Exact Exponential Algorithms for Two Poset Problems

László Kozma
Freie Universität Berlin, Institute of Computer Science, Germany
`http://www.lkozma.net/`
laszlo.kozma@fu-berlin.de

Abstract

Partially ordered sets (posets) are fundamental combinatorial objects with important applications in computer science. Perhaps the most natural algorithmic task, given a size-n poset, is to compute its number of *linear extensions*. In 1991 Brightwell and Winkler showed this problem to be #P-hard. In spite of extensive research, the fastest known algorithm is still the straightforward $O(n2^n)$-time dynamic programming (an adaptation of the Bellman-Held-Karp algorithm for the TSP). Very recently, Dittmer and Pak showed that the problem remains #P-hard for *two-dimensional* posets, and no algorithm was known to break the 2^n-barrier even in this special case. The question of whether the two-dimensional problem is easier than the general case was raised decades ago by Möhring, Felsner and Wernisch, and others. In this paper we show that the number of linear extensions of a two-dimensional poset can be computed in time $O(1.8286^n)$.

The related *jump number problem* asks for a linear extension of a poset, minimizing the number of neighboring *incomparable* pairs. The problem has applications in scheduling, and has been widely studied. In 1981 Pulleyblank showed it to be NP-complete. We show that the jump number problem can be solved (in arbitrary posets) in time $O(1.824^n)$. This improves (slightly) the previous best bound of Kratsch and Kratsch.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases poset, linear extension, jump number, exponential time

Digital Object Identifier 10.4230/LIPIcs.SWAT.2020.30

Funding *László Kozma*: Research supported by DFG grant KO 6140/1-1.

1 Introduction

A *partially ordered set* (*poset*) $\mathcal{P} = (X, \prec)$ consists of a ground set X and an irreflexive and transitive binary relation $\prec$ on X. A *linear extension* of $\mathcal{P}$ is a total order on X that contains $\prec$. The main problem considered in this paper is to determine, given a poset $\mathcal{P}$ on a ground set of size n, the number of linear extensions $\mathrm{LE}(\mathcal{P})$ of $\mathcal{P}$. We refer to this counting problem as #LE. A poset can alternatively be seen as a transitive directed acyclic graph (DAG), where #LE asks for the number of topological orderings of the graph.

Posets are fundamental objects in combinatorics (for a detailed treatment we refer to the monographs [44, 37], [40, § 3], [19, § 8]) with several applications in computer science. For instance, every comparison-based algorithm (e.g. for sorting) implicitly defines a sequence of posets on the input elements, where each poset captures the pairwise comparisons known to the algorithm at a given time. An efficient sorter must find comparisons whose outcomes split the number of linear extensions in a balanced way. A central and long-standing open question in this area is whether a comparison with ratio (at worst) 1/3 : 2/3 exists in every poset [5]; slightly weaker constant ratios are known to be achievable [23, 4].

Counting linear extensions (exactly or approximately) is a bottleneck in experimental work, e.g. when testing combinatorial conjectures. In computer science the #LE problem is relevant, besides the mentioned task of optimal comparison-based sorting, for learning graphical models [45, 34], probabilistic ranking [46, 18, 30], reconstruction of partial orders from sequential data [31], convex rank tests [32], multimedia delivery in networks [1], and others.

The complexity of #LE has been thoroughly studied (see Linial [28] for an early reference). Lovász [29, § 2.4] mentions the problem as a special case of polytope volume computation; Stanley [39] gives a broad overview of the polytope-formulation of #LE. Brightwell and Winkler [6] show that #LE is #P-hard, and thus unlikely to admit a polynomial-time solution. In fact, despite the significant attention the problem has received (e.g. the mentioned papers and references therein and thereof), the best upper bound on the running time remains $O(n2^n)$. This bound can be achieved via dynamic programming over the subsets of the ground set [26, 11], an approach[1] that closely resembles the Bellman-Held-Karp algorithm for the *traveling salesman problem* (*TSP*) [3, 20].

A bound of 2^n appears to be a natural barrier[2] for the running time of #LE, similarly to some of the most prominent combinatorial optimization problems (e.g. *set cover/hitting set*, *CNF-SAT*, *graph coloring*, *TSP*).[3] We show that #LE can be solved faster when the input poset is *two-dimensional.* Dimension is perhaps the most natural complexity-measure of posets, and can be seen informally as a measure of the *nonlinearity* of a poset (see e.g. Trotter [44]). As one-dimensional posets are simply total orders, the first nontrivial case is dimension two. The structure of two-dimensional posets is, however, far from trivial. Posets in this class capture the *point-domination order* in the plane.

The question of the complexity of #LE in two-dimensional posets was raised in the 1980s by Möhring [33] and later by Felsner and Wernisch [17]. An even earlier mention of the problem is by Atkinson, Habib, and Urrutia, in a discussion of open problems concerning posets, cf. Rival [37, p. 481].

Efficient algorithms for #LE are known for various restricted classes of posets, e.g. *series-parallel* [33], *low treewidth* [25, 24, 15], *small width* [11], avoiding certain *substructures* [16], and others; see Möhring [33] for an early survey of tractable special cases. However, as the techniques used in these works rely on certain kinds of *sparsity* in the input, they are not applicable for the case of two-dimensional posets. It is easy to see that the latter may be arbitrarily dense, containing, for example, a complete bipartite graph of linear size. In fact, Dittmer and Pak [12] recently showed that #LE is #P-hard already for this class of inputs. Our first result is stated in the following theorem.

▶ **Theorem 1.** *The number of linear extensions of a two-dimensional poset of size n can be computed in time $O(1.8286^n)$.*

Our second result is an algorithm for the *jump number problem.* In this (optimization) problem a linear extension of $\mathcal{P}$ is sought, such as to minimize the number of adjacent pairs of elements that are incomparable in $\mathcal{P}$ (such pairs are called *jumps*). The problem is known to be NP-hard [36], and has been well-studied due to its applications in scheduling.

Similarly to #LE, the jump number problem can be solved by dynamic programming in time $2^n n^{O(1)}$. An improved algorithm with running time $O(1.8638^n)$ was given by Kratsch and Kratsch [27]. We also refer to their paper for further background and motivation for the problem. Improving the bound of Kratsch and Kratsch, we obtain the following result.

[1] A finer bound on the running time is $O(w \cdot |I|)$, where w is the *width* of the poset, and I is its set of *ideals* (i.e. downsets); in the worst case, however, this expression does not improve the given bound.

[2] We only study exact algorithms in this paper; for *approximating* $\mathrm{LE}(\mathcal{P})$, fully polynomial-time randomized schemes are known [14, 7].

[3] The *strong exponential time hypothesis* [21] states that a running time $O(c^n)$ with $c < 2$ is not achievable for CNF-SAT, and a similar barrier has been conjectured for set cover [10].

▶ **Theorem 2.** *The jump number problem can be solved in time $O(1.824^n)$.*

Note that in this case no assumption is made on the dimension of the input poset. Whether *jump number* remains NP-hard in two-dimensional posets is a long-standing open question [33, 8, 42].

Poset dimension. Formally, the dimension $\dim(\mathcal{P})$ of a poset $\mathcal{P} = (X, \prec)$ is the smallest number d of total orders, whose intersection is $\mathcal{P}$. In other words, if $\dim(\mathcal{P}) = d$, then there exists a collection of orders $<_1, \dots, <_d$ (called *realizers* of $\mathcal{P}$), such that for all $x, y \in X$, we have $x \prec y$ if and only if $x <_k y$ for all $1 \leq k \leq d$.

Poset dimension was introduced by Dushnik and Miller in 1941 [13], and the concept has since been extensively studied; we refer to the monograph of Trotter dedicated to poset dimension theory [44]. Various kinds of *sparsity* of $\mathcal{P}$ are known to imply upper bounds on $\dim(\mathcal{P})$ (see e.g. [22, 38] for recent results in a long line of such works). The converse is, in general, not true, as two-dimensional posets may already be arbitrarily dense, and are known not to have a characterisation in terms of finitely many forbidden substructures [2, 33].

The term *dimension* is motivated by the following natural geometric interpretation. Suppose $\mathcal{P}$ is a d-dimensional, size-n poset with realizers $<_1, \dots, <_d$. The ground set can then be viewed as a set of n points in d-dimensional Euclidean space, with no two points aligned on any coordinate, such that the ordering of the points according to the k-th coordinate coincides with the order $<_k$, for all $1 \leq k \leq d$. The partial order $\prec$ is then exactly the *point-domination order*, i.e. $x \prec y$ if and only if all d coordinates of y are larger than the corresponding coordinates of x. In this geometric view, a linear extension of a low-dimensional poset can be seen as a tour that visits all points, never moving behind the *Pareto front* of the already visited points.

Two-dimensional posets are particularly natural, as they are in bijection with permutations (the ranks of points by $<_1$ and $<_2$ can be seen respectively as the *index* and *value* of a permutation-entry). Swapping the two coordinates yields a *dual* poset, turning chains into antichains and vice versa. It follows that the complement of the *comparability graph* is itself a comparability graph, which is yet another exact characterization of two-dimensional posets.[4] It is not hard to see that two-dimensional posets are exactly the *inclusion-posets* of intervals on a line.

Yet another interpretation of two-dimensional posets relates them to the weak Bruhat order on permutations. In this setting the number of linear extensions of a two-dimensional poset equals the number of permutations that are *reachable* from a given permutation π by a sequence of swaps between mis-sorted adjacent elements; a question of independent interest [17, 12].

2 Counting linear extensions in two-dimensional posets

Denote $[k] = \{1, \dots, k\}$. For a set Y with partial order $\prec$, let $\max(Y)$ denote the *set of maxima* of Y, i.e. the set of elements $x \in Y$ with the property that $x \prec y$ implies $y \notin Y$.

Let $\mathcal{P} = (X, \prec)$ be a size-n poset. To introduce the main elements of our #LE algorithm, we review first the classical $O(n2^n)$ time algorithm.

[4] Given a poset $\mathcal{P} = (X, \prec)$, its *comparability graph* is $\mathsf{C}(\mathcal{P}) = (X, E)$, where $\{x, y\} \in E$ if $x \prec y$ or $y \prec x$. The *width* of $\mathcal{P}$ is the size of the largest *antichain* in $\mathcal{P}$, i.e. independent set in $\mathsf{C}(\mathcal{P})$, and the *height* of $\mathcal{P}$ is one less than the size of the largest *chain* in $\mathcal{P}$, i.e. clique in $\mathsf{C}(\mathcal{P})$.

For all $Y \subseteq X$, let $\mathrm{LE}(Y)$ denote the number of linear extensions of the subposet of $\mathcal{P}$ induced by Y, and let $\mathrm{LE}(\emptyset) = 1$. We recursively express $\mathrm{LE}(Y)$ for all nonempty Y, by removing in turn all elements that can appear at the end of a total order on Y:

$$\mathrm{LE}(Y) = \sum_{x \in \max(Y)} \mathrm{LE}\left(Y \setminus \{x\}\right). \tag{1}$$

To compute $\mathrm{LE}(\mathcal{P}) = \mathrm{LE}(X)$, we evaluate recurrence (1), saving all intermediate entries $\mathrm{LE}(Y)$ for $Y \subseteq X$. There are at most 2^n such entries, and computing each takes $O(n)$ time, once the results of the recursive calls are available. (With simple bookkeeping, $\max(Y)$ is available for all calls without additional overhead.)

2.1 A first improvement

A well-known observation is that when computing $\mathrm{LE}(X)$ by (1) only those subproblems $Y \subseteq X$ arise where $y \in Y$ and $x \prec y$ imply $x \in Y$, i.e. the *downsets* of $\mathcal{P}$. In general, the number of downsets can be as high as 2^n, when $\mathcal{P}$ consists of a single antichain. Nonetheless, we can give better bounds on the number of downsets, if necessary, by modifying the input poset $\mathcal{P}$.

Large matching case. An observation already made in previous works (e.g. [27]) is the following. Consider a size-m *matching* M in the comparability graph $\mathsf{C}(\mathcal{P})$, with matched edges $\{x_i, y_i\}$, where $x_i \prec y_i$, for all $i \in [m]$. Let W denote the set of vertices matched by M and let $A = X \setminus W$.

Then, the sets $Y \subseteq X$ where $Y \cap \{x_i, y_i\} = \{y_i\}$ for some $i \in [m]$ are not downsets and cannot be reached by recursive calls. The remaining sets can be partitioned as $T_0 \cup T_1 \cup \cdots \cup T_m$, where $T_0 \subseteq A$ is an arbitrary subset of the unmatched vertices, and $T_i \in \{\emptyset, \{x_i\}, \{x_i, y_i\}\}$ for $i \in [m]$.

The number of sets of this form is $2^{n-2m} \cdot 3^m$. If $m = \alpha n$, this quantity equals $(2 \cdot (\frac{3}{4})^{\alpha})^n$. When $\alpha \geq 1/3$, the number of subproblems is thus less than 1.8172^n, and the running time is within the required bounds.

Small matching case. Let us assume from now on that M is a *maximum* matching of size $m = \alpha n$ for $\alpha < 1/3$. The maximality of M implies that the unmatched vertices A form an independent set in $\mathsf{C}(\mathcal{P})$, i.e. an antichain of $\mathcal{P}$, of size $|A| = (1 - 2\alpha)n$. We assume $\alpha > 0$, as otherwise $\mathcal{P}$ is a single antichain and the problem is trivial.

For $x \in A$, let $N(x)$ denote the *open neighborhood* of x in $\mathsf{C}(\mathcal{P})$, i.e. the set of elements in X that are comparable with x. Observe that $N(x) \cap A = \emptyset$ for all $x \in A$.

If $N(x) \cap A = \emptyset$ for an element $x \in W$, then we say that x is *incomparable* with A. Otherwise, if $x \prec y$ for some $y \in N(x) \cap A$, we say that x is *below* A, and if $y \prec x$, for some $y \in N(x) \cap A$, we say that x is *above* A. Observe that x cannot be both below and above A, as that would make two elements of A comparable, contradicting the fact that A is an antichain.

The sets $N(x)$ define a *partition* of A, where $x, y \in A$ are in the same class if and only if $N(x) = N(y)$. In general posets there can be as many as $\min\{2^{n-|A|}, |A|\}$ classes. The following lemma states that in two-dimensional posets the number of classes is much smaller.

▶ **Lemma 3.** *Let $\mathcal{P} = (X, \prec)$ be a size-n poset, with $\dim(\mathcal{P}) \leq 2$, and let $A \subseteq X$ be an antichain. Then, $N(\cdot)$ partitions A into at most $2(n - |A|)$ classes.*

Before proving Lemma 3, we show that it can be used to compute LE($\mathcal{P}$) more efficiently. Let $A_1, \dots, A_\ell$ be the partition of A defined by $N(\cdot)$, and for each $i \in [\ell]$, denote $a_i = |A_i|$. Let x_i^k, where $i \in [\ell]$ and $k \in [a_i]$, be a *virtual element*, and let Q denote the set of all such virtual elements.

Construct a new poset $\mathcal{P}' = (X', \prec')$ as follows. Let $X' = W \cup Q$. In words, the ground set X' contains all vertices matched by M, and instead of the elements of the antichain A, it contains the virtual elements of Q. Observe that $|Q| = |A|$ and therefore $|X'| = |X|$.

The relation $\prec'$ is defined as follows, covering all cases:

- if $x, y \in W$, then $x \prec' y \iff x \prec y$,
- if $x = x_i^p$ and $y = x_j^q$, then $x \prec' y \iff i = j$ and $p < q$,
- if $x \in W$ and $y = x_i^p$, then $x \prec' y \iff x \prec z$, for some $z \in A_i$,
- if $x = x_i^p$ and $y \in W$, then $x \prec' y \iff z \prec y$, for some $z \in A_i$.

In words, $\prec'$ preserves the relation $\prec$ between elements of W. Virtual elements with the same index i form a chain $x_i^1 \prec' \cdots \prec' x_i^{a_i}$, for all $i \in [\ell]$. Virtual elements with different indices are incomparable. The relation between a virtual element x_i^k and an element $y \in W$ preserves the relation $\prec$ between an arbitrary element $z \in A_i$ and y. The choice of z is indeed arbitrary, as the elements in A_i are by definition indistinguishable.

Intuitively, x_i^k is a placeholder for the element of A_i that appears as the k-th among all elements of A_i in some linear extension of $\mathcal{P}$. The sequence $x_i^1, \dots, x_i^{a_i}$ corresponds to an arbitrary permutation of the elements of A_i. This intuition is captured by the following statement.

▶ **Lemma 4.** *With the above definitions:*

$$LE(\mathcal{P}) = \prod_{i \in [\ell]} (a_i!) \cdot LE(\mathcal{P}').$$

Let us postpone proving Lemma 4 as well, and state our first algorithm, #LE-2D, as Algorithm 1. The algorithm constructs the poset $\mathcal{P}'$ and computes its number of linear extensions using recurrence (1), then computes the correct count for $\mathcal{P}$ via Lemma 4.

Algorithm 1 Algorithm #LE-2D.

```
   Input: Poset $\mathcal{P} = (X, \prec)$, where $|X| = n$.
   Output: The number of linear extensions LE($\mathcal{P}$) of $\mathcal{P}$.
1: Find a maximum matching $M$ of C($\mathcal{P}$) with vertex set $W$.
2: Let $A = X \setminus W$.
3: Let $A_1, \dots, A_\ell$ be the partition of $A$ by the neighborhoods in C($\mathcal{P}$).
4: Let $a_i = |A_i|$ for $i \in [\ell]$.
5: Construct $\mathcal{P}' = (X', \prec')$, as described.
6: Compute $N$ = LE($\mathcal{P}'$) using (1).
7: return $\prod_{i \in [\ell]} (a_i!) \cdot N$.
```

Analysis of the running time. Step 1 amounts to running a standard maximum matching algorithm (see e.g. [43]). Computing the partition in Step 3 takes linear time with careful data structuring. Steps 2,4,5,7 clearly take linear time overall.

The polynomial-time overhead of steps other then Step 6, as well as the polynomial factor in the analysis of (1) are absorbed in the exponential running time of Step 6, where we round the base of the exponential upwards. To derive a worst-case upper bound on the running time of Step 6, it only remains to bound the number of downsets of $\mathcal{P}'$.

Observe that the ground set X' can, by construction, be partitioned into chains. The matched vertices of W are partitioned into m chains $x_i \prec' y_i$, for $i \in [m]$, as before. The virtual elements of Q are partitioned into ℓ chains of lengths $a_1, \dots, a_\ell$, where the i-th chain is $x_i^1 \prec' \cdots \prec' x_i^{a_i}$. All downsets of $\mathcal{P}'$ are then of the form $(T_1 \cup \cdots \cup T_m) \cup (Q_1 \cup \cdots \cup Q_\ell)$, where $T_i \in \{\emptyset, \{x_i\}, \{x_i, y_i\}\}$ for $i \in [m]$, and $Q_i = \{x_i^j : j \leq t_i\}$, for some threshold $0 \leq t_i \leq a_i$, for $i \in [\ell]$.

The number of such sets is $3^m \cdot \prod_{i \in [\ell]} (a_i + 1)$. Recall that $m = \alpha n$ and $|A| = (1 - 2\alpha)n$. The quantity $\prod_{i=1}^{\ell} (a_i + 1)$ is maximized when the values $a_i + 1$ are all equal, and thus equal to $(|A| + \ell)/\ell$, yielding the overall upper bound $\left(3^\alpha \left(\frac{(1-2\alpha)n}{\ell} + 1\right)^\ell\right)^n$. (Observe that $\ell \geq 1$ always holds.)

Since the quantity is increasing in ℓ, and $\ell \leq 2(n - |A|) = 4\alpha n$ by Lemma 3, we obtain the upper bound $\left(3^\alpha \left(\frac{1+2\alpha}{4\alpha}\right)^{4\alpha}\right)^n$. In the range of interest $0 < \alpha < 1/3$ the base achieves its maximum for $\alpha \approx 0.258$ at a value below 1.975, resulting in the bound $O(1.975^n)$ on the running time.

To reach the bound given in Theorem 1, we need further ideas. Let us first prove the two lemmas from which the correctness of the current algorithm and its analysis follow.

Proof of Lemma 4. Let $q = \prod_{i=1}^{\ell} (a_i!)$. We describe an explicit mapping from linear extensions of $\mathcal{P}$ to linear extensions of $\mathcal{P}'$.

Consider a linear extension $<$ of $\mathcal{P}$ viewed as a sequence $z = (z_1, \dots, z_n)$, where $z_1 < \cdots < z_n$. The sequence z contains ℓ disjoint subsequences of lengths $a_1, \dots, a_\ell$ formed respectively by the elements of $A_1, \dots, A_\ell$. Let $z' = (z'_1, \dots, z'_n)$ be the sequence obtained from z by replacing, for all $i \in [\ell]$, the elements of A_i in the sequence z, in the order of their appearance, by the virtual elements $x_i^1, \dots, x_i^{a_i}$.

We proceed via two claims about the mapping $z \to z'$ from which the statement follows: (1) z' is a linear extension of $\mathcal{P}'$, and (2) for every linear extension z' of $\mathcal{P}'$ there are q different linear extensions of $\mathcal{P}$ that map to z'.

For (1), let i_1, i_2 be two arbitrary indices $1 \leq i_1 < i_2 \leq n$. We need to show that $z'_{i_2} \not\prec' z'_{i_1}$. The four cases to consider are: (1a) $z'_{i_1}, z'_{i_2} \in W$, (1b) $z'_{i_1} = x_i^p$ and $z'_{i_2} = x_j^q$, (1c) $z'_{i_1} \in W$ and $z'_{i_2} = x_i^p$, and (1d) $z'_{i_1} = x_i^p$, and $z'_{i_2} \in W$. These correspond to the four cases in the definition of $\prec'$ and the claim easily follows in each case by the construction of z'.

For (2), consider a linear extension (sequence) z' of $\mathcal{P}'$, and for all $i \in [\ell]$ replace the elements $\{x_i^1, \dots, x_i^{a_i}\}$ in z' by an arbitrary permutation of the elements of A_i. In this way we obtain q different linear extensions of $\mathcal{P}$, and when applying the above mapping to these linear extensions, they all yield the same z'. ◀

Proof of Lemma 3. Let $t = |A|$, and let us label the elements of A as $z_1, \dots, z_t$. Let $<_1$ and $<_2$ be the realizers of the two-dimensional poset $\mathcal{P}$. Then, as A is an antichain, its elements can be labeled such that $z_1 <_1 \cdots <_1 z_t$, and $z_t <_2 \cdots <_2 z_1$. The crucial observation is that the neighborhood of an arbitrary $y \in X \setminus A$ in A is defined by an *interval* of indices.

Formally, for $y \in X \setminus A$ that is above or below A, let z_i, z_j be the elements of $N(y) \cap A$ with smallest, resp. largest index (it may happen that $i = j$). Define $b(y) = i - 0.5$ and $b'(y) = j + 0.5$ the *boundaries* of the neighborhood of y. If y is incomparable with A, set the boundaries to dummy values $b(y) = 0$, $b'(y) = t + 1$.

If y is above A, then for all k such that $b(y) < k < b'(y)$, we have $z_k \prec y$. To see this, observe that $z_k <_1 z_j <_1 y$, and $z_k <_2 z_i <_2 y$.

Symmetrically, if y is below A, then for all k such that $b(y) < k < b'(y)$, we have $y \prec z_k$. To see this, observe that $y <_1 z_i <_1 z_k$, and $y <_2 z_j <_2 z_k$.

Let $b_1, \dots, b_{2(n-t)}$ be the multiset of neighborhood boundaries sorted in increasing order. Their number is $2(n-t)$ as each of the $n-t$ elements of $X \setminus A$ contribute exactly two boundaries. Let us add the two dummy boundaries $b_0 = 0$ and $b_{2(n-t)+1} = t+1$ (in case they never occurred during the process).

The classes of A defined by the partition $N(\cdot)$ are then of the form $\{z_j : b_i < j < b_{i+1}\}$ where $0 \le i \le 2(n-t)$. There are at most $2(n-t)+1$ such classes (not all boundaries are necessarily distinct, and we can now remove empty classes due to duplicate boundaries). Moreover, the two classes delimited by b_0 to the left, respectively by $b_{2(n-t)+1}$ to the right are identical, corresponding to elements of A incomparable to all $y \in X \setminus A$. The claimed bound on the maximum number of classes follows. ◀

2.2 A faster algorithm

We now describe the improvements to Algorithm #LE-2D and its analysis that lead to the running time claimed in Theorem 1.

Canonical matchings. Observe that set A in Lemma 3 denotes an arbitrary antichain. When A is assumed to be the complement of a maximum matching with a certain property, a stronger statement can be shown.

Let M be a maximum matching of $\mathsf{C}(\mathcal{P})$, let W be its vertex set, and let $A = X \setminus W$. We call an edge $\{x_i, y_i\}$ of M *separated*, if there exist $x_1, x_2 \in A$ such that $x_i \prec x_1$ and $x_2 \prec y_i$. (In other words, x_i is below A, and y_i is above A.) Observe that, in this case, x_1 and x_2 must be the same, as otherwise M could be made larger by replacing edge $\{x_i, x_j\}$ by the two edges $\{x_i, x_1\}$, $\{x_2, x_j\}$. A matching is *canonical* if it contains no separated edges.

We argue that in an arbitrary poset a canonical matching of the same size as the maximum matching can be found in polynomial time. Indeed, start with an arbitrary maximum matching M. If M contains no separated edges, we are done. Otherwise, let $\{x_i, y_i\}$ be an edge of M with $x \in A$ such that $x_i \prec x \prec y_i$. (Such a triplet can easily be found in polynomial time.) Replace the edge $\{x_i, y_i\}$ in M by the edge $\{x, y_i\}$. As x was previously not matched, the resulting set of edges is still a maximum matching. We claim that with $O(n^2)$ such swaps we obtain a canonical matching (i.e. one without separated edges). To see this, consider as potential function the sum of ranks of all vertices in the current matching, according to an arbitrary fixed linear extension of $\mathcal{P}$. Each swap increases the potential by at least one (since x must come after x_i in every linear extension). Since the sum of ranks is an integer in $O(n^2)$, the number of swaps until we are done is also in $O(n^2)$. In the following, we can therefore assume that M is a canonical maximum matching. We can now state the stronger structural lemma.

▶ **Lemma 5.** *Let $\mathcal{P} = (X, \prec)$ be a size-n poset, with $\dim(\mathcal{P}) \le 2$. Let M be a canonical maximum matching in $\mathsf{C}(\mathcal{P})$ with vertex set W, and let $A = X \setminus W$. Then, $N(\cdot)$ partitions A into at most $|W|$ classes.*

Proof. Since M is canonical, for every edge $\{x_i, y_i\}$, one of the following must hold:

(i) x_i and y_i are both above A,
(ii) x_i and y_i are both below A,
(iii) x_i is incomparable with A and y_i is above A,
(iv) y_i is incomparable with A and x_i is below A.

Recall that in the proof of Lemma 3, we considered, for all $y \in X \setminus A$, the two boundaries of the interval $N(y) \cap A$. Now, in cases (iii) and (iv), only one of x_i and y_i need to be considered, as the neighborhood of the other is disjoint from A.

In cases (i) and (ii), it is also sufficient to consider only one of x_i and y_i as we have $N(x_i) \cap A = N(y_i) \cap A$. Furthermore, in this case $|N(x_i) \cap A| = 1$. To see this, suppose that there are $z, z' \in A$ such that $z \in N(x_i)$ and $z' \in N(y_i)$. Then M could be extended by replacing the edge $\{x_i, y_i\}$ with the edges $\{z, x_i\}$ and $\{z', y_i\}$, contradicting the maximality of M.

It follows that in the argument of Lemma 3 we only need to consider the intervals created by $|M|$ elements of $X \setminus A$, yielding the bound $2|M| = |W|$ on the number of classes. It is easy to construct examples where the bound is tight. ◀

It follows that, if we require the matching M in Step 1 of Algorithm #LE-2D to be canonical, then by Lemma 4, the bound on the number of downsets improves to $\left(3^{\alpha}\left(\frac{1}{2\alpha}\right)^{2\alpha}\right)^{n}$. In the range of interest $0 < \alpha < 1/3$ this quantity is easily upper bounded by 1.8912^n with maximum at $\alpha \approx 0.319$.

Packing triplets and quartets. The final improvement in running time comes from the attempt to find, instead of a matching (i.e. a packing of edges), a packing of larger connected structures. Beyond the concrete improvement, the technique may be of more general applicability and interest, which we illustrate in § 3 for the jump number problem.

Assume, as before, that M is a canonical maximum matching of $\mathsf{C}(\mathcal{P})$ of size αn with vertex set W and that A denotes the antichain $X \setminus W$. Let us form an auxiliary bipartite graph B with vertex sets L and R, where $L = A$, and $R = M$, i.e. R consists of the *edges* of M. A vertex $x \in L$ is connected to a vertex $\{x_i, y_i\} \in R$ exactly if x is comparable to one or both of x_i and y_i. Let M_B be a maximum matching of B, of size βn. Clearly, $\beta \leq \alpha$.

Edges of M_B connect vertices in A to matched edges of M, forming *triplets* of vertices of X that induce connected subgraphs in $\mathsf{C}(\mathcal{P})$. Let T denote the set of all triplets created by edges of M_B.

Let us form now another auxiliary bipartite graph B' with vertex sets L' and R', where L' consist of the vertices of A *unmatched* in M_B, and let $R' = T$, i.e. the triplets found in the previous round. A vertex $x \in L'$ is connected to a vertex $z \in R'$ exactly if x is comparable to at least one of the vertices forming the triplet z. Let $M_{B'}$ be a maximum matching of B', and denote its size by γn. Clearly, $\gamma \leq \beta$.

Edges of $M_{B'}$ connect vertices in A to triplets of T, forming *quartets* of vertices of X that induce connected subgraphs in $\mathsf{C}(\mathcal{P})$. Let Q denote the set of all quartets created by edges of $M_{B'}$.

Let A' denote the vertices of A that were not matched in either of the two matching rounds. Observe that $|A'| = n(1 - 2\alpha - \beta - \gamma)$. We make the following observations.

(1) The endpoints of edges of M that were *unmatched* in M_B are not comparable to any vertex in A' (assuming that A' is nonempty), as otherwise M_B would not have been maximal. There are $n(\alpha - \beta)$ such unmatched edges. These contribute a factor of $3^{n(\alpha-\beta)}$ to the number of downsets.

(2) The vertices in triplets of T that were *unmatched* in $M_{B'}$ are not comparable to any vertex in A' (assuming that A' is nonempty), as otherwise $M_{B'}$ would not have been maximal. There are $n(\beta - \gamma)$ such triplets. A simple case-analysis shows that the number of downsets of a size-3 poset with connected comparability graph is at most 5. It follows that these triplets contribute a factor of at most $5^{n(\beta-\gamma)}$ to the number of downsets.

(3) There are γn quartets in Q. A case-analysis[5] shows that the number of downsets of a size-4 poset with connected comparability graph is at most 9. It follows that these quartets contribute a factor of at most $9^{n\gamma}$ to the number of downsets.

(4) All vertices in $X \setminus A'$ are accounted for. As for the vertices in A', we partition them into classes $A_1, \dots, A_\ell$ by $N(\cdot)$, and apply the same transformation as previously, creating a new poset $\mathcal{P}'$. By the previous discussion, only the vertices from the quartets in Q may be comparable to vertices in A'. Furthermore, in each quartet, only the vertices coming from the original matching M may be comparable to a vertex in A' (other vertices come from the antichain $A \supseteq A'$). Thus, by Lemma 5, the number of classes created on A' is $\ell \le 2\gamma n$.

Putting everything together, assuming $\gamma > 0$ (the case $\gamma = 0$ is discussed later), we obtain the upper bound τ^n on the number of downsets of $\mathcal{P}'$, where $\tau = \tau(\alpha, \beta, \gamma) = 3^{(\alpha-\beta)} \cdot 5^{(\beta-\gamma)} \cdot 9^{\gamma} \cdot \left(\frac{1-2\alpha-\beta+\gamma}{2\gamma}\right)^{2\gamma}$. Under the constraint $0 < \gamma \le \beta \le \alpha < 1/3$, the bound $\tau < 1.8286$ holds, with the maximum attained for $\alpha = \beta = \gamma (\approx 0.1882)$. Observe that the least favorable case occurs when all edges of M are matched into triplets, and all triplets are matched into quartets.

When $\gamma = 0$, no quartets are created, and A' forms a single class, transformed in $\mathcal{P}'$ into a single chain, contributing a linear factor to the overall bound. Thus, the upper bound $n \cdot \tau^n$ holds, with $\tau = \tau(\alpha, \beta) = 3^{(\alpha-\beta)} \cdot 5^{\beta} < 1.71$, maximum attained for $\alpha = \beta (\approx 1/3)$.

The resulting algorithm #LE-2D* is listed as Algorithm 2. The correctness and running time bounds (Theorem 1) follow from the previous discussion. We defer some remarks about the algorithm and its analysis to § 4.

Open questions. The following questions about counting linear extensions are suggested in increasing order of difficulty. (1) Can #LE be solved in two-dimensional posets faster than the algorithm of Theorem 1? (2) Can #LE be solved in time $O(c^n)$ for $c < 2$ in d-dimensional posets, for $d \ge 3$? (3) Can #LE be solved in time $O(c^n)$ for $c < 2$ in arbitrary posets?

3 The jump number problem

In this section we present our improvement for the jump number problem. We start with a formal definition of the problem, and the straightforward dynamic programming. We then review the algorithm of Kratsch and Kratsch, followed by our extension. The result is intended as an illustration of the matching technique of § 2, which is not specific to two-dimensional posets.

Given a linear extension $x_1 < \cdots < x_n$ of a poset $\mathcal{P} = (X, \prec)$, a pair of neighbors (x_i, x_{i+1}) is a *jump* if $x_i \not\prec x_{i+1}$, and is a *bump* if $x_i \prec x_{i+1}$. The number of jumps, resp. bumps of the linear extension $<$ of $\mathcal{P}$ is denoted as jump($<$), resp. bump($<$). The jump number problem asks to compute the minimum possible value jump($<$) for a linear extension $<$ of $\mathcal{P}$. Additionally, a linear extension realizing this value should be constructed. In the algorithms we describe, obtaining a linear extension that realizes the minimum jump number is a mere technicality, we thus focus only on computing the minimum jump number.

An easy observation is that the relation jump($<$) + bump($<$) $= n - 1$ holds for all linear extensions $<$ of $\mathcal{P}$. Minimizing the number of jumps is thus equivalent to maximizing the number of bumps, allowing us to focus on the latter problem.

[5] An easy induction shows more generally that the maximum number of downsets of a size-n poset with connected comparability graph is $2^{n-1} + 1$.

Let $\text{bump}(\mathcal{P})$ denote the maximum bump number of a linear extension of $\mathcal{P}$. For all $Y \subseteq X$, and $x \in \max(Y)$, let $\text{bump}(Y, x)$ denote the maximum bump number of a linear extension of the subposet of $\mathcal{P}$ induced by Y that ends with element x. Let us define $\text{bump}(\{x\}, x) = 0$, for all $x \in X$. We recursively express $\text{bump}(Y, x)$ by removing x from the end and trying all remaining elements in turn as the new last element:

$$\text{bump}(Y, x) = \max_{y \in \max(Y \setminus \{x\})} \Big(\text{bump}(Y \setminus \{x\}, y) + [y \prec x]\Big). \tag{2}$$

The term $[y \prec x]$ denotes the value 1 if $y \prec x$, i.e. if the last pair forms a bump, and 0 otherwise. Executing recurrence (2) naïvely leads to an algorithm that computes $\text{bump}(\mathcal{P})$ in time $O(2^n \cdot n^2)$.

We now describe the improvement of Kratsch and Kratsch [27]. Observe that jumps partition a linear extension of $\mathcal{P}$ uniquely into a sequence of chains of $\mathcal{P}$, such that the last element of each chain is incomparable with the first element of the next chain, and all other neighboring pairs are comparable.

Consider a linear extension with minimum jump number and let $C_1, \dots, C_k$ denote the non-trivial chains of its decomposition (i.e. all chains of length at least 2). Let C denote the set of vertices of chains $C_1, \dots, C_k$. Then, as all bumps occur between elements of C, the bump number of $\mathcal{P}$ equals the bump number of the subposet induced by C. In other words, to compute the maximum bump number, it is sufficient to consider in recurrence (2) the subsets of the ground set X that are candidate sets C in the optimum.

Kratsch and Kratsch consider a maximum matching M of $\mathsf{C}(\mathcal{P})$ with vertex set W and observe that the vertices of the antichain $A = X \setminus W$ that participate in nontrivial chains (i.e. that are in C) form a matching with vertices of W. (This is because a vertex $v \in A$ can only form a bump together with a vertex from $X \setminus A$, and two vertices $v, v' \in A$ cannot form bumps with the same vertex, as that would contradict their incomparability.) Moreover, $v, v' \in A$ cannot be the neighbors of the two endpoints of a matched edge of M, as that would contradict the maximality of M.

Thus, it suffices to compute (2) over subsets of X that consist of $W \cup A'$, where $A' \subseteq A$ and $|A'| \leq |M|$. Furthermore, only *downsets* of $\mathcal{P}$ need to be considered, leading to a further saving due to the fact that W forms a matching. Denoting $|M| = \alpha n$, the overall number of subsets of X that need to be considered is $\binom{(1-2\alpha)n}{\leq \alpha n} \cdot 3^{\alpha n}$.

Packing triplets. We now describe our improvement. Again, let M be a canonical maximum matching of $\mathsf{C}(\mathcal{P})$ of size αn with vertex set W and let A denote the antichain $X \setminus W$. Form an auxiliary bipartite graph B with vertex sets L and R, where $L = A$, and $R = M$. A vertex $x \in L$ is connected to a vertex $\{x_i, y_i\} \in R$ (i.e. an edge of M) exactly if $x \prec y_i$ or $x_i \prec x$. Let M_B be a maximum matching of B, and denote its size by βn. Clearly, $\beta \leq \alpha$.

Edges of M_B connect vertices in A to matched edges of M, forming *triplets* of vertices of X that induce connected subgraphs in $\mathsf{C}(\mathcal{P})$. Let T denote the set of all such triplets. (To keep the argument simple we forgo in this case further rounds of matching and the forming of quartets. The result is thus not optimized to the fullest extent.)

Let A' denote the vertices of A that were not matched in M_B. Observe that $|A'| = n(1 - 2\alpha - \beta)$. We make the following observations.

(1) The endpoints of edges of M that were *unmatched* in M_B are not comparable to any vertex in A' (assuming that A' is nonempty), as otherwise M_B would not have been maximal. There are $n(\alpha - \beta)$ such edges. These edges contribute a factor of $3^{n(\alpha-\beta)}$ to the number of downsets.

(2) There are βn triplets in T. These contribute a factor of at most $5^{n\beta}$ to the number of downsets.

(3) All vertices in $X \setminus A'$ are accounted for. Vertices of A' that participate in non-trivial chains of the optimal linear extension can be matched to vertices in the triplets of T. A vertex $v \in A'$ can only be connected to those vertices of a triplet in T that are endpoints of an edge in M (all other vertices come from A, are thus incomparable with v). Furthermore, $v, v' \in A'$ may not connect to different endpoints of the same edge in M, as that would contradict the maximality of M. It follows that each vertex in A' that participates in C must be matched to a unique triplet in T. Thus, at most βn vertices of A' need to be considered.

The resulting Algorithm JN is listed as Algorithm 3. Its correctness follows from the previous discussion.

Running time. In the large matching ($\alpha \geq 1/3$) case, the bound given in § 2 on the number of downsets holds, and the running time is within the bound of Theorem 2. We assume therefore that $\alpha < 1/3$.

In the special case $\beta = 0$ only vertices in M need to be considered, with an overall upper bound $3^{\alpha n}$ on the number of downsets. For $\alpha < 1/3$, this quantity is below 1.443^n.

Assuming $\beta > 0$, we have an upper bound τ^n on the number of downsets of $\mathcal{P}$, where $\tau^n = 3^{(\alpha-\beta)n} \cdot 5^{\beta n} \cdot \binom{n(1-2\alpha-\beta)}{\leq \beta n}$. To obtain a simpler expression, we use a standard upper bound [9, p. 406] on the sum of binomial coefficients. Assuming $0 \leq 2b \leq a \leq 1$, we have $\binom{na}{\leq nb} = \sum_{k=0}^{nb} \binom{na}{k} \leq n^{O(1)} \cdot \left(\frac{a^a}{b^b \cdot (a-b)^{(a-b)}}\right)^n$.

Plugging in $a = 1 - 2\alpha - \beta$ and $b = \beta$, and assuming $2b \leq a$, we have $2\alpha + 3\beta \leq 1$. Omitting the polynomial factor, we obtain $\tau \leq 3^{(\alpha-\beta)} \cdot 5^{\beta} \cdot \frac{(1-2\alpha-\beta)^{(1-2\alpha-\beta)}}{\beta^{\beta} \cdot (1-2\alpha-2\beta)^{(1-2\alpha-2\beta)}}$. In the critical region $0 < \beta \leq \alpha < 1/3$ we obtain the bound $\tau < 1.824$, with the maximum attained for $\alpha = \beta (\approx 0.1918)$.

When $2\alpha + 3\beta \geq 1$, we use the easier upper bound on the sum of binomial coefficients $\binom{na}{\leq nb} \leq 2^{na}$, obtaining $\tau \leq 3^{(\alpha-\beta)} \cdot 5^{\beta} \cdot 2^{(1-2\alpha-\beta)}$. In the allowed range $0 < \beta \leq \alpha < 1/3$ and additionally requiring $2\alpha + 3\beta \geq 1$, the quantity is maximized for $\alpha = \beta = 0.2$, yielding $\tau < 1.8206$, within the required bounds.

4 Discussion

We start with some remarks about algorithm #LE-2D*. It is straightforward to extend this algorithm beyond pairs, triplets, and quartets, to also form k-tuples for $k > 4$ via further matching rounds. A similar analysis, however, indicates no further improvements in the upper bound. When forming triplets and quartets, other strategies are also possible. For instance, we may try to combine connected pairs of edges from M into quartets. The quartets formed in this way are, in fact, preferable to those obtained by augmenting triplets, as their number of downsets is strictly less than the value 9 given before. (The value 9 is attained when 3 of the 4 vertices form an antichain, which is not possible if the quartet consists of two matched edges.)

We observe that in some instances, the largest antichain may be significantly larger than the antichain A obtained as the complement of the maximum matching. One can find the largest antichain in time $O(n^{5/2})$ via a reduction to bipartite matching (see e.g. [43]). In these cases, using the partition of the antichain $A_i, \dots, A_\ell$ (without arguing about matchings) may lead to a better running time. In two-dimensional posets with a realization $<_1, <_2$, the largest antichain can be found in time $O(n \log n)$ by reduction to the *largest decreasing subsequence* problem. In our analysis, we assumed the classes A_i to be of equal size. The running time can, of course, be significantly lower when the distribution of class sizes is far from uniform.

We further remark that the actual time and space requirement of our algorithms is dominated by the number of downsets of a given poset $\mathcal{P}$ (or rather, of the transformed poset $\mathcal{P}'$). The number of downsets (order ideals) is known to equal the number of antichains [40, § 3]. Counting antichains is, in general, #P-hard [35], but solvable in two-dimensional posets in polynomial time [41, 33]. Thus, assuming that the transformed poset $\mathcal{P}'$ is also two-dimensional, we can efficiently compute a precise, *instance-specific* estimate of the time and space requirements of our algorithms.

To see that indeed, $\dim(\mathcal{P}') \leq 2$, recall that $\mathcal{P}'$ is obtained from $\mathcal{P}$ by replacing antichains A_i by chains of equal size, such that the comparability of the involved elements to elements in $X \setminus A$ is preserved. We can obtain a two-dimensional embedding of $\mathcal{P}'$ by starting with a two-dimensional embedding of $\mathcal{P}$, with points having integer coordinates, and no two points aligned on either coordinate. For an arbitrary $x \in A_i$, form a 0.5×0.5 box around the point x, and place the chain replacing A_i on the main diagonal of this box. Then the comparability of points in the chain with elements in $X \setminus A_i$ is the same as for the point x.

Optimization. Our first two bounds in § 2 depend only on the fraction α of matched vertices, and their maxima are found using standard calculus. The final bounds given in Theorem 1 and Theorem 2 however, require us to optimize over unwieldy multivariate quantities with constrained variables. The given numerical bounds were obtained using Wolfram Mathematica software. We have, however, independently certified the bounds, by the method illustrated next.

Suppose we want to show that $\tau < 1.8286$, where $\tau = \tau(\alpha, \beta, \gamma) = 3^{(\alpha-\beta)} \cdot 5^{(\beta-\gamma)} \cdot 9^{\gamma} \cdot \left(\frac{1-2\alpha-\beta+\gamma}{2\gamma}\right)^{2\gamma}$, and $A \leq \gamma \leq \beta \leq \alpha \leq B$, for $A, B \in (0, 1/3)$.

Consider a box $\mathcal{B} = [\alpha_1, \alpha_2] \times [\beta_1, \beta_2] \times [\gamma_1, \gamma_2] \subseteq [A, B]^3$. Then, at an *arbitrary* point $(\alpha, \beta, \gamma) \in \mathcal{B}$, the following (rather weak) upper bound holds.

$$\tau(\alpha, \beta, \gamma) \leq 3^{\alpha_2} \cdot \left(\frac{5}{3}\right)^{\beta_2} \cdot \left(\frac{9}{5}\right)^{\gamma_2} \cdot \left(\frac{1 - 2\alpha_1 - \beta_1 + \gamma_2}{2\gamma_1}\right)^{2\gamma_2}.$$

To show $\tau < 1.8286$, it is sufficient to exhibit a collection of boxes, such that (1) for all boxes, the stated upper bound evaluates to a value smaller than 1.8286, and (2) the union of the boxes covers the entire domain of the variables. We can find such a collection of boxes if we start with a single box that contains the entire domain of the variables, and recursively split boxes into two equal parts (along the longest side) whenever the upper bound evaluates to a value larger than the required value.

Higher dimensions. A straightforward extension of Algorithms #LE-2D and #LE-2D* to higher dimensional posets does not yield improvements over the naïve dynamic programming. The crux of the argument in two dimensions is that a large antichain in $\mathcal{P}$ is split, according to the neighborhoods in $\mathsf{C}(\mathcal{P})$ into a small number of classes. In dimensions three and above it is easy to construct posets with an antichain containing *almost all* elements, e.g. such that $|X \setminus A| = O(\sqrt{n})$, with the property that all elements of A have unique neighborhoods. In this case, the number of classes is $|A|$ and the described techniques yield no significant savings.

Algorithm 2 Algorithm #LE-2D*.

```
    Input: Poset $\mathcal{P} = (X, \prec)$, where $|X| = n$.
    Output: The number of linear extensions $\mathrm{LE}(\mathcal{P})$ of $\mathcal{P}$.
1:  Find a maximum matching $M$ of $\mathsf{C}(\mathcal{P})$ with vertex set $W$.
2:  Let $A = X \setminus W$.
3:  Find $T$ and $Q$ as described, and let $A'$ be the unmatched part of $A$.
4:  Let $A_1, \dots, A_\ell$ be the partition of $A'$ by the neighborhoods in $\mathsf{C}(\mathcal{P})$.
5:  Let $a_i = |A_i|$ for $i \in [\ell]$.
6:  Construct $\mathcal{P}' = (X', \prec')$.
7:  Compute $N = \mathrm{LE}(\mathcal{P}')$ using (1).
8:  return $\prod_{i \in [\ell]} (a_i!) \cdot N$.
```

Algorithm 3 Algorithm JN.

```
    Input: Poset $\mathcal{P} = (X, \prec)$, where $|X| = n$.
    Output: The minimum jump number of a linear extension of $\mathcal{P}$.
1:  Find a maximum matching $M$ of $\mathsf{C}(\mathcal{P})$ with vertex set $W$.
2:  Let $A = X \setminus W$.
3:  Find $T$ as described, and let $A'$ be the unmatched part of $A$.
4:  Let $\beta = |T|$.
5:  Compute $B = \mathrm{bump}(\mathcal{P})$ by (2), using downsets of $\mathcal{P}$ with at most $\beta n$ vertices from $A'$.
6:  return $n - 1 - B$.
```

References

1 P. D. Amer, C. Chassot, T. J. Connolly, M. Diaz, and P. Conrad. Partial-order transport service for multimedia and other applications. *IEEE/ACM Transactions on Networking*, 2(5):440–456, October 1994. doi:10.1109/90.336326.

2 KA Baker, Peter C Fishburn, and Fred S Roberts. Partial orders of dimension 2, interval orders, and interval graphs, 1970.

3 Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *J. Assoc. Comput. Mach.*, 9:61–63, 1962.

4 G. R. Brightwell, S. Felsner, and W. T. Trotter. Balancing pairs and the cross product conjecture. *Order*, 12(4):327–349, December 1995. doi:10.1007/BF01110378.

5 Graham Brightwell. Balanced pairs in partial orders. *Discrete Mathematics*, 201(1):25–52, 1999. doi:10.1016/S0012-365X(98)00311-2.

6 Graham Brightwell and Peter Winkler. Counting linear extensions. *Order*, 8(3):225–242, September 1991. doi:10.1007/BF00383444.

7 Russ Bubley and Martin Dyer. Faster random generation of linear extensions. *Discrete Mathematics*, 201(1):81–88, 1999. doi:10.1016/S0012-365X(98)00333-1.

8 Stéphan Ceroi. A weighted version of the jump number problem on two-dimensional orders is np-complete. *Order*, 20(1):1–11, 2003.

9 T.M. Cover and J.A. Thomas. *Elements of Information Theory*. A Wiley-Interscience publication. Wiley, 2006.

10 Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as cnf-sat. *ACM Trans. Algorithms*, 12(3):41:1–41:24, May 2016. doi:10.1145/2925416.

11 Karel De Loof, Hans De Meyer, and Bernard De Baets. Exploiting the lattice of ideals representation of a poset. *Fundam. Inf.*, 71(2,3):309–321, February 2006.

12 Samuel Dittmer and Igor Pak. Counting linear extensions of restricted posets, 2018. arXiv:1802.06312.

13 Ben Dushnik and E. W. Miller. Partially ordered sets. *American Journal of Mathematics*, 63(3):600–610, 1941.

14 Martin Dyer, Alan Frieze, and Ravi Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, January 1991. doi:10.1145/102782.102783.

15 E. Eiben, R. Ganian, K. Kangas, and S. Ordyniak. Counting linear extensions: Parameterizations by treewidth. *Algorithmica*, 81(4):1657–1683, April 2019. doi:10.1007/s00453-018-0496-4.

16 Stefan Felsner and Thibault Manneville. Linear extensions of n-free orders. *Order*, 32(2):147–155, 2015. doi:10.1007/s11083-014-9321-0.

17 Stefan Felsner and Lorenz Wernisch. Markov chains for linear extensions, the two-dimensional case. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, 5-7 January 1997, New Orleans, Louisiana, USA.*, pages 239–247, 1997. URL: http://dl.acm.org/citation.cfm?id=314161.314262.

18 Peter C. Fishburn and William V. Gehrlein. A comparative analysis of methods for constructing weak orders from partial orders. *The Journal of Mathematical Sociology*, 4(1):93–102, 1975. doi:10.1080/0022250X.1975.9989846.

19 R. L. Graham, M. Grötschel, and L. Lovász, editors. *Handbook of Combinatorics (Vol. 1)*. MIT Press, Cambridge, MA, USA, 1995.

20 Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *J. Soc. Indust. Appl. Math.*, 10:196–210, 1962.

21 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, December 2001. doi:10.1006/jcss.2001.1774.

22 Gwenaël Joret, Piotr Micek, and Veit Wiechert. Sparsity and dimension. *Combinatorica*, 38(5):1129–1148, October 2018. doi:10.1007/s00493-017-3638-4.

23 Jeff Kahn and Michael Saks. Balancing poset extensions. *Order*, 1(2):113–126, June 1984. doi:10.1007/BF00565647.

24 Kustaa Kangas, Teemu Hankala, Teppo Mikael Niinimäki, and Mikko Koivisto. Counting linear extensions of sparse posets. In *IJCAI 2016*, pages 603–609, 2016. URL: http://www.ijcai.org/Abstract/16/092.

25 Kustaa Kangas, Mikko Koivisto, and Sami Salonen. A faster tree-decomposition based algorithm for counting linear extensions. In *IPEC 2018*, pages 5:1–5:13, 2018. doi:10.4230/LIPIcs.IPEC.2018.5.

26 Donald E. Knuth and Jayme Luiz Szwarcfiter. A structured program to generate all topological sorting arrangements. *Inf. Process. Lett.*, 2(6):153–157, 1974. doi:10.1016/0020-0190(74)90001-5.

27 Dieter Kratsch and Stefan Kratsch. The jump number problem: Exact and parameterized. In *IPEC 2013*, pages 230–242, 2013. doi:10.1007/978-3-319-03898-8_20.

28 Nathan Linial. Hard enumeration problems in geometry and combinatorics. *SIAM J. Algebraic Discrete Methods*, 7(2):331–335, April 1986. doi:10.1137/0607036.

29 L. Lovász. *An Algorithmic Theory of Numbers, Graphs, and Convexity*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, 1986.

30 Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo I. Simari. Probabilistic preference logic networks. In *ECAI 2014*, pages 561–566, 2014. doi:10.3233/978-1-61499-419-0-561.

31 Heikki Mannila and Christopher Meek. Global partial orders from sequential data. In *ACM SIGKDD 2000*, pages 161–168, 2000. doi:10.1145/347090.347122.

32 Jason Morton, Lior Pachter, Anne Shiu, Bernd Sturmfels, and Oliver Wienand. Convex rank tests and semigraphoids. *SIAM J. Discrete Math.*, 23(3):1117–1134, 2009. doi:10.1137/080715822.

33 Rolf Möhring. *Computationally Tractable Classes of Ordered Sets*, pages 105–193. Springer, January 1989. `doi:10.1007/978-94-009-2639-4_4`.

34 Teppo Mikael Niinimäki and Mikko Koivisto. Annealed importance sampling for structure learning in bayesian networks. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 1579–1585, 2013. URL: `http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6885`.

35 J. Scott Provan and Michael O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Comput.*, 12(4):777–788, 1983. `doi:10.1137/0212053`.

36 William R Pulleyblank. *On minimizing setups in precedence constrained scheduling.* Universität Bonn. Institut für Ökonometrie und Operations Research, 1981.

37 I. Rival. *Ordered Sets: Proceedings of the NATO Advanced Study Institute held at Banff, Canada, August 28 to September 12, 1981.* Nato Science Series C:. Springer Netherlands, 2012.

38 Alex Scott and David R. Wood. Better bounds for poset dimension and boxicity. *CoRR*, abs/1804.03271, 2018. `arXiv:1804.03271`.

39 Richard P Stanley. Two poset polytopes. *Discrete & Computational Geometry*, 1(1):9–23, 1986.

40 R.P. Stanley. *Enumerative Combinatorics:*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1997.

41 George Steiner. On estimating the number of order ideals in partial orders, with some applications. *Journal of statistical planning and inference*, 34(2):281–290, 1993.

42 George Steiner and Lorna K Stewart. A linear time algorithm to find the jump number of 2-dimensional bipartite partial orders. *Order*, 3(4):359–367, 1987.

43 R.E. Tarjan. *Data Structures and Network Algorithms.* CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1983.

44 W.T. Trotter. *Combinatorics and partially ordered sets: dimension theory.* Johns Hopkins Series in the Mathematical Sciences. J. Hopkins University Press, 1992.

45 Chris S. Wallace, Kevin B. Korb, and Honghua Dai. Causal discovery via mml. In *ICML 1996*, pages 516–524, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

46 P. Winkler. Average height in a partially ordered set. *Discrete Mathematics*, 39(3):337–341, 1982. `doi:10.1016/0012-365X(82)90157-1`.