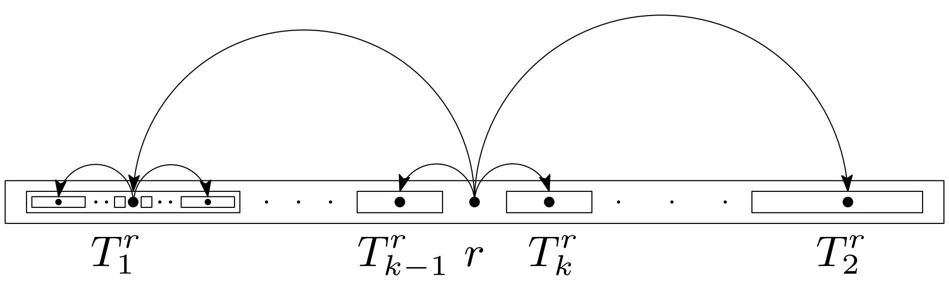
The construction of a maximum projective arrangement is quite similar to the construction of a minimum projective arrangement.

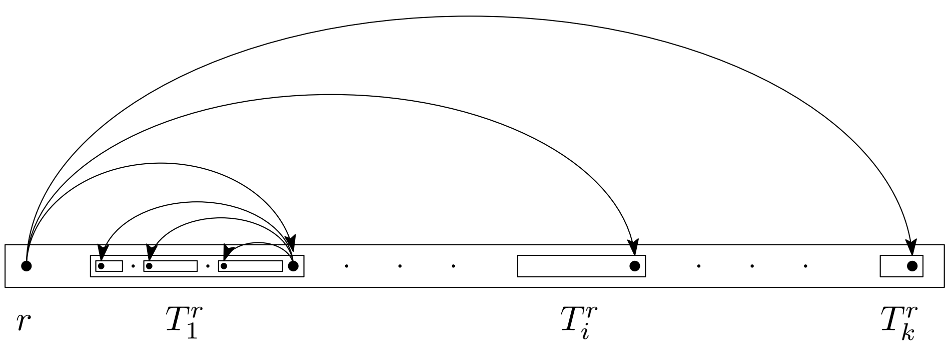
Minimum projective arrangements

- Arrange the subtrees of the tree to both sides of the root in a balanced manner.
- The subtrees are sorted from largest to smallest (inwards).
- The root of each subtree is always placed in between its subtrees in a balanced manner.
- Apply the steps above recursively for each subtree.

Maximum projective arrangements

- Place the root at one end of the arrangement (left)
- The subtrees of the root are sorted from largest to smallest (left to right).
- The root of each subtree is placed at the opposite end compared to its parent's.
- Apply the steps above recursively for each subtree.





Maximum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

Naïve method: to get $M_{\mathsf{pl}}[D(T)]$, calculate $M_{\mathsf{pr}}[D(T^u)]$ for all $u \to O\big(n^2\big)$ algorithm:

$$M_{\mathsf{pl}}[D(T)] = \max_{u \in V}\{M_{\mathsf{pr}}[D(T^u)]\}.$$

Maximum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

Naïve method: to get $M_{\mathsf{pl}}[D(T)]$, calculate $M_{\mathsf{pr}}[D(T^u)]$ for all $u \to O\big(n^2\big)$ algorithm:

$$M_{\mathsf{pl}}[D(T)] = \max_{u \in V}\{M_{\mathsf{pr}}[D(T^u)]\}.$$

Our contribution: the values $M_{\mathsf{pr}}[D(T^u)]$ and $M_{\mathsf{pr}}[D(T^v)]$ are related when $uv \in E$. At every step of a Breadth-First Search traversal (from vertex u to v), calculate $M_{\mathsf{pr}}[D(T^v)]$ using the value $M_{\mathsf{pr}}[D(T^u)]$ in time $O(1)$. Total time $O(n)$.

Maximum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

Naïve method: to get $M_{\text{pl}}[D(T)]$, calculate $M_{\text{pr}}[D(T^u)]$ for all $u \to O\left(n^2\right)$ algorithm:

$$M_{\text{pl}}[D(T)] = \max_{u \in V}\{M_{\text{pr}}[D(T^u)]\}.$$

Our contribution: the values $M_{\text{pr}}[D(T^u)]$ and $M_{\text{pr}}[D(T^v)]$ are related when $uv \in E$.
At every step of a Breadth-First Search traversal (from vertex u to v), calculate $M_{\text{pr}}[D(T^v)]$ using the value $M_{\text{pr}}[D(T^u)]$ in time $O(1)$. Total time $O(n)$.

For any $uv \in E$:

$$M_{\text{pr}}[D(T^v)] - M_{\text{pr}}[D(T^u)] = f(v,u) - f(u,v),$$

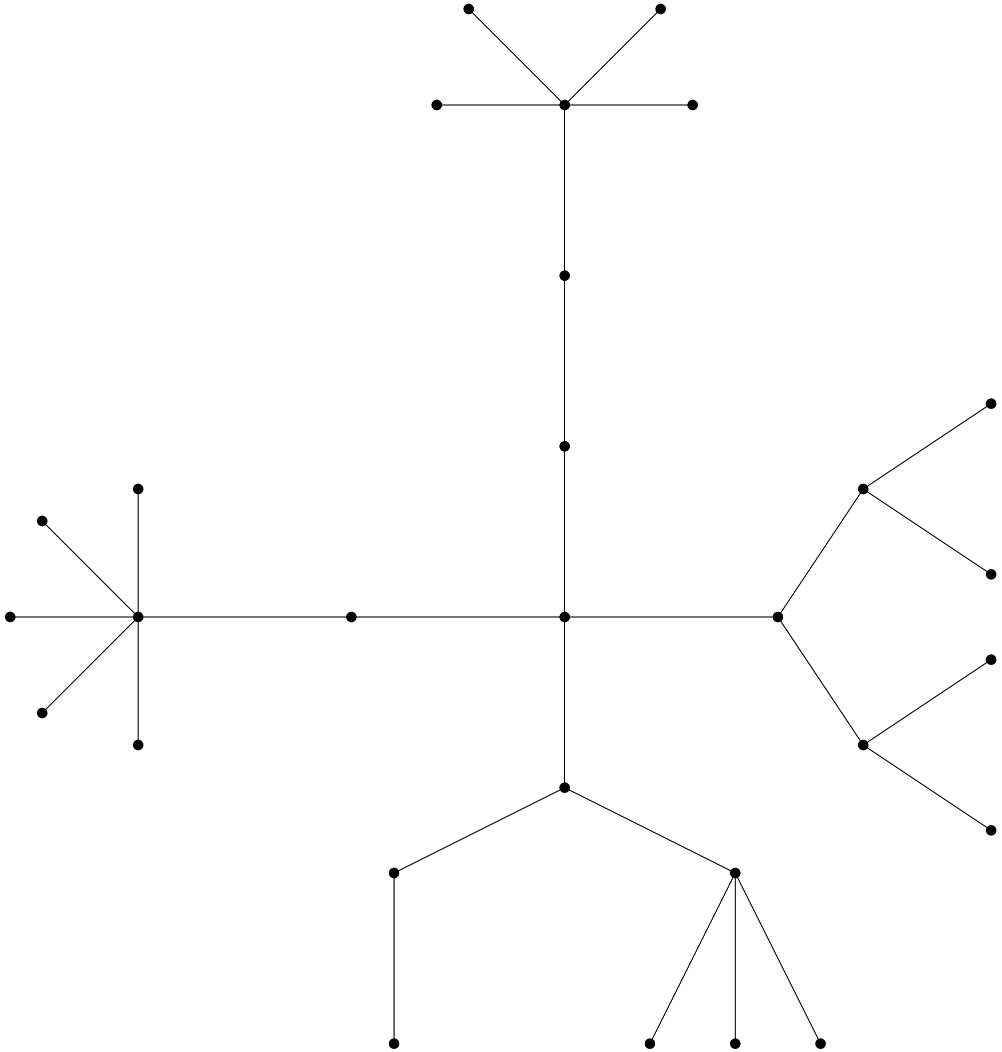
$$f(u,v) := (d(u) - j)s_u(v) + \boxed{\sum_{i=1}^{j} s_u(u,i),}$$

The size of the j largest subtrees of T^u.

where v is the jth largest child of u. We use a data structure similar to an adjacency list as in the calculation of `Projective minLA` and `Planar minLA`.

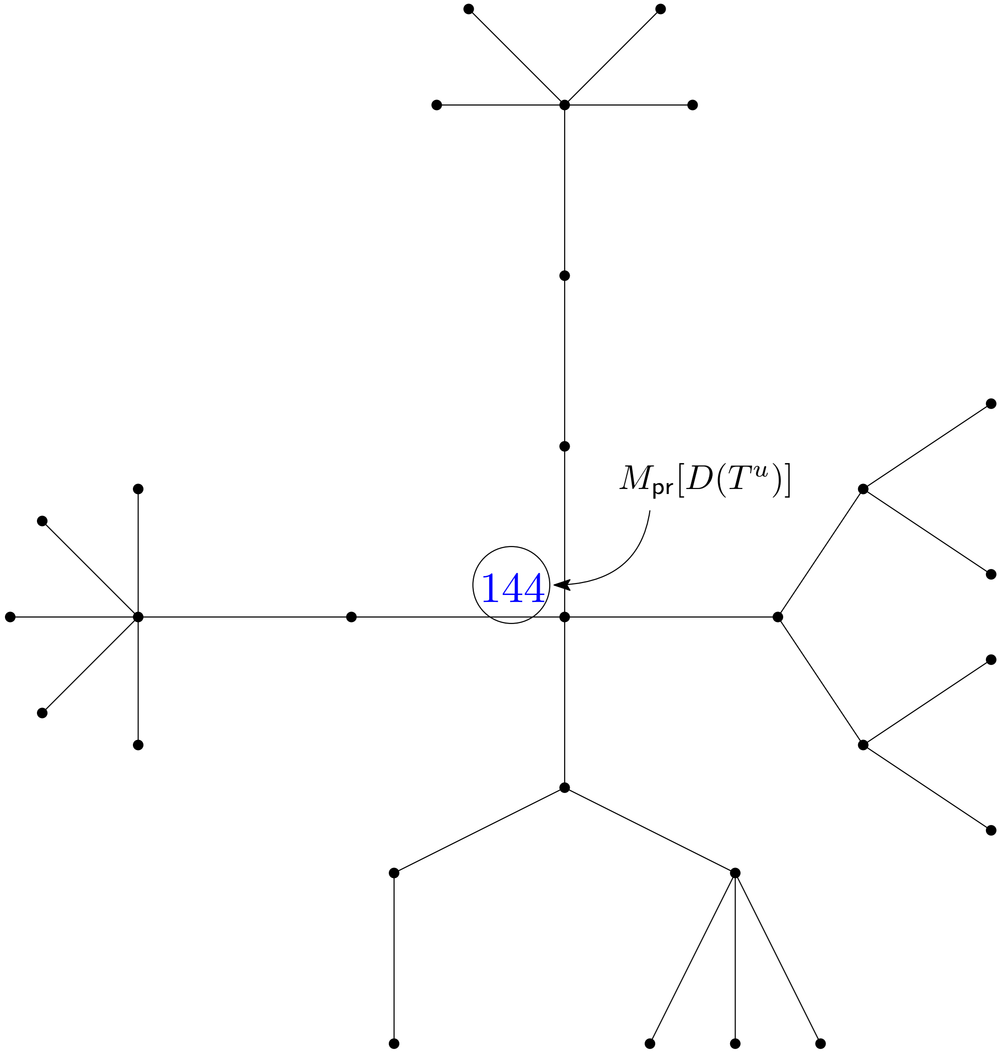
Maximum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400



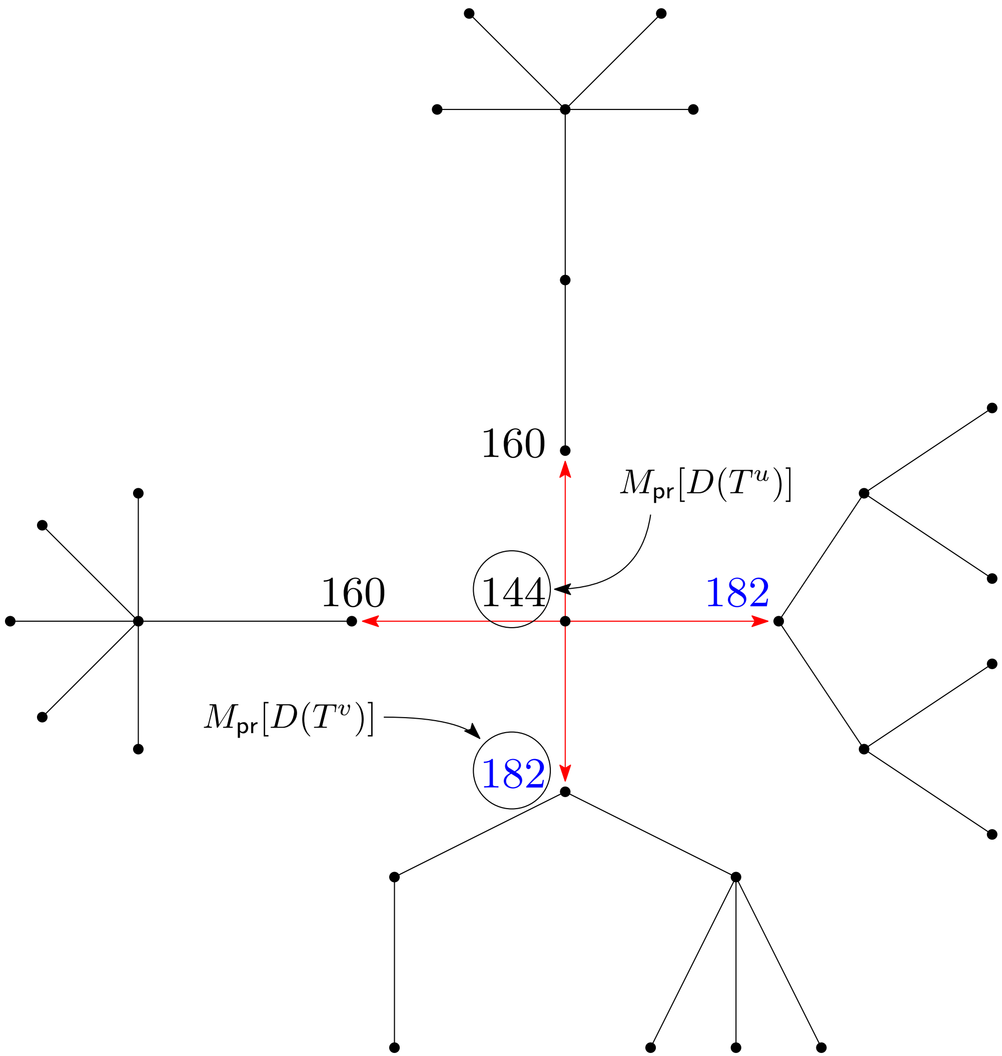
Maximum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400



Maximum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

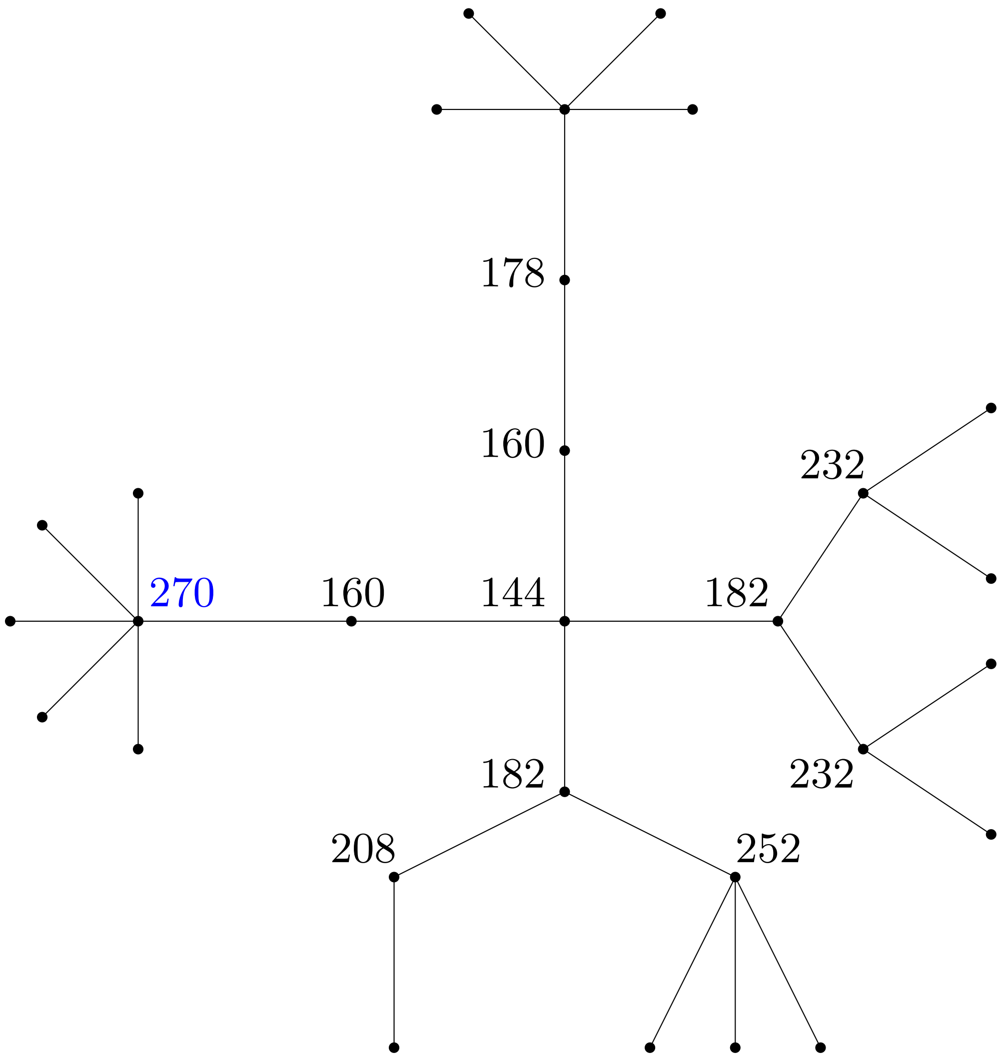


$$M_{\mathsf{pr}}[D(T^v)] = M_{\mathsf{pr}}[D(T^u)] + f(v,u) - f(u,v),$$

$$f(u,v) := (d(u) - j)s_u(v) + \sum_{i=1}^{j} s_u(u,i)$$

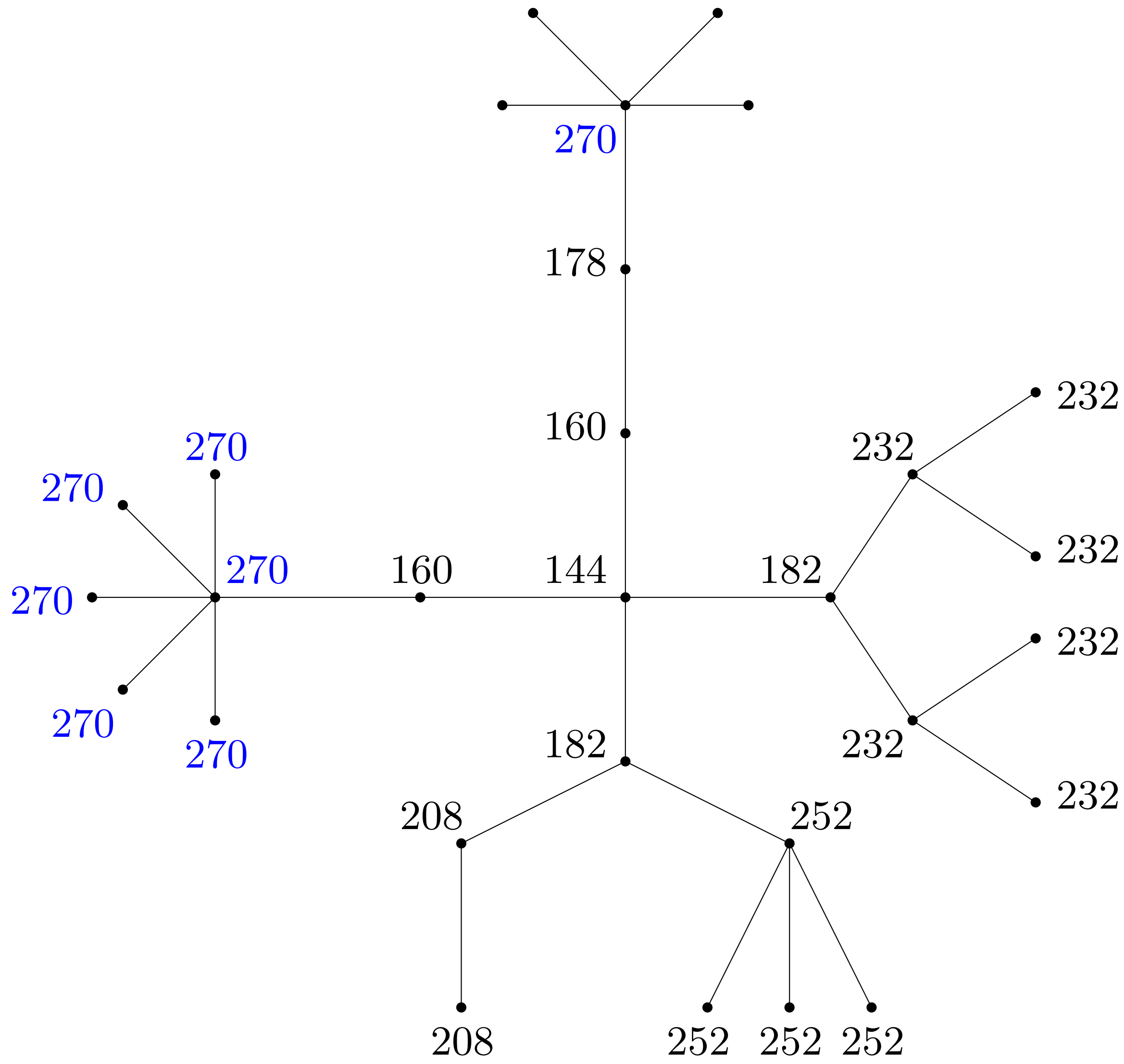
Maximum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400



Maximum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

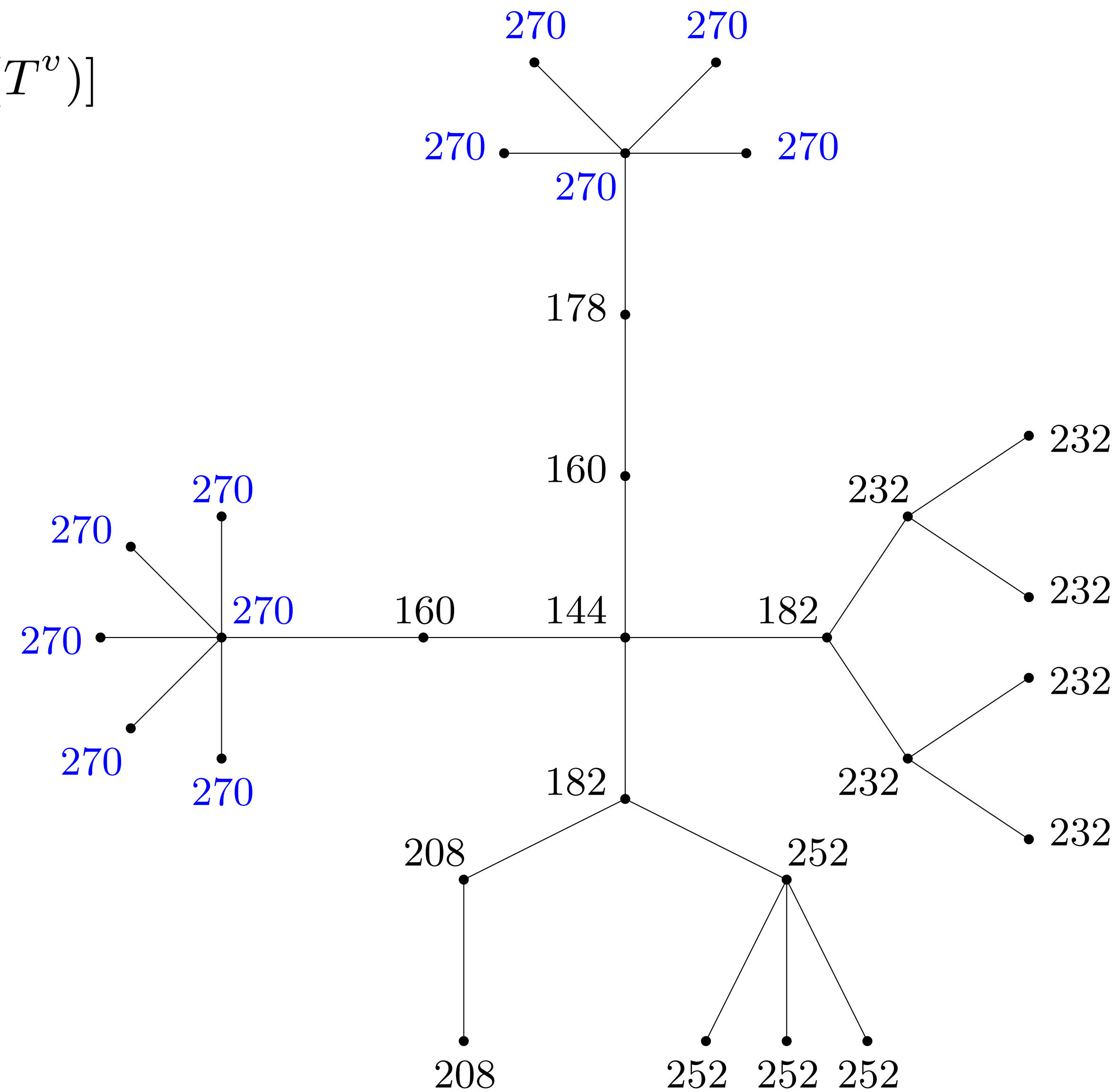


Maximum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

For any leaf u and its only neighbor v,

$$M_{\text{pr}}[D(T^u)] = M_{\text{pr}}[D(T^v)]$$



Maximum arrangements – Planar case

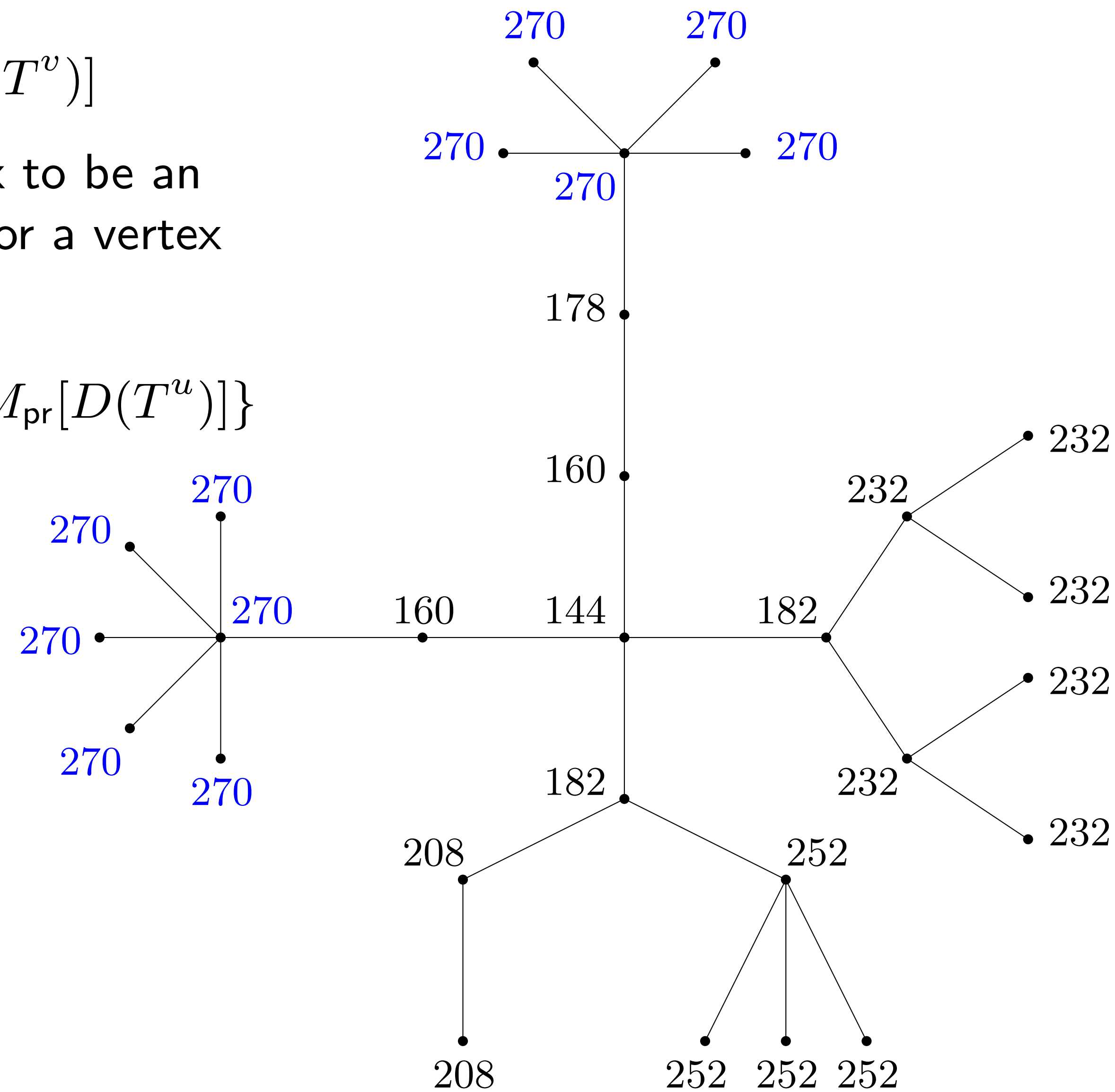
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

For any leaf u and its only neighbor v,

$$M_{\mathsf{pr}}[D(T^u)] = M_{\mathsf{pr}}[D(T^v)]$$

Necessary condition for a vertex to be an optimal root: it is either a leaf or a vertex adjacent to a leaf.

$$M_{\mathsf{pl}}[D(T)] = \max_{u \text{ adjacent to a leaf}} \{M_{\mathsf{pr}}[D(T^u)]\}$$



minLA/MaxLA – Projective and Planar – Common structure

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2024). The maximum linear arrangement problem for trees under projectivity and planarity. *Information Processing Letters*, *183*, 106400.
https://doi.org/10.1016/j.ipl.2023.106400

1. Find 'optimal' root vertex v for the tree. Projective cases: v is the root of the tree. In `Planar minLA`, v is a centroidal vertex (Iordanskii, 1987; Hochberg & Stallmann, 2003). In `Planar MaxLA`, v has to be a leaf or have a leaf attached.
2. Calculate all subtree sizes with respect to v.
3. In a top-down fashion, first set $u := v$,
   (a) Sort the subtrees of T_u^v by their size.
   (b) Find an optimal position for u, the root of the current subtree. In `Projective minLA`, the subtrees are arranged to both sides of the root in a balanced manner (Hochberg & Stallmann, 2003; A.-P., Esteban, & Ferrer-i-Cancho, 2022). In `Projective MaxLA`, the subtrees are placed to the left (or to the right) of the root.
   (c) Compute the interval $[a, b]$ in which each subtree is to be arranged.
   (d) Recursively apply these steps.

Maximum arrangements

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

A 2-approximation algorithm is known (Hassin & Rubinstein, 2001). No polynomial time algorithm is known for trees, but solutions to several classes of trees are known (Ferrer-i-Cancho, Gómez-Rodríguez, & Esteban, 2021). DeVos and Nurse (2018) and Nurse (2019) discovered three cornerstone characterizations of maximum arrangements of graphs, and devised an algorithm to solve `MaxLA` for trees in time $O\left(n^{4\Delta}\right)$.

Maximum arrangements

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

A 2-approximation algorithm is known (Hassin & Rubinstein, 2001). No polynomial time algorithm is known for trees, but solutions to several classes of trees are known (Ferrer-i-Cancho, Gómez-Rodríguez, & Esteban, 2021). DeVos and Nurse (2018) and Nurse (2019) discovered three cornerstone characterizations of maximum arrangements of graphs, and devised an algorithm to solve `MaxLA` for trees in time $O\left(n^{4\Delta}\right)$.

Main contributions

- Complemented existing properties of maximum arrangements of graphs.
- Identified two key variants of `MaxLA`
  - `1-thistle MaxLA`: time $O\left(n^3 2^{\Delta}\right)$. Typically over n-vertex trees: $O\left(n^4\right)$.
  - `Bipartite MaxLA`: time $O(n)$.
- Empirical results checked (all trees $n \leq 24$, random sampling $25 \leq n \leq 48$) with a Branch & Bound algorithm. Fast enough for $n \leq 35$.

Maximum arrangements

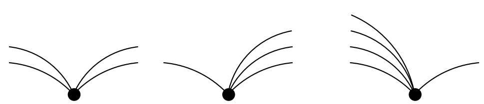
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

A 2-approximation algorithm is known (Hassin & Rubinstein, 2001). No polynomial time algorithm is known for trees, but solutions to several classes of trees are known (Ferrer-i-Cancho, Gómez-Rodríguez, & Esteban, 2021). DeVos and Nurse (2018) and Nurse (2019) discovered three cornerstone characterizations of maximum arrangements of graphs, and devised an algorithm to solve `MaxLA` for trees in time $O\left(n^{4\Delta}\right)$.

Main contributions

- Complemented existing properties of maximum arrangements of graphs.
- Identified two key variants of `MaxLA`
  - `1-thistle MaxLA`: time $O\left(n^3 2^{\Delta}\right)$. Typically over n-vertex trees: $O\left(n^4\right)$.

    Vertex u is a *thistle* in π if its neighbors are distributed to both sides of u in π.

  - `Bipartite MaxLA`: time $O(n)$.
- Empirical results checked (all trees $n \leq 24$, random sampling $25 \leq n \leq 48$) with a Branch & Bound algorithm. Fast enough for $n \leq 35$.

Maximum arrangements

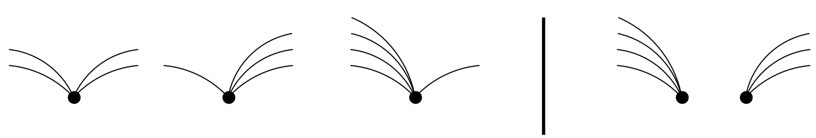
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

A 2-approximation algorithm is known (Hassin & Rubinstein, 2001). No polynomial time algorithm is known for trees, but solutions to several classes of trees are known (Ferrer-i-Cancho, Gómez-Rodríguez, & Esteban, 2021). DeVos and Nurse (2018) and Nurse (2019) discovered three cornerstone characterizations of maximum arrangements of graphs, and devised an algorithm to solve `MaxLA` for trees in time $O\left(n^{4\Delta}\right)$.

Main contributions

- Complemented existing properties of maximum arrangements of graphs.
- Identified two key variants of `MaxLA`
  - `1-thistle MaxLA`: time $O\left(n^3 2^{\Delta}\right)$. Typically over n-vertex trees: $O\left(n^4\right)$.

    Vertex u is a *thistle* in π if its neighbors are distributed to both sides of u in π.

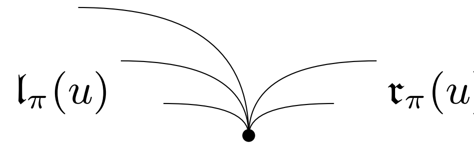
    

  - `Bipartite MaxLA`: time $O(n)$.
- Empirical results checked (all trees $n \leq 24$, random sampling $25 \leq n \leq 48$) with a Branch & Bound algorithm. Fast enough for $n \leq 35$.

Maximum arrangements – Known properties

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Known properties of maximum (unconstrained) arrangements *of graphs* due to DeVos and Nurse (2018) and Nurse (2019) based on the concept of *vertex level*.



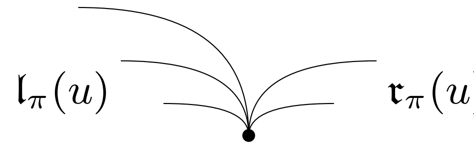
$$\begin{aligned} l_\pi(u) &= \mathfrak{r}_\pi(u) - \mathfrak{l}_\pi(u) \\ &= 2 - 3 \\ &= -1 \end{aligned}$$

Maximum arrangements – Known properties

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Known properties of maximum (unconstrained) arrangements *of graphs* due to DeVos and Nurse (2018) and Nurse (2019) based on the concept of *vertex level*.

Given an arrangement $\pi = (u_1, \dots, u_i, \dots, u_n)$:



$$l_\pi(u) = \mathfrak{r}_\pi(u) - \mathfrak{l}_\pi(u) = 2 - 3 = -1$$

Property 1 (Necessary condition) The sequence of level values is non-increasing.

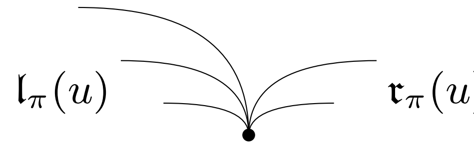
$$l_\pi(u_1) \geq \cdots \geq l_\pi(u_i) \geq \cdots \geq l_\pi(u_n)$$

Maximum arrangements – Known properties

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Known properties of maximum (unconstrained) arrangements *of graphs* due to DeVos and Nurse (2018) and Nurse (2019) based on the concept of *vertex level*.

Given an arrangement $\pi = (u_1, \ldots, u_i, \ldots, u_n)$:



$$
\begin{aligned}
l_\pi(u) &= \mathfrak{r}_\pi(u) - \mathfrak{l}_\pi(u) \\
&= 2 - 3 \\
&= -1
\end{aligned}
$$

Property 1 (Necessary condition) The sequence of level values is non-increasing.

$$l_\pi(u_1) \geq \cdots \geq l_\pi(u_i) \geq \cdots \geq l_\pi(u_n)$$

Property 2 (Necessary condition) There cannot be two vertices $uv \in E$ such that

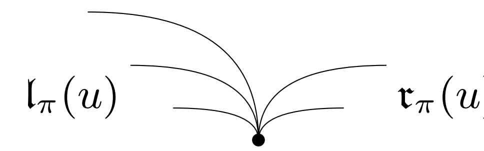
$$l_\pi(u) = l_\pi(v)$$

Maximum arrangements – Known properties

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Known properties of maximum (unconstrained) arrangements *of graphs* due to DeVos and Nurse (2018) and Nurse (2019) based on the concept of *vertex level*.

Given an arrangement $\pi = (u_1, \dots, u_i, \dots, u_n)$:



$$\begin{aligned} l_\pi(u) &= \mathfrak{r}_\pi(u) - \mathfrak{l}_\pi(u) \\ &= 2 - 3 \\ &= -1 \end{aligned}$$

Property 1 (Necessary condition) The sequence of level values is non-increasing.

$$l_\pi(u_1) \geq \dots \geq l_\pi(u_i) \geq \dots \geq l_\pi(u_n)$$

Property 2 (Necessary condition) There cannot be two vertices $uv \in E$ such that

$$l_\pi(u) = l_\pi(v)$$

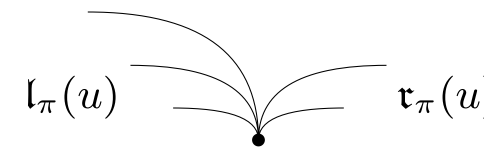
Property 3 (Symmetry breaking) Vertices of equal level value can be permuted arbitrarily.

Maximum arrangements – Known properties

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Known properties of maximum (unconstrained) arrangements *of graphs* due to DeVos and Nurse (2018) and Nurse (2019) based on the concept of *vertex level*.

Given an arrangement $\pi = (u_1, \dots, u_i, \dots, u_n)$:



$$
\begin{aligned}
l_\pi(u) &= \mathfrak{r}_\pi(u) - \mathfrak{l}_\pi(u) \\
&= 2 - 3 \\
&= -1
\end{aligned}
$$

Property 1 (Necessary condition) The sequence of level values is non-increasing.

$$l_\pi(u_1) \geq \cdots \geq l_\pi(u_i) \geq \cdots \geq l_\pi(u_n)$$

Property 2 (Necessary condition) There cannot be two vertices $uv \in E$ such that

$$l_\pi(u) = l_\pi(v)$$

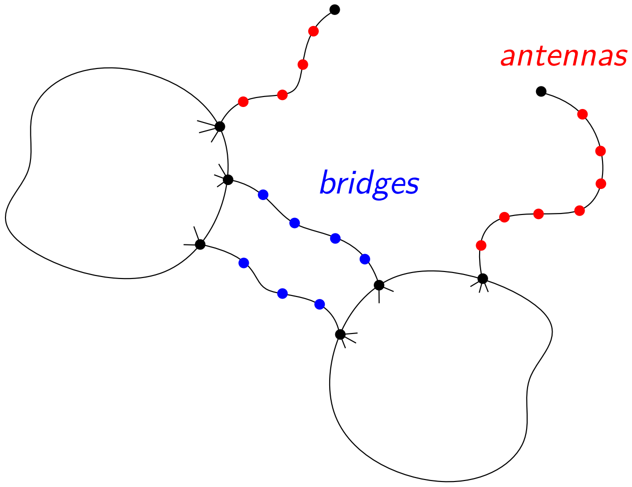
Property 3 (Symmetry breaking) Vertices of equal level value can be permuted arbitrarily.

These properties do not tell *a priori* the level value that a vertex should have in a maximum arrangement.

Maximum arrangements – New property

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

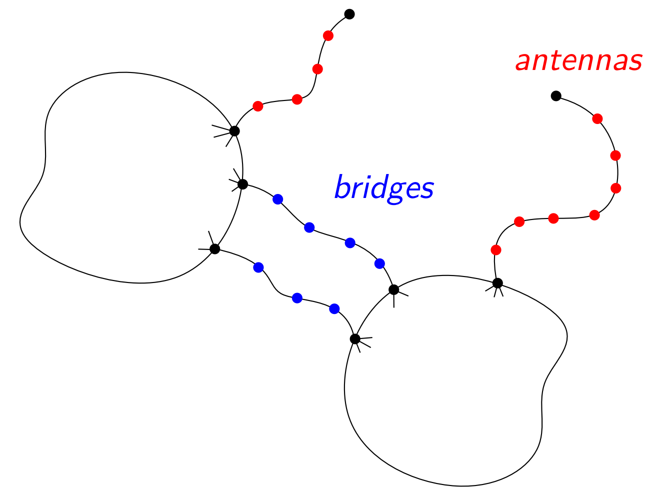
Path Optimization Lemma: a new property that limits the level values that the vertices in *path subgraphs* can have in a maximum arrangement of any simple graph G.



Maximum arrangements – New property

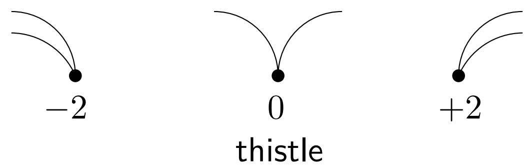
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Path Optimization Lemma: a new property that limits the level values that the vertices in *path subgraphs* can have in a maximum arrangement of any simple graph G.



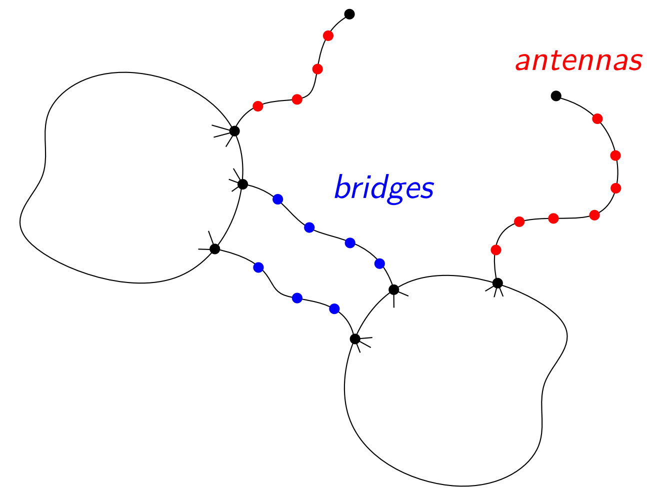
Vertex u is a *thistle* in π if $|l_\pi(u)| < d(u)$.

For vertices of degree 2:



Maximum arrangements – New property

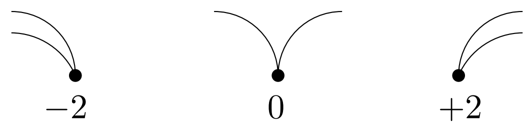
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487



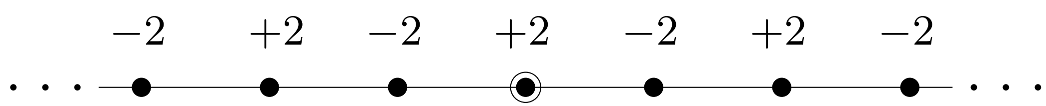
Path Optimization Lemma: a new property that limits the level values that the vertices in *path subgraphs* can have in a maximum arrangement of any simple graph G.

Vertex u is a *thistle* in π if $|l_\pi(u)| < d(u)$.

For vertices of degree 2:

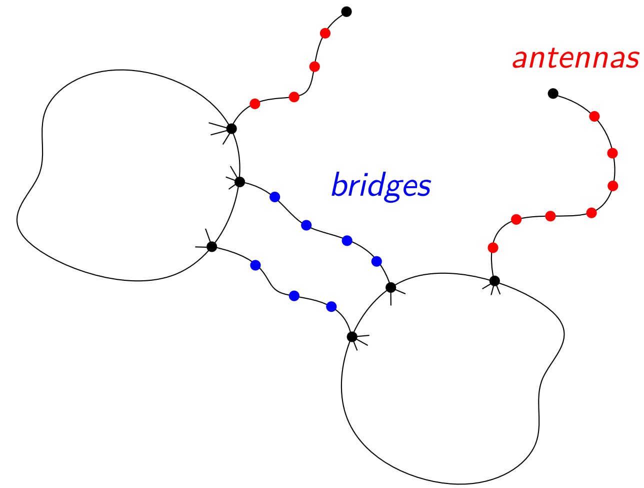


(i) Vertices in antennas cannot be thistles $\rightarrow$ alternation of level values in a maximum arrangement



Maximum arrangements – New property

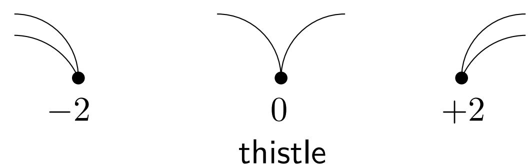
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487



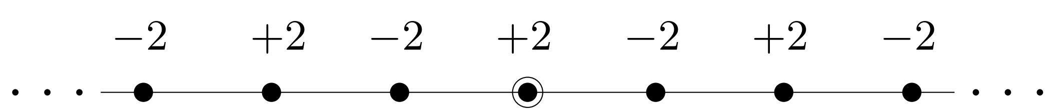
Path Optimization Lemma: a new property that limits the level values that the vertices in *path subgraphs* can have in a maximum arrangement of any simple graph G.

Vertex u is a *thistle* in π if $|l_\pi(u)| < d(u)$.

For vertices of degree 2:



(i) Vertices in antennas cannot be thistles $\rightarrow$ alternation of level values in a maximum arrangement



(ii) At most one vertex in bridges can be a thistle, and *any* vertex can be a thistle.

Speed up the Branch & Bound algorithm.

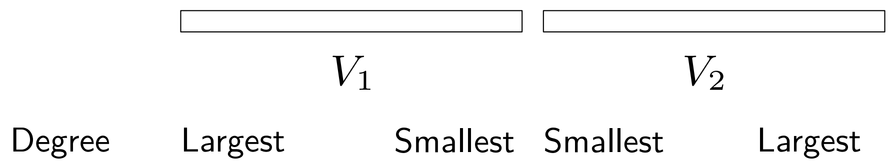
- Propagation of level values (in bridges and antennas).
- Fix one vertex for each bridge to be the only (allowed) thistle.

And solve `MaxLA` for specific classes of trees.

Maximum arrangements – Bipartite MaxLA

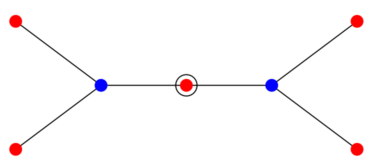
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

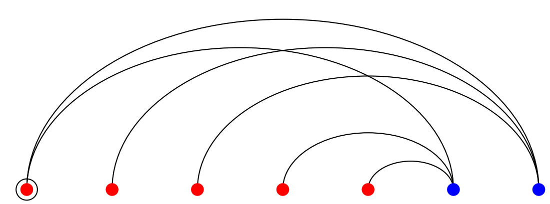
Approximating via `Bipartite MaxLA`: find a maximal bipartite arrangement of a (connected) bipartite graph $B = (V_1 \cup V_2, E)$.



Function $M_{\text{bip}}(B)$ **is**

- **In:** B a connected bipartite graph.
- **Out:** A maximal bipartite arrangement B.
- Sort vertices in V_1 and V_2 by degree. (Counting sort does the trick!)
- Arrange the vertices in V_1 by non-increasing degree.
- Arrange the vertices in V_2 by non-decreasing degree.





$D = 23$

Maximum arrangements – Bipartite MaxLA

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Are maximal bipartite arrangements actually useful?

Corollary For any tree T

$$\frac{M[D(T)]}{M_{\mathsf{bip}}[D(T)]} \leq \frac{3}{2}.$$

Improvement (in trees) over previous work where the approximation factor given is 2 in general graphs (Hassin & Rubinstein, 2001).

Maximum arrangements – Bipartite MaxLA

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Are maximal bipartite arrangements actually useful?

Corollary For any tree T

$$\frac{M[D(T)]}{M_{\mathsf{bip}}[D(T)]} \leq \frac{3}{2}.$$

Improvement (in trees) over previous work where the approximation factor given is 2 in general graphs (Hassin & Rubinstein, 2001).

Let p_n be the proportion of n-vertex trees T such that $M[D(T)] = M_{\mathsf{bip}}[D(T)]$.

Maximum arrangements – Bipartite MaxLA

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Are maximal bipartite arrangements actually useful?

Corollary For any tree T

$$\frac{M[D(T)]}{M_{\mathsf{bip}}[D(T)]} \leq \frac{3}{2}.$$

Improvement (in trees) over previous work where the approximation factor given is 2 in general graphs (Hassin & Rubinstein, 2001).

Let p_n be the proportion of n-vertex trees T such that $M[D(T)] = M_{\mathsf{bip}}[D(T)]$.

Conjecture The value of p_n as n tends to infinity is large

$$\lim_{n\to\infty} p_n = c, \qquad c \approx 0.5.$$

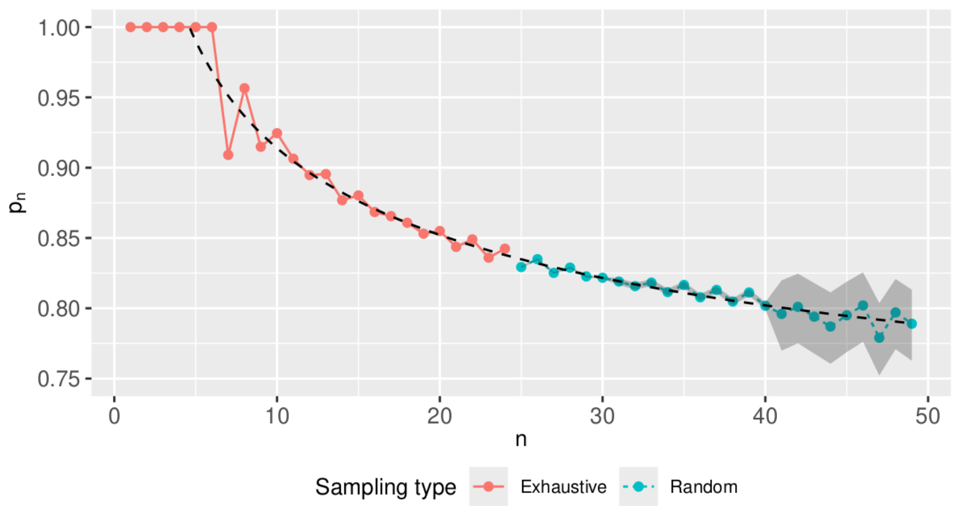
Conjecture p_n decays slowly: $p_n = \Theta\big(n^{-b}\big)$ for some $b \in (0,1)$.

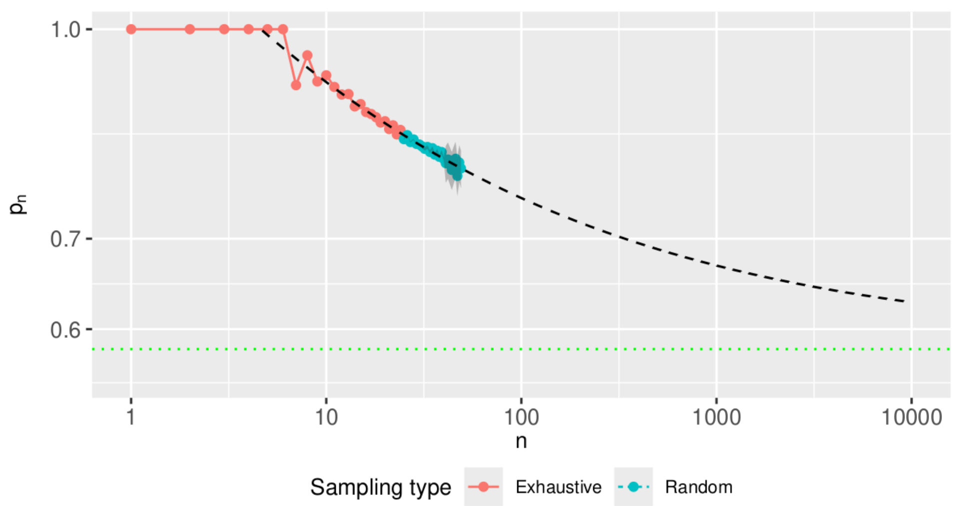
Maximum arrangements – Bipartite MaxLA

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Are maximal bipartite arrangements actually useful?

How often can `Bipartite MaxLA` solve `MaxLA`?





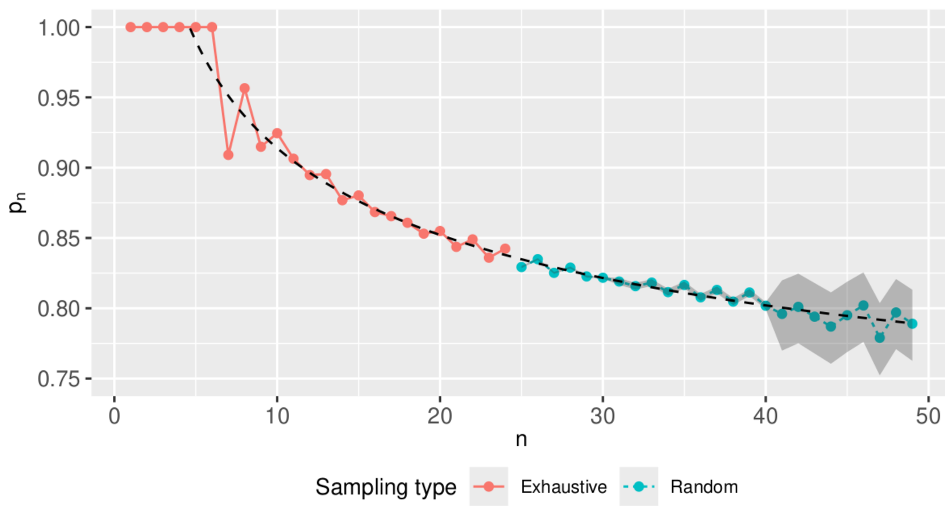
(log-log scale)

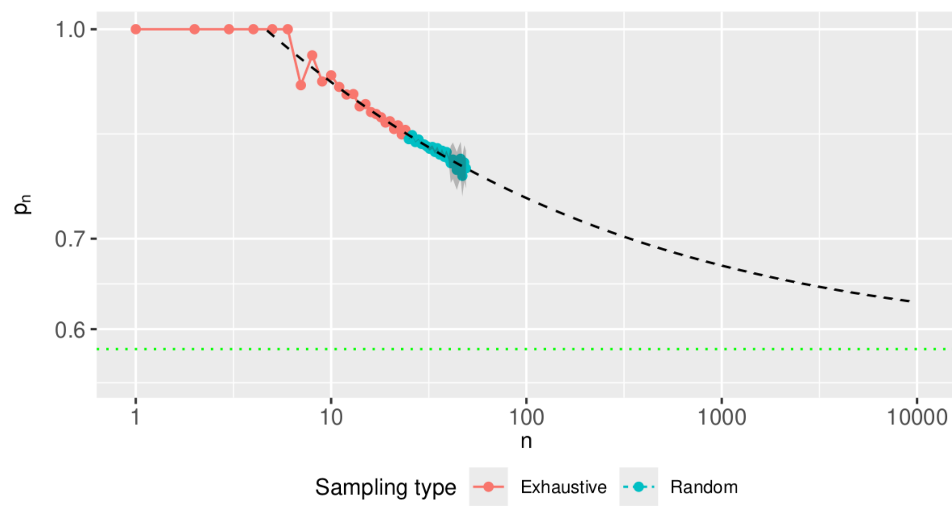
Maximum arrangements – Bipartite MaxLA

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Are maximal bipartite arrangements actually useful?

How often can `Bipartite MaxLA` solve `MaxLA`?





(log-log scale)

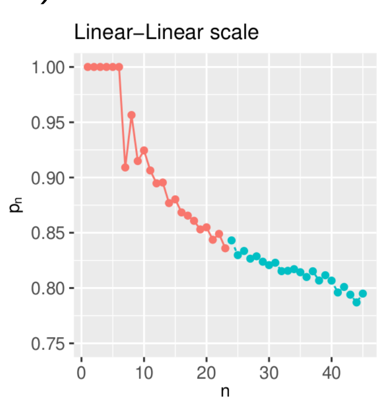
Best model (power law)

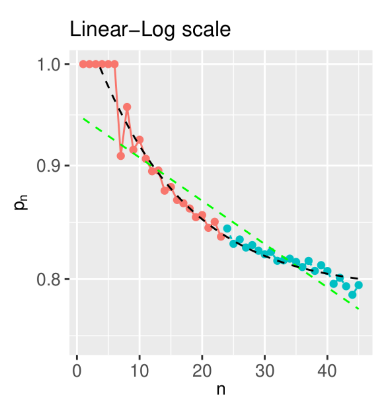
$$p_n = an^b + c,$$

$$a = 0.655$$

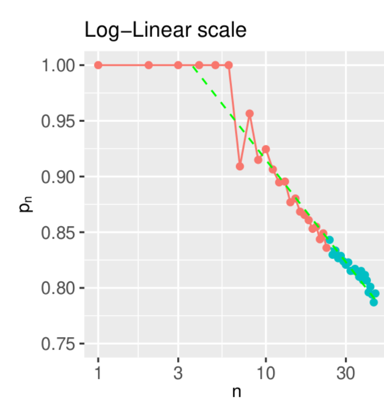
$$b = -0.303$$

$$c = 0.588$$

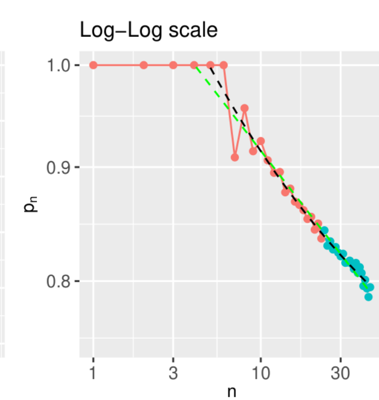




Exponential model



Logarithmic model



Power law
(*Best model)

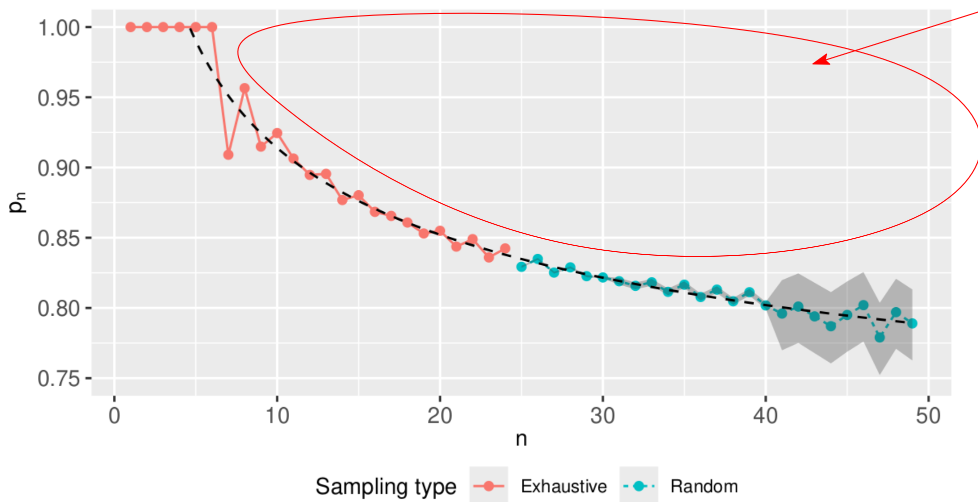
Maximum arrangements – Bipartite MaxLA

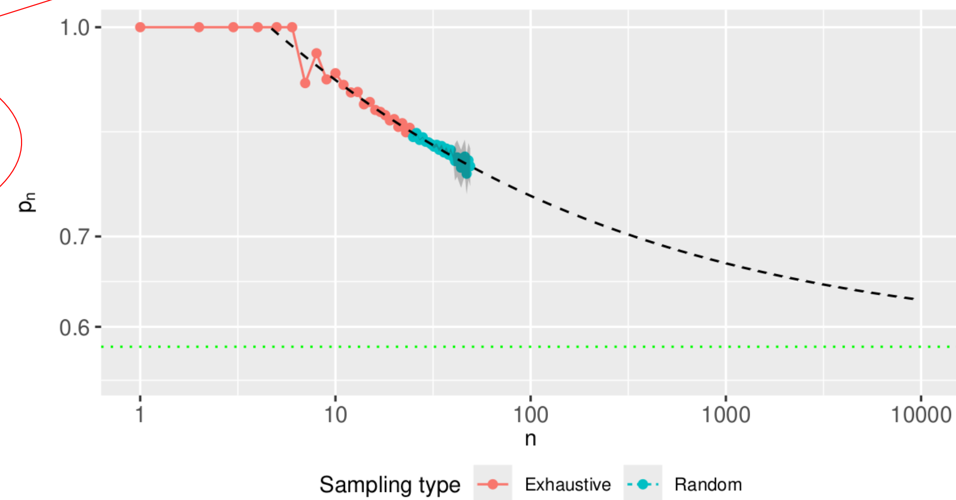
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

Are maximal bipartite arrangements actually useful?

How often can `Bipartite MaxLA` solve `MaxLA`?

Most of these trees have at least one maximum arrangement with exactly *one* thistle vertex





(log-log scale)

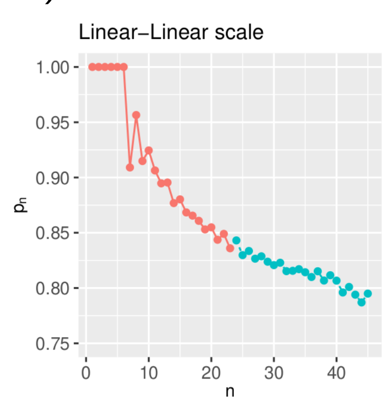
Best model (power law)

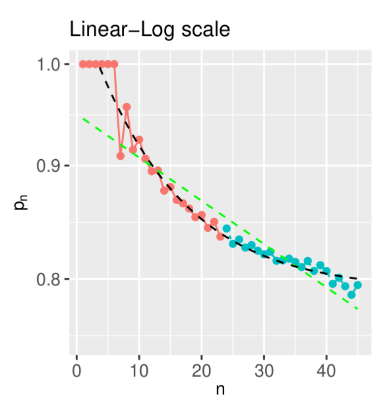
$$p_n = an^b + c,$$

$$a = 0.655$$

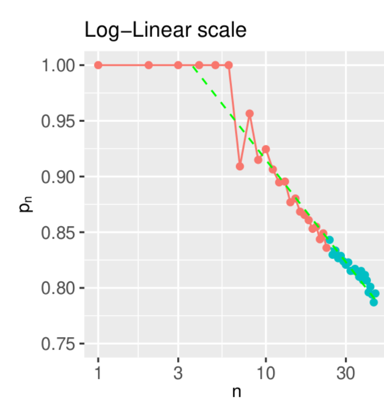
$$b = -0.303$$

$$c = 0.588$$

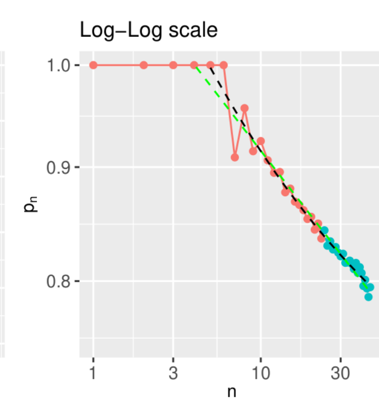




Exponential model



Logarithmic model

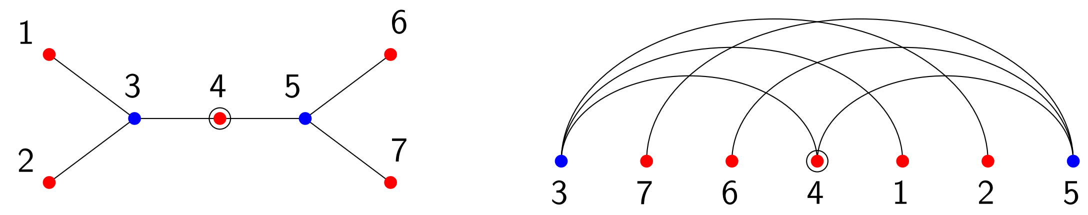


Power law
(*Best model)

Maximum arrangements – 1-thistle MaxLA

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

We continue approximating maximum arrangements with maximal non-bipartite arrangements with exactly one thistle. Our algorithm has time complexity $O(n^3 2^\Delta)$ (where Δ is the maximum degree of the tree).

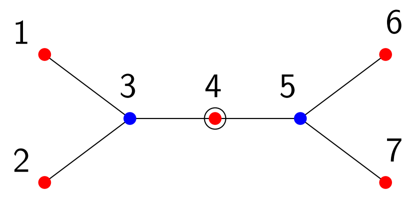


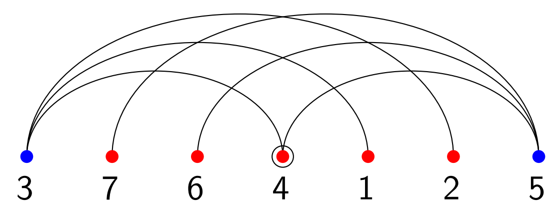
$D = 24$

Maximum arrangements – 1-thistle MaxLA

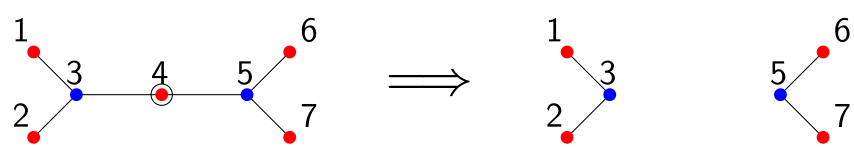
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

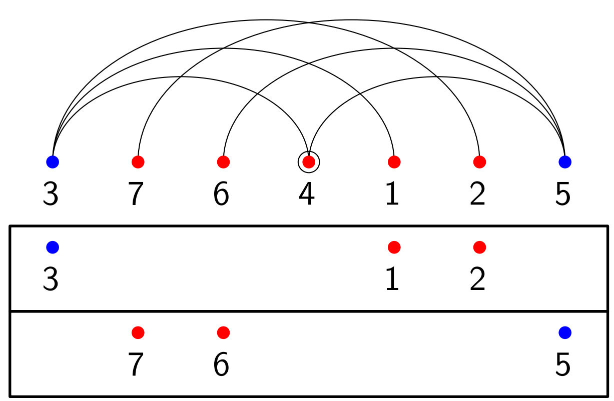
We continue approximating maximum arrangements with maximal non-bipartite arrangements with exactly one thistle. Our algorithm has time complexity $O(n^3 2^\Delta)$ (where Δ is the maximum degree of the tree).





$D = 24$

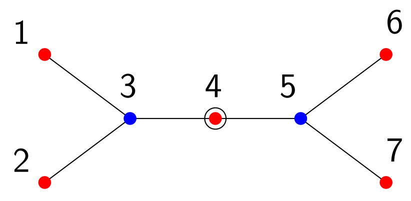


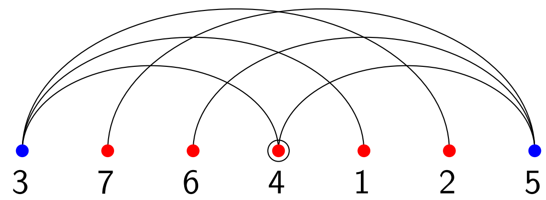


Maximum arrangements – 1-thistle MaxLA

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

We continue approximating maximum arrangements with maximal non-bipartite arrangements with exactly one thistle. Our algorithm has time complexity $O(n^3 2^{\Delta})$ (where Δ is the maximum degree of the tree).



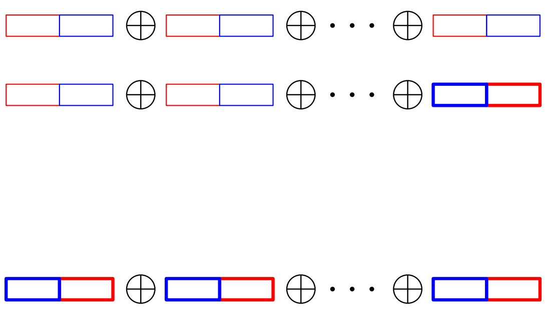


$D = 24$

Overview of 1-thistle MaxLA of a tree T: for every vertex v

- Take a vertex v and let $T_1, \ldots, T_d$ be the subtrees of v.
- Construct maximal bipartite arrangements of the subtrees $\pi_1, \ldots, \pi_d$.
- Try all 2^d combinations of the bipartite arrangements.

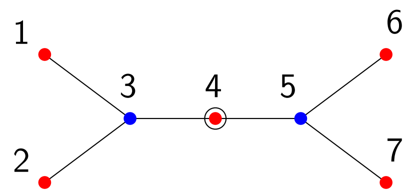
$$\pi_1 \oplus \cdots \oplus \pi_{d-1} \oplus \pi_d$$
$$\pi_1 \oplus \cdots \oplus \pi_{d-1} \oplus \tilde{\pi}_d$$
$$\vdots$$
$$\tilde{\pi}_1 \oplus \cdots \oplus \tilde{\pi}_{d-1} \oplus \tilde{\pi}_d$$

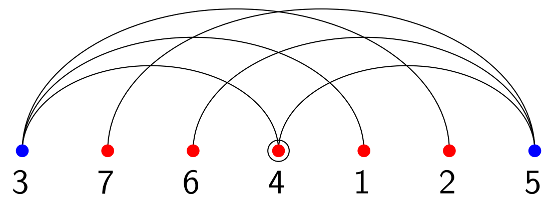


Maximum arrangements – 1-thistle MaxLA

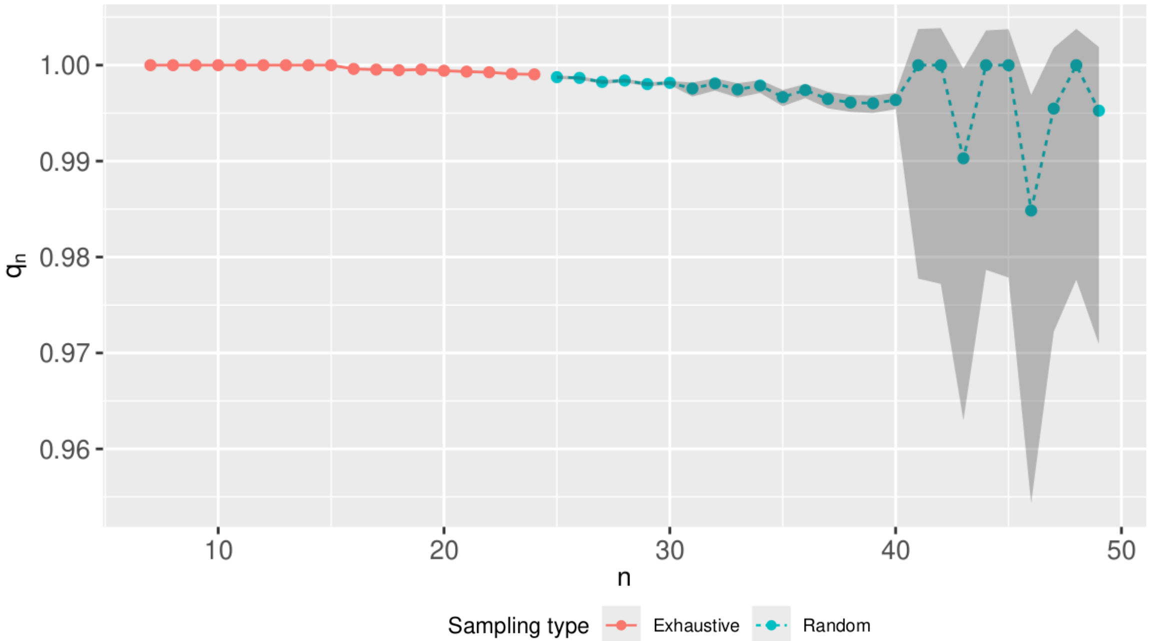
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

We continue approximating maximum arrangements with maximal non-bipartite arrangements with exactly one thistle. Our algorithm has time complexity $O(n^3 2^\Delta)$ (where Δ is the maximum degree of the tree).





How often `1-thistle MaxLA` solves `MaxLA` (when `Bipartite MaxLA` cannot solve it)?

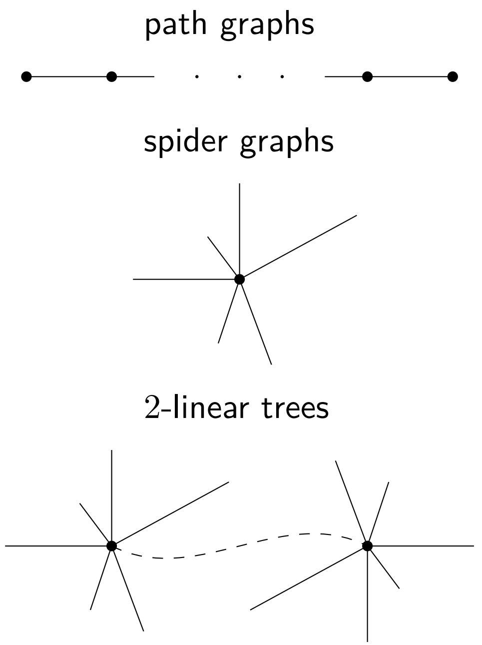


Maximum arrangements – Classes of trees

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

We applied POL to prove that `MaxLA` can be solved in time $O(n)$ for

- path graphs (0-linear trees)
- spider graphs (1-linear trees)
- 2-linear trees.

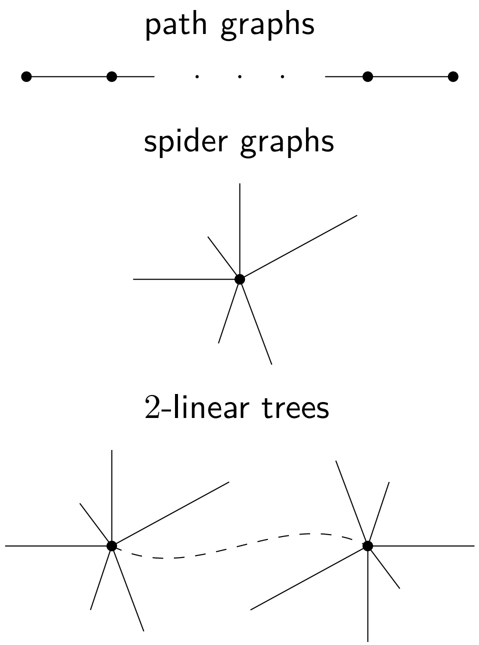


Maximum arrangements – Classes of trees

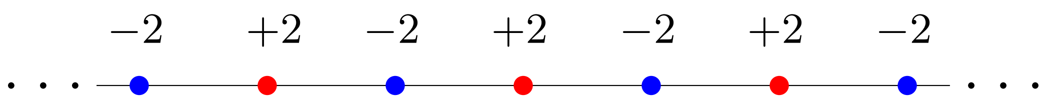
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

We applied POL to prove that $\texttt{MaxLA}$ can be solved in time $O(n)$ for

- path graphs (0-linear trees)
- spider graphs (1-linear trees)
- 2-linear trees.



Path graphs are just big anntenas: its vertices can only have level values $\pm 2, \pm 1$. Then *all* its maximum arrangements are bipartite.

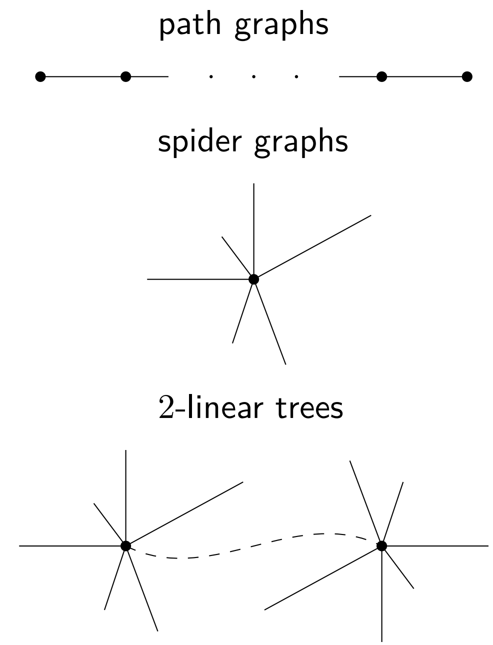


Maximum arrangements – Classes of trees

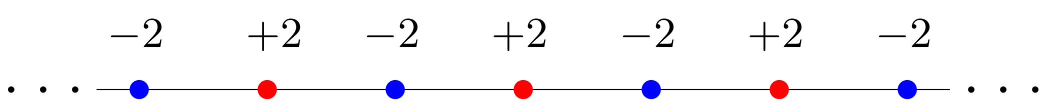
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

We applied POL to prove that `MaxLA` can be solved in time $O(n)$ for

- path graphs (0-linear trees)
- spider graphs (1-linear trees)
- 2-linear trees.



Path graphs are just big anntenas: its vertices can only have level values $\pm 2, \pm 1$. Then *all* its maximum arrangements are bipartite.



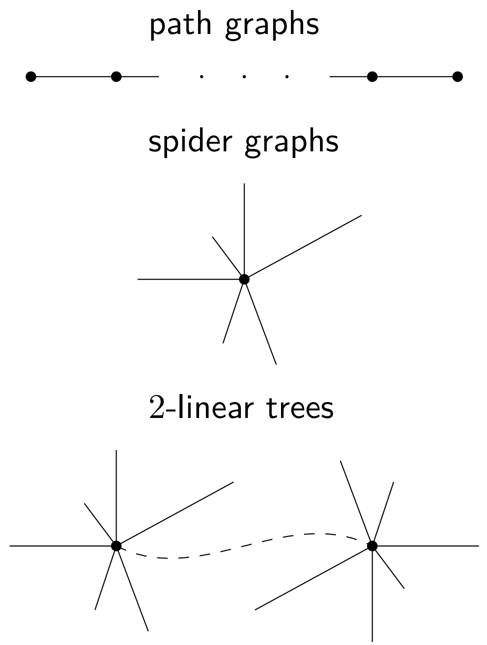
Spider graphs can be seen as antennas joined at one end: the only possible thistle is the hub vertex. We showed that the hub cannot be a thistle in a maximum arrangement. Then *all* its maximum arrangements are bipartite.

Maximum arrangements – Classes of trees

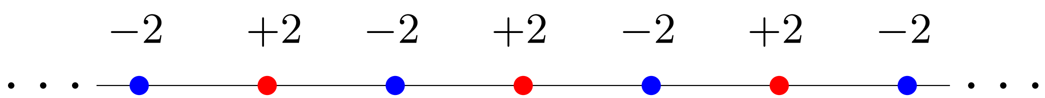
A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2023). On The Maximum Linear Arrangement Problem for Trees. *arXiv*. https://arxiv.org/abs/2312.04487

We applied POL to prove that `MaxLA` can be solved in time $O(n)$ for

- path graphs (0-linear trees)
- spider graphs (1-linear trees)
- 2-linear trees.



Path graphs are just big anntenas: its vertices can only have level values $\pm 2, \pm 1$. Then *all* its maximum arrangements are bipartite.



Spider graphs can be seen as antennas joined at one end: the only possible thistle is the hub vertex. We showed that the hub cannot be a thistle in a maximum arrangement. Then *all* its maximum arrangements are bipartite.

`MaxLA` can be solved in time $O(n)$ for 2-linear trees after proving that none of its hubs can be a thistle. Then, take the maximum between `Bipartite MaxLA` and `1-thistle MaxLA`.

Software

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2021). The Linear Arrangement Library. A new tool for research on syntactic dependency structures. *Proceedings of the Second Workshop on Quantitative Syntax (Quasy, SyntaxFest 2021)*, 1–16. https://aclanthology.org/2021.quasy-1.1

A piece of code using some of LAL's features

```
import lal
n = 10
t = lal.graphs.free_tree(n)
t.set_edges([(u, u+1) for u in range(0,n-1)])

DMax, arrs = lal.linarr.max_sum_edge_lengths_all(t, n_threads)

algo_HS = lal.linarr.algorithms_Dmin_planar.HochbergStallmann
algo_AEF = lal.linarr.algorithms_Dmin_planar.AlemanyEstebanFerrer
Dmin_plan, arr_plan = lal.linarr.min_sum_edge_lengths_planar(t, algo_*)

E_D_proj = lal.properties.exp_sum_edge_lengths_projective(t)

gen_plan = lal.generate.rand_planar_arrangements(t)

tr = lal.graphs.rooted_tree(t, r)
gen_proj = lal.generate.all_projective_arrangements(tr)

# and much more ...
```

Conclusions

All algorithms and formulas are available in LAL: tested, open and accessible.

We devised formulas and algorithms to calculate exact expected values. These replace uniform random sampling methods.

Clarified, succintly enough, inaccuracies and mistakes in previous research involving `Projective/Planar minLA`.

Linear-time algorithms for `Projective/Planar MaxLA`.

Obvious common structure between `Projective/Planar minLA` and `Projective/Planar MaxLA`: subtrees are organized in a disjoint manner and optimized locally. `Projective/Planar MaxLA` have been solved optimally in time and space $O(n)$.

Conclusions

Most of our work concerning `MaxLA` is based on Nurse and De Vos theoretical apparatus, which we extended with the Path Optimization Lemma:

- Better understanding of the structure of maximum arrangements.
- Easily find solutions for 0-, 1-, and 2-linear trees.

`MaxLA` is still not solved for trees (it could be **NP**-Hard) but

- It is approximated quite well,
  - `Bipartite MaxLA` is solved in time $O(n)$, but
  - `1-thistle MaxLA` has exponential cost in the worst case.
- Our adaptation of the B&B algorithm works well enough in practice for sentences of average length – 23 words on average for most languages (Ferrer-i-Cancho, Gómez-Rodríguez, & Esteban, 2018, Table 5).

Thank you all!

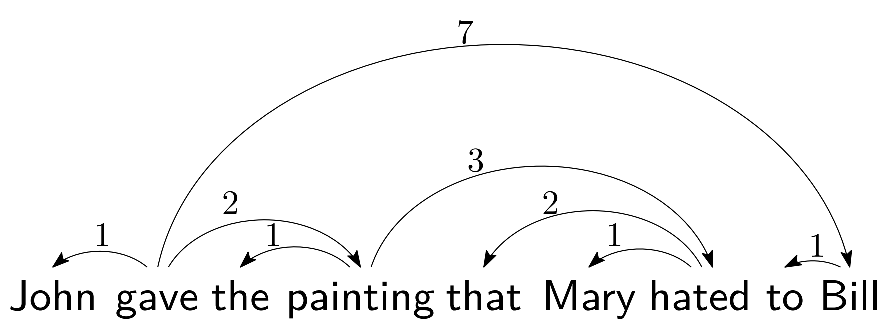
QUESTIONS

Application – Quantitative Dependency Syntax – DDm

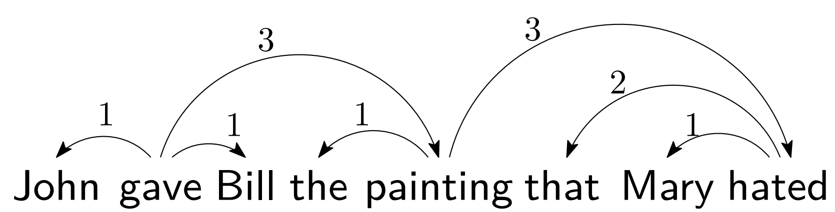
Dependency Distance minimization principle (DDm): tendency of syntactically-related words to be close together in the sentence. Largely regarded as a Linguistic Law. First formulations are due to Behaghel (1930).

A large separation of syntactically-related words incurs in a high cognitive cost (Heringer, Strecker, & Wimmer, 1980; Hudson, 1995). The sum of all dependency distances D (or edge lengths) is used as a *proxy* to measure this cost.

An example in two sentences



$D = 18$



$D = 12$

Source: (Morrill, 2000)

Application – Quantitative Dependency Syntax – DDm

First evidence of DDm from a Romanian corpus (Ferrer-i-Cancho, 2004)

Expected sum of edge lengths in a uniformly random linear arrangement

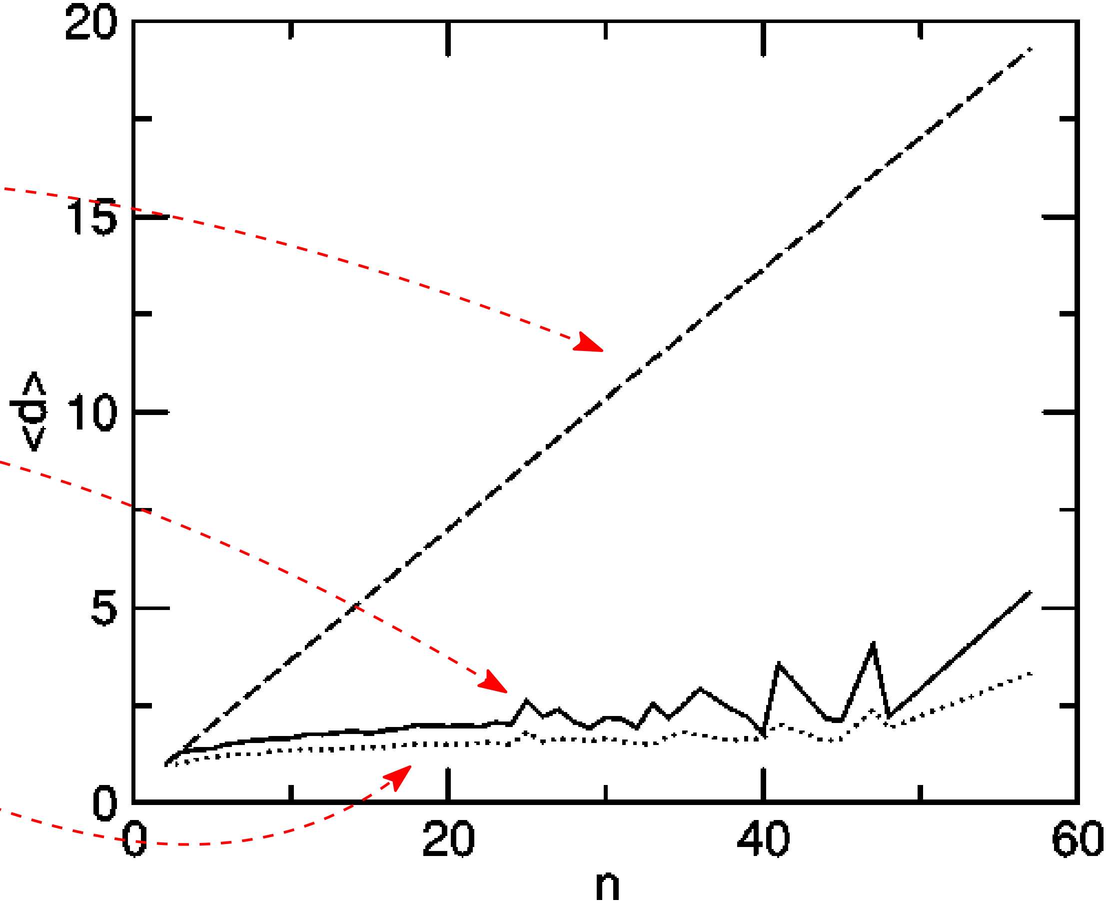
$$\mathbb{E}[D(T)]/(n-1)$$

Real sum of edge lengths in each sentence of the corpus

$$D(T)/(n-1)$$

Minimum sum of edge lengths in each sentence of the corpus

$$m[D(T)]/(n-1)$$



Quantitative Linguistics – baselines and normalization scores

Normalize D using the random and minimum baselines (Ferrer-i-Cancho, Gómez-Rodríguez, Esteban, et al., 2022)

$$\Omega_\pi(T) = \frac{\mathbb{E}[D(T)] - D_\pi(T)}{\mathbb{E}[D(T)] - m[D(T)]}$$

But Ω does not apply projectivity or planarity. What if we...

- used constrained expected values?
- maximum (unconstrained and constrained) values?

Planar arrangements (uar)

Partition

$$\Pi_{\mathsf{pl}}(T) = \bigcup_{u \in V} \Pi^{\diamond}_{\mathsf{pr}}(T^u)$$

and thus

$$\mathbf{N}_{\mathsf{pl}}(T) = \sum_{u \in V} \mathbf{N}^{\diamond}_{\mathsf{pr}}(T^u).$$

$$\mathbf{N}_{\mathsf{pr}}(T^r) = (d(r)+1)! \prod_{u \in \Gamma(r)} \mathbf{N}_{\mathsf{pr}}(T^r_u)$$

$$= \prod_{v \in V} (d_r(v)+1)!.$$

For any $u \in V$,

$$\mathbf{N}^{\diamond}_{\mathsf{pr}}(T^u) = d(u)! \prod_{v \in \Gamma(u)} \mathbf{N}_{\mathsf{pr}}(T^u_v) = \prod_{v \in V} d(v)!.$$

Notice that

$$\mathbf{N}^{\diamond}_{\mathsf{pr}}(T^{u_1}) = \cdots = \mathbf{N}^{\diamond}_{\mathsf{pr}}(T^{u_n}).$$

Finally,

$$\mathbf{N}_{\mathsf{pl}}(T) = n\mathbf{N}^{\diamond}_{\mathsf{pr}}(T^u),$$

for any $u \in V$.

The algorithm produces a planar arrangement u.a.r. because,

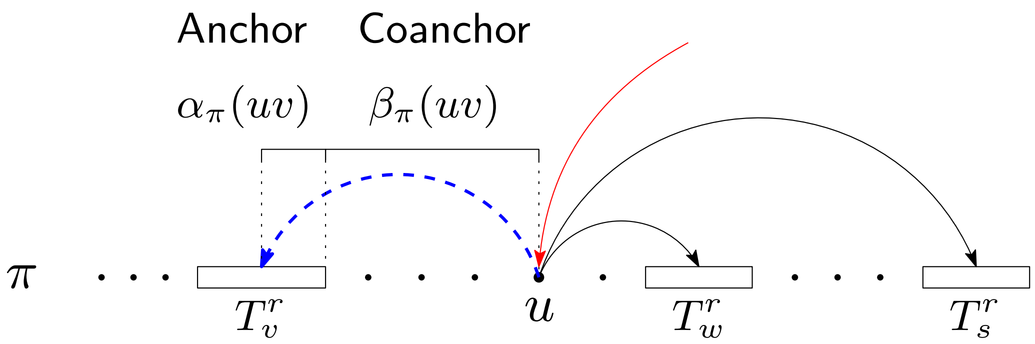
$$\frac{1}{n}\frac{1}{d(u)!} \prod_{v \in \Gamma(u)} \frac{1}{\mathbf{N}_{\mathsf{pr}}(T^u_v)} = \frac{1}{n}\frac{1}{d(u)!} \prod_{v \in V \setminus \{u\}} \frac{1}{d(v)!} = \frac{1}{n}\frac{1}{\mathbf{N}^{\diamond}_{\mathsf{pr}}(T^u)} = \frac{1}{\mathbf{N}_{\mathsf{pl}}(T)}.$$

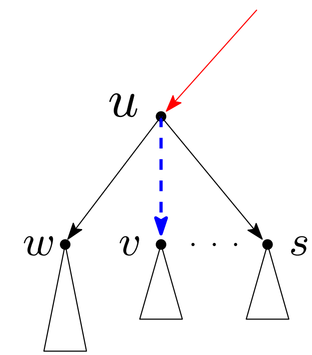
Expectation of D – Projective case

→ A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

Derivation technique for $\mathbb{E}_{\text{pr}}[\delta(uv)]$: split the edge into two parts, called *anchor* and *coanchor*.



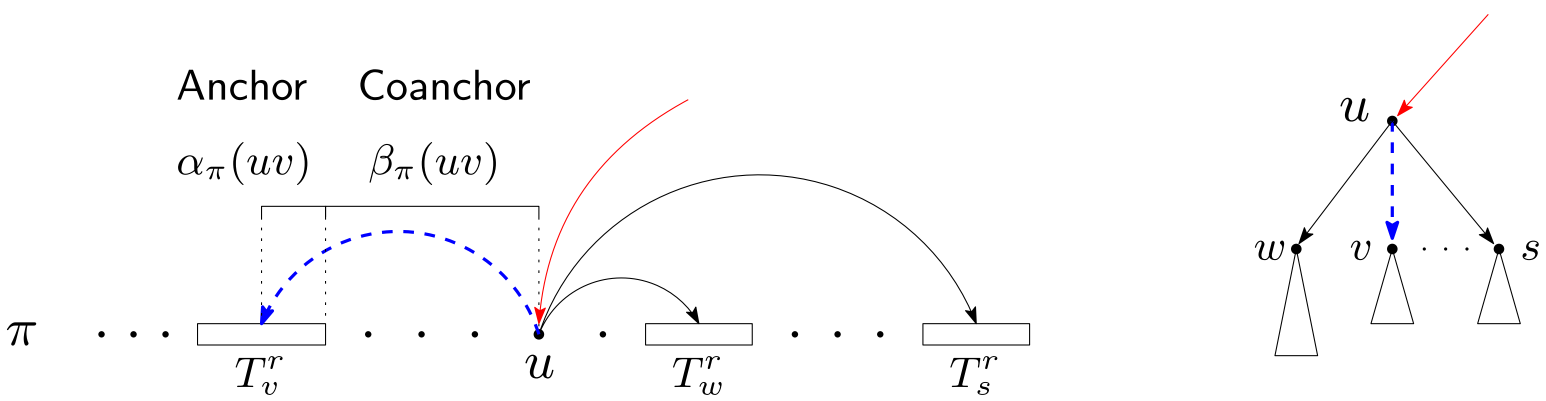


Expectation of D – Projective case

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

Derivation technique for $\mathbb{E}_{\text{pr}}[\delta(uv)]$: split the edge into two parts, called *anchor* and *coanchor*.



Notice that $\mathbb{E}_{\text{pr}}[\delta(uv)] = \mathbb{E}_{\text{pr}}[\alpha(uv)] + \mathbb{E}_{\text{pr}}[\beta(uv)]$, for which we have that

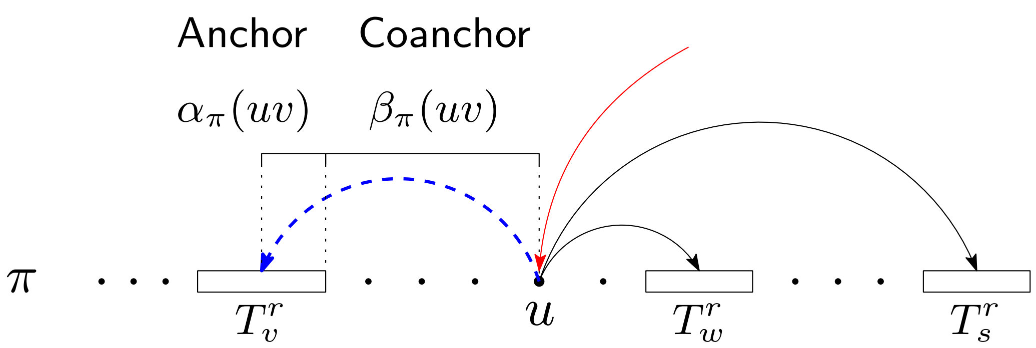
$$\mathbb{E}_{\text{pr}}[\alpha(uv)] = \frac{s_r(v)+1}{2}, \qquad \mathbb{E}_{\text{pr}}[\beta(uv)] = \frac{s_r(u)-s_r(v)-1}{3}$$

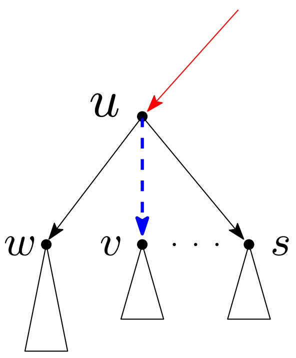
Expectation of D – Projective case

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

Derivation technique for $\mathbb{E}_{\text{pr}}[\delta(uv)]$: split the edge into two parts, called *anchor* and *coanchor*.





Notice that $\mathbb{E}_{\text{pr}}[\delta(uv)] = \mathbb{E}_{\text{pr}}[\alpha(uv)] + \mathbb{E}_{\text{pr}}[\beta(uv)]$, for which we have that

$$\mathbb{E}_{\text{pr}}[\alpha(uv)] = \frac{s_r(v)+1}{2}, \qquad \mathbb{E}_{\text{pr}}[\beta(uv)] = \frac{s_r(u)-s_r(v)-1}{3}$$

$\mathbb{E}_{\text{pr}}[D(T^r)]$ follows from linearity of expectation, grouping the edges by subtrees

$$\mathbb{E}_{\text{pr}}[D(T^r)] = \sum_{u\in\Gamma(r)} \mathbb{E}_{\text{pr}}[\delta(ru)] + \sum_{u\in\Gamma(r)} \mathbb{E}_{\text{pr}}[D(T^r_u)]$$

Expectation of D – Planar case

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

→ A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

The derivation of $\mathbb{E}_{\text{pl}}[D(T)]$ is slightly different:

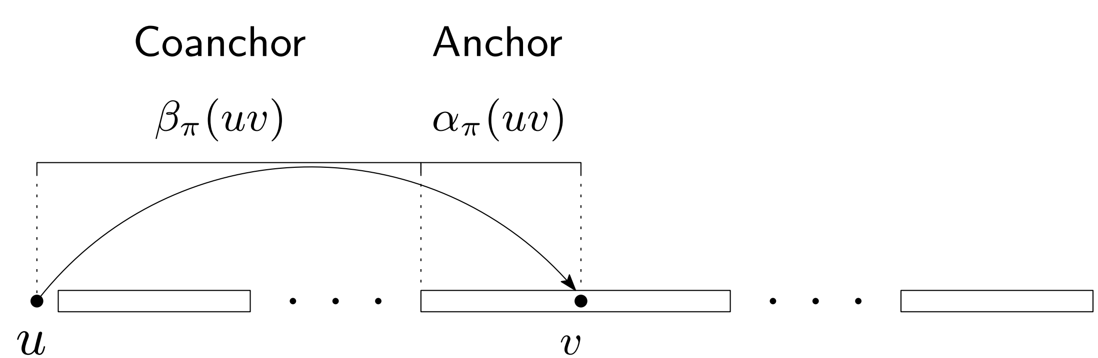
Expectation of D – Planar case

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

→ A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

The derivation of $\mathbb{E}_{\text{pl}}[D(T)]$ is slightly different:

$$\mathbb{E}_{\text{pl}}[D(T)] = \sum_{u \in V} \mathbb{E}_{\text{pl}}[D(T) \mid \pi(u) = 1] \mathbb{P}_{\text{pl}}(\pi(u) = 1)$$



Expectation of D – Planar case

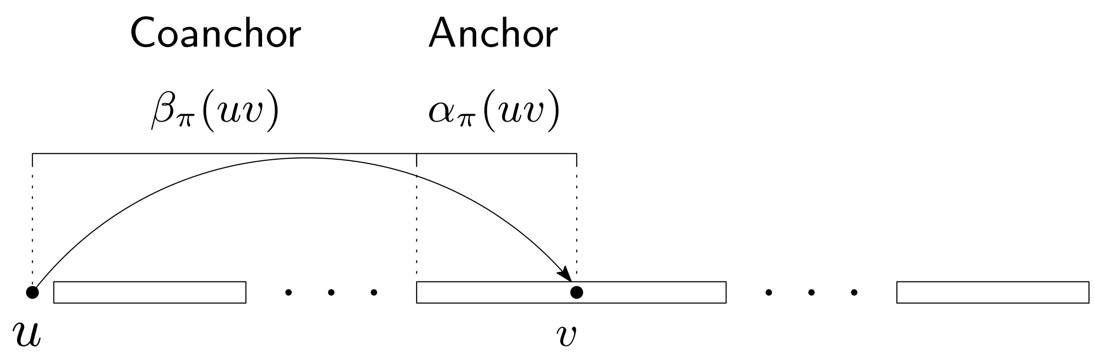
A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

→ A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

The derivation of $\mathbb{E}_{\mathrm{pl}}[D(T)]$ is slightly different:

$$\mathbb{E}_{\mathrm{pl}}[D(T)] = \sum_{u \in V} \mathbb{E}_{\mathrm{pl}}[D(T) \mid \pi(u) = 1] \mathbb{P}_{\mathrm{pl}}(\pi(u) = 1)$$

Expected value of D when u is fixed at position 1.



Expectation of D – Planar case

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

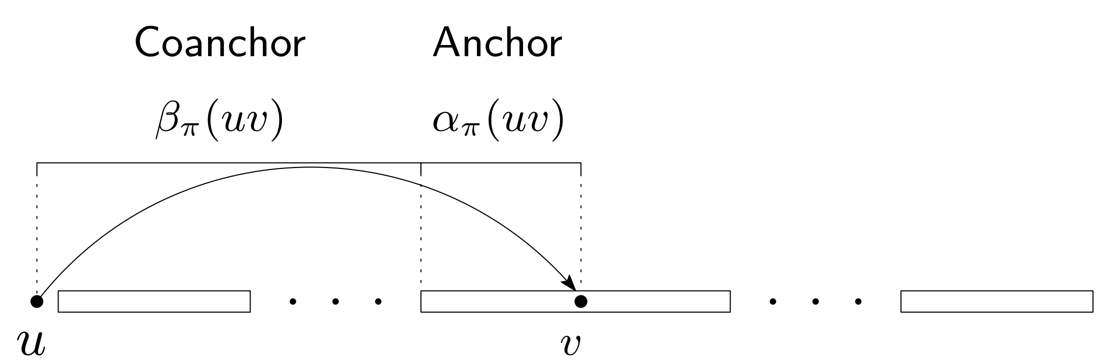
→ A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

The derivation of $\mathbb{E}_{\mathsf{pl}}[D(T)]$ is slightly different:

$$\mathbb{E}_{\mathsf{pl}}[D(T)] = \sum_{u \in V} \mathbb{E}_{\mathsf{pl}}[D(T) \mid \pi(u) = 1] \mathbb{P}_{\mathsf{pl}}(\pi(u) = 1)$$

Proportion of planar arrangements in which vertex u is placed at position 1.

Expected value of D when u is fixed at position 1.



Expectation of D – Planar case

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

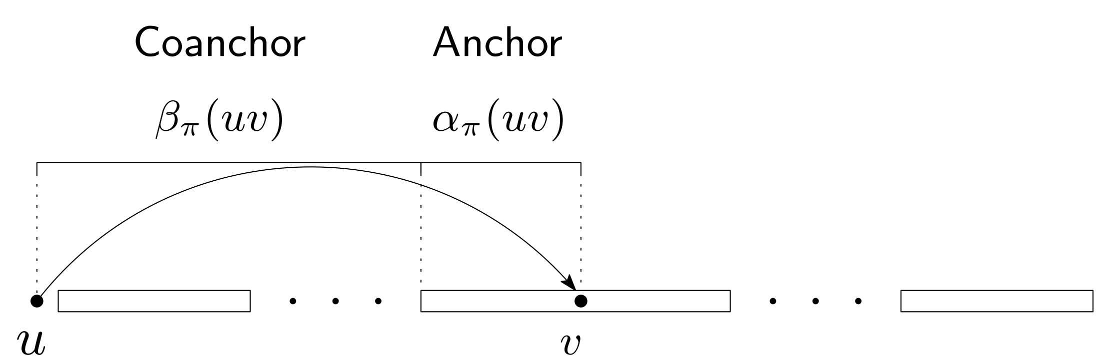
The derivation of $\mathbb{E}_{\text{pl}}[D(T)]$ is slightly different:

$$\mathbb{E}_{\text{pl}}[D(T)] = \sum_{u \in V} \mathbb{E}_{\text{pl}}[D(T) \mid \pi(u) = 1] \mathbb{P}_{\text{pl}}(\pi(u) = 1)$$

$$= \frac{1}{n} \sum_{u \in V} \mathbb{E}^{\diamond}_{\text{pr}}[D(T^u)]$$

Proportion of planar arrangements in which vertex u is placed at position 1.

Expected value of D when u is fixed at position 1.



Expectation of D – Planar case

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

→ A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

The derivation of $\mathbb{E}_{\text{pl}}[D(T)]$ is slightly different:

$$\mathbb{E}_{\text{pl}}[D(T)] = \sum_{u \in V} \mathbb{E}_{\text{pl}}[D(T) \mid \pi(u) = 1]\,\mathbb{P}_{\text{pl}}(\pi(u) = 1)$$

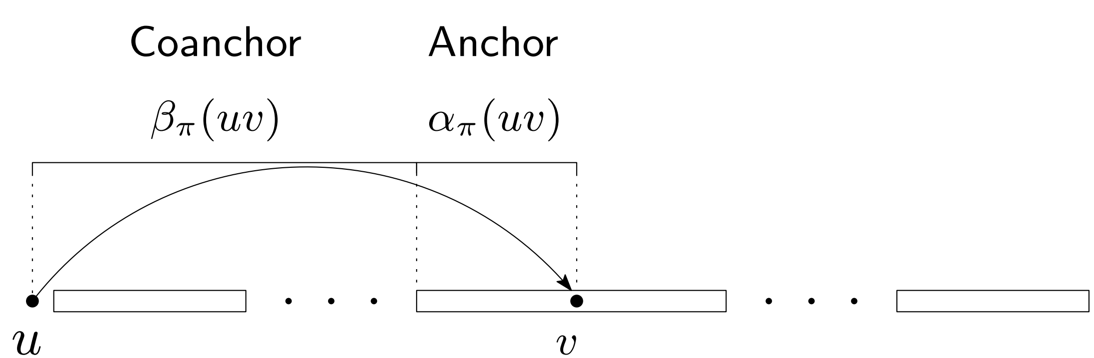
Proportion of planar arrangements in which vertex u is placed at position 1.

Expected value of D when u is fixed at position 1.

$$= \frac{1}{n} \sum_{u \in V} \mathbb{E}^{\diamond}_{\text{pr}}[D(T^u)]$$

$$\mathbb{E}^{\diamond}_{\text{pr}}[D(T^u)] = \sum_{v \in \Gamma(u)} \mathbb{E}^{\diamond}_{\text{pr}}[\delta(uv) \mid u] + \sum_{v \in \Gamma(u)} \mathbb{E}_{\text{pr}}[D(T^u_v)]$$

Expected length of an edge incident to u in projective arrangements of T^u when u is fixed to position 1.



Expectation of D – Planar case

A.-P., L., & Ferrer-i-Cancho, R. (2022). Linear-time calculation of the expected sum of edge lengths in projective linearizations of trees. *Computational Linguistics*, *48*(3), 491–516. https://doi.org/10.1162/coli_a_00442

→ A.-P., L., & Ferrer-i-Cancho, R. (2024). The expected sum of edge lengths in planar linearizations of trees. *Journal of Language Modelling*, (1), 1–42. https://doi.org/10.15398/jlm.v12i1.362

The derivation of $\mathbb{E}_{\text{pl}}[D(T)]$ is slightly different:

$$\mathbb{E}_{\text{pl}}[D(T)] = \sum_{u \in V} \mathbb{E}_{\text{pl}}[D(T) \mid \pi(u) = 1] \mathbb{P}_{\text{pl}}(\pi(u) = 1)$$

Proportion of planar arrangements in which vertex u is placed at position 1.

Expected value of D when u is fixed at position 1.

$$= \frac{1}{n} \sum_{u \in V} \mathbb{E}^{\diamond}_{\text{pr}}[D(T^u)]$$

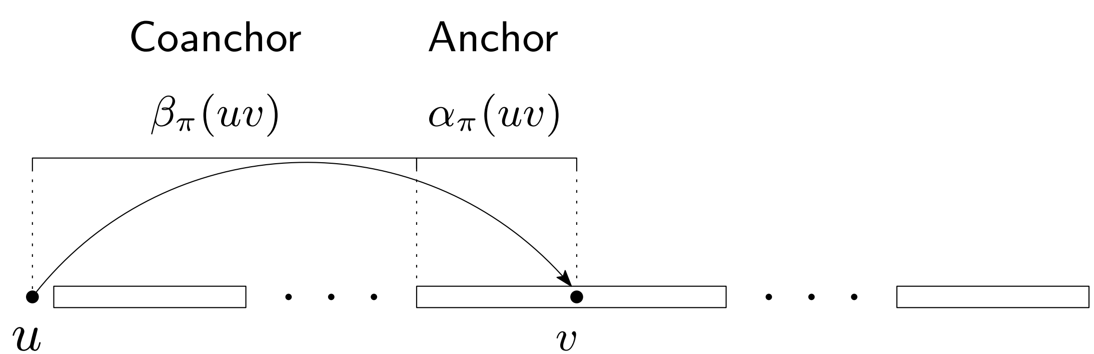
$$\mathbb{E}^{\diamond}_{\text{pr}}[D(T^u)] = \sum_{v \in \Gamma(u)} \mathbb{E}^{\diamond}_{\text{pr}}[\delta(uv) \mid u] + \sum_{v \in \Gamma(u)} \mathbb{E}_{\text{pr}}[D(T^u_v)]$$

Expected length of an edge incident to u in projective arrangements of T^u when u is fixed to position 1.

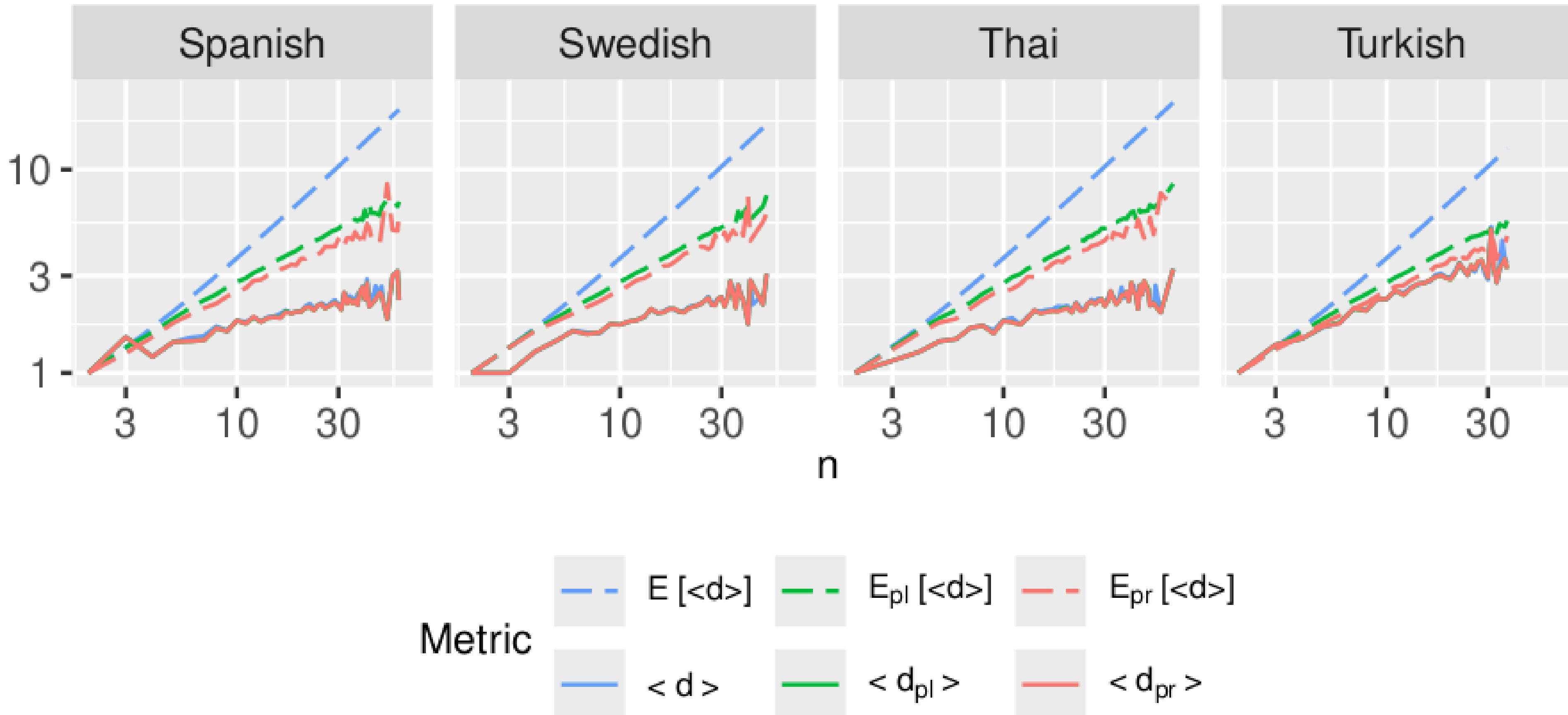
Finally,

$$\mathbb{E}^{\diamond}_{\text{pr}}[\alpha(uv) \mid u] = \mathbb{E}_{\text{pr}}[\alpha(uv) \mid u]$$

$$\mathbb{E}^{\diamond}_{\text{pr}}[\beta(uv) \mid u] = \frac{3}{2}\mathbb{E}_{\text{pr}}[\beta(uv) \mid u]$$



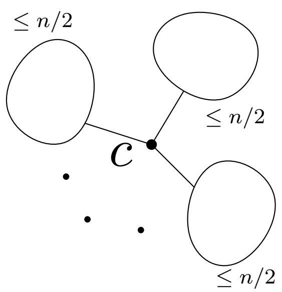
Some values of $\mathbb{E}_{\text{pr}}[D(T^r)]$ and $\mathbb{E}_{\text{pl}}[D(T)]$



Minimum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

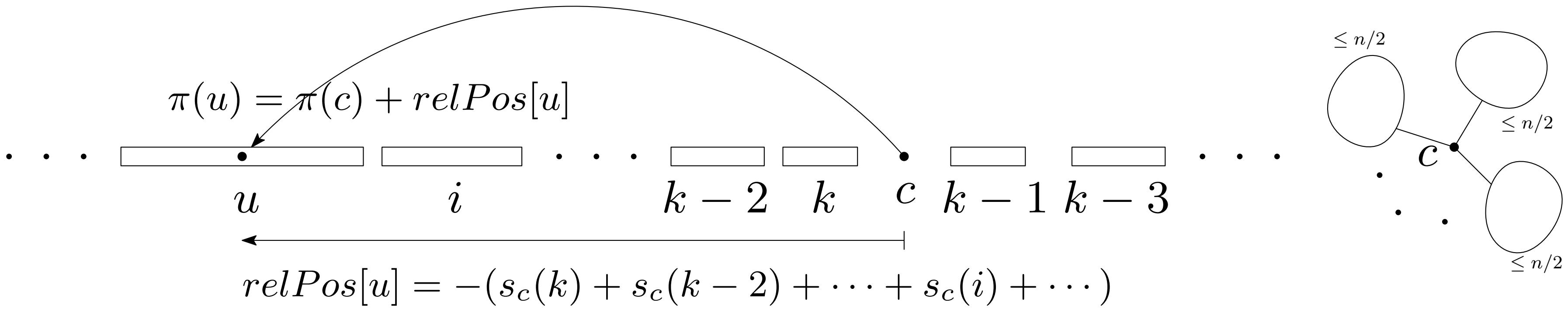
Hochberg and Stallmann (2003): place a centroidal vertex c between the subtrees of T^c. The embedding method for subtrees calculates, for each vertex u, a displacement/offset $relPos[u]$ with respect to c.



Minimum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

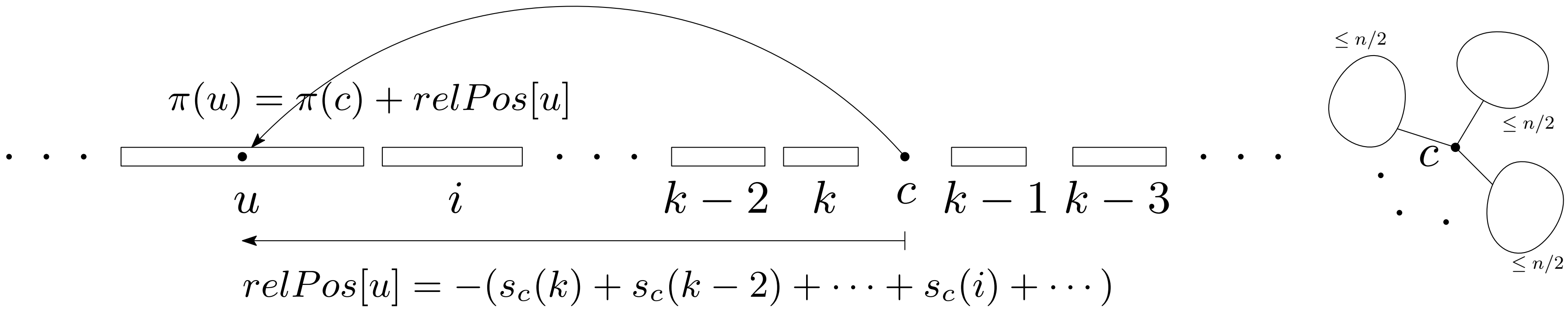
Hochberg and Stallmann (2003): place a centroidal vertex c between the subtrees of T^c. The embedding method for subtrees calculates, for each vertex u, a displacement/offset $relPos[u]$ with respect to c.



Minimum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

Hochberg and Stallmann (2003): place a centroidal vertex c between the subtrees of T^c. The embedding method for subtrees calculates, for each vertex u, a displacement/offset $relPos[u]$ with respect to c.



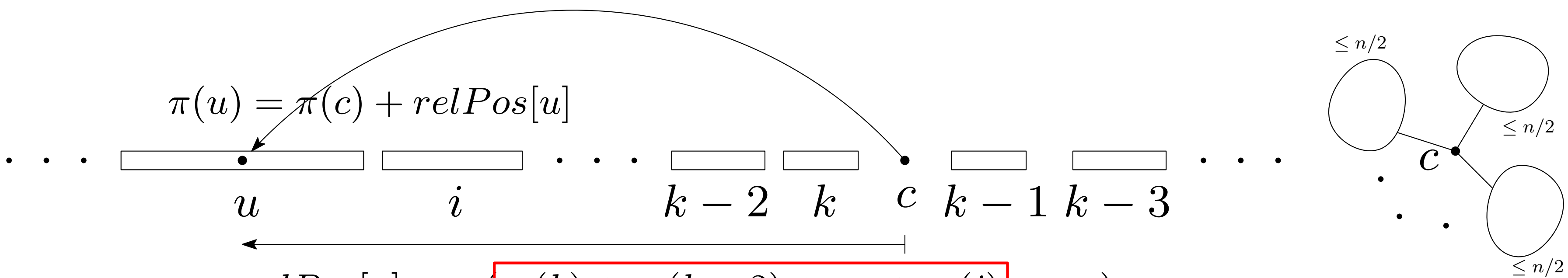
Embedding branches

```
procedure EMBEDBRANCH (v, base, dir)
  before ← after ← 0
  for i = k downto 1 do
    if i is even then
      EMBEDBRANCH(v_i, base − dir ∗ before, −dir)
      before ← before + n_i
    else ▷ i is odd
      EMBEDBRANCH(v_i, base + dir ∗ after, dir)
      after ← after + n_i
    endif
  end do
  relPos[v] ← base + dir ∗ (before + 1)
```

Minimum arrangements – Planar case

A.-P., L., Esteban, J. L., & Ferrer-i-Cancho, R. (2022). Minimum projective linearizations of trees in linear time. *Information Processing Letters*, *174*, 106204. https://doi.org/10.1016/j.ipl.2021.106204

Hochberg and Stallmann (2003): place a centroidal vertex c between the subtrees of T^c. The embedding method for subtrees calculates, for each vertex u, a displacement/offset $relPos[u]$ with respect to c.



$$relPos[u] = -(\boxed{s_c(k) + s_c(k-2) + \cdots + s_c(i)} + \cdots)$$

Embedding branches

procedure EMBEDBRANCH $(v, base, dir)$

```
before ← after ← 0
for i = k downto 1 do
    if i is even then
        EMBEDBRANCH(v_i, base − dir ∗ before, −dir)
        before ← before + n_i
    else ▷ i is odd
        EMBEDBRANCH(v_i, base + dir ∗ after, dir)
        after ← after + n_i
    endif
end do
relPos[v] ← base + dir ∗ (before + 1)
```

Algorithm 3.2: EMBED_BRANCH corrected.

```
Function EMBED_BRANCH(L^c, v, base, dir, relPos) is
    C_v ← L^c[v]
    before ← after ← 0
    under_anchor ← 0
    for i = 2 to |C_v| step 2 do
        v_i, n_i ← C_v[i]
        under_anchor ← under_anchor + n_i
    base ← base + dir ∗ (under_anchor + 1)
    for i = |C_v| downto 1 do
        v_i, n_i ← C_v[i]
        if i is even then
            EMBED_BRANCH(L^c, v_i, base − dir ∗ before, −dir, relPos)
            before ← before + n_i
        else
            EMBED_BRANCH(L^c, v_i, base + dir ∗ after, dir, relPos)
            after ← after + n_i
    relPos[v] ← base
```

Branch and Bound for MaxLA

Construct the arrangement from left to right. Rules to add a new vertex:

- Symmetry breaking constraints, such as Nurse's properties, Path Optimization Lemma (for bridges), parallelization over vertex orbits
- Prediction and propagation of level values (Path Optimization Lemma)
- Curated upper bounds for better bounding
- Specific algorithms for different states of the algorithm
- Initialized with the maximum between `Bipartite MaxLA` and `1-thistle MaxLA`

n	Time (ms)	n	Time (ms)	n	Time (ms)	n	Time (ms)	n	Time (s)	n	Time
1	0.00	8	0.03	15	0.21	22	8.76	29	0.468	36	27.8 s
2	0.02	9	0.04	16	0.35	23	15.4	30	0.838	37	46.7 s
3	0.02	10	0.04	17	0.56	24	29.5	31	1.41	38	1 m 33 s
4	0.02	11	0.05	18	0.99	25	43.2	32	2.88	39	2 m 42 s
5	0.02	12	0.09	19	1.66	26	86.2	33	5.06	40	4 m 51 s
6	0.02	13	0.10	20	2.92	27	146	34	8.69		
7	0.03	14	0.14	21	4.82	28	253	35	15.3		