

Linear extensions of partially ordered sets

Júlia Éles

Applied mathematics MSc

2026

Introduction

- ▶ Given a set P and a partial ordering \prec on it, let us count the number of linear extensions of it
- ▶ This is the number of topological orderings of the *cover graph* $G = (V, A)$ of P , where $V = P$ and $A = \{uv \in P^2 : u \prec v, \nexists w \in P \setminus \{u, v\} : u \prec w \prec v\}$

Previous results

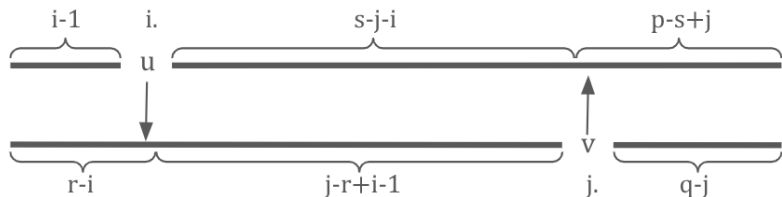
- ▶ G. Brightwell and P. Winkler: the problem is $\#P$ -hard [2]
- ▶ Kangas et al.: dynamic programming-based algorithm with time complexity of $O(n^{t+4})$ [4]
- ▶ K. Kangas, M. Koivisto and S. Salonen: improved to $O(n^{t+3})$ using nice tree decomposition [5]

An interesting special case

- ▶ When the cover graph is a tree, there exists an algorithm with a running time of $O(n^2)$, given by M. D. Atkinson [1]
- ▶ *Spectrum*: $\sigma_G : V \rightarrow \mathbb{Z}_+^n$
- ▶ $\sigma_G(v)_i$ is the number of topological orders in which the vertex v is in the i -th position
- ▶ Let uv be an arc, and split up the graph along this arc. Determine the spectrum of u and v recursively, and then join them together

Common spectrum, arc spectrum

- ▶ *Common spectrum*: $\tau_G : V^2 \rightarrow \mathbb{Z}_+^{n \times n}$
- ▶ $\tau_G(u, v)_{ij}$ is the number of topological orders in which the vertex u is in the i -th position, vertex v is in the j -th position



Spectrum for more vertices

- ▶ Spectrum for k vertices: $\tau_G^k : V^k \rightarrow \mathbb{Z}_+^{\{1, \dots, n\}^k}$
- ▶ Conjecture: it is possible to calculate τ_G^k from the spectra of fewer vertices

Position-specific ordering

- ▶ $G = (V, A)$ directed, acyclic graph, $f : V \rightarrow 2^{\{1, \dots, n\}}$ function of the allowed positions
- ▶ Task: determine, if there exists any topological ordering $\pi : V \rightarrow \{1, \dots, n\}$ of G such that $\pi(v) \in f(v) \forall v \in V$
- ▶ *One-position case*: for every vertex v , $f(v)$ is either the set $\{1, \dots, n\}$ or $|f(v)| = 1$
- ▶ If $|f(v)| = 1$, then v is a *fixed vertex*, otherwise v is a *free vertex*. We define *fixed positions* and *free positions* similarly

Greedy algorithms for one-position case

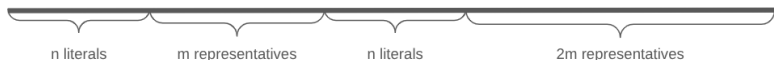
- ▶ T. Hagerup and M. Maas [3]: for every vertex v , determine the values $\text{Low}(v)$ and $\text{High}(v)$
- ▶ Iterate through the vertices in increasing order of their High values
- ▶ Let the current vertex be v , the set of occupied positions be S , and let J be the set of positions reachable by v :
$$J := [\text{Low}(v), \text{High}(v)] \setminus S$$
- ▶ Assign $\pi(v)$ to $\min J$ if J is not empty
- ▶ If J is never empty, then there exists a solution, and the given π satisfies the function f

Greedy algorithms for one-position case

- ▶ Another approach: for every free vertex v define the values $l(v)$ and $h(v)$ as the number of free positions needed before and after the vertex v
- ▶ For every vertex pair u, v , where exists at least one directed $u \rightsquigarrow v$ path, let $d(u, v)$ be the number of free positions needed between the two vertices
- ▶ Based on the l , h and d values, for each free vertex v determine the set I_v of possible positions
- ▶ Iterate through the free positions in increasing order, and for position i choose the vertex v such that $i \in I_v$ and $\max I_v$ is minimal. Assign $\pi(v) := i$. If such v does not exist, then there is no solution

NP-hardness

- ▶ If G is the union of directed stars and $|f(v)| \in \{2, 3\}$ for every vertex v , then the position specific ordering problem is NP-hard
- ▶ For any 3-SAT problem with n literals and m clauses, we construct a directed, acyclic graph G with position prescription f , whose vertices are the literals and the representatives of clauses
- ▶ The 3-SAT problem has a solution if and only if there is a topologic ordering of G respect to the function f



Future plans

- ▶ Verifying the conjecture regarding the spectrum of multiple vertices
- ▶ Explore further variants of position-specific ordering problems

References



M. D. Atkinson.

On computing the number of linear extensions of a tree.
Order, 7(1):23–25, 1990.



G. Brightwell and P. Winkler.

Counting linear extensions is #P-complete.
In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, pages 175–181, 1991.



T. Hagerup and M. Maas.

Generalized topological sorting in linear time.
Nord. J. Comput., 1(1):38–49, 1994.



K. Kangas, T. Hankala, T. M. Niinimäki, and M. Koivisto.

Counting linear extensions of sparse posets.
In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 603–609, 2016.



K. Kangas, M. Koivisto, and S. Salonen.

A faster tree-decomposition based algorithm for counting

Usage of AI tools

- ▶ Writing small helper scripts in Python
- ▶ Translating, spell checking