

An Efficient Algorithm for Partial Order Production*

Jean Cardinal, Samuel Fiorini, Gwenaël Joret,[†]
 Raphaël M. Jungers,[‡] J. Ian Munro[§]

Abstract

We consider the problem of *partial order production*: arrange the elements of an unknown totally ordered set T into a target partially ordered set S , by comparing a minimum number of pairs in T . Special cases include sorting by comparisons, selection, multiple selection, and heap construction.

We give an algorithm performing $ITLB + o(ITLB) + O(n)$ comparisons in the worst case. Here, n denotes the size of the ground sets, and $ITLB$ denotes a natural information-theoretic lower bound on the number of comparisons needed to produce the target partial order.

Our approach is to replace the target partial order by a weak order (that is, a partial order with a layered structure) extending it, without increasing the information theoretic lower bound too much. We then solve the problem by applying an efficient multiple selection algorithm. The overall complexity of our algorithm is polynomial. This answers a question of Yao (SIAM J. Comput. 18, 1989).

We base our analysis on the entropy of the target partial order, a quantity that can be efficiently computed and provides a good estimate of the information-theoretic lower bound.

Keywords: Partial order, graph entropy

1 Introduction

We consider the PARTIAL ORDER PRODUCTION problem:

Given a set $S = \{s_1, s_2, \dots, s_n\}$ partially ordered by a known partial order \preccurlyeq and a set $T = \{t_1, t_2, \dots, t_n\}$ totally ordered by an unknown linear order \leq , find a permutation π of $\{1, 2, \dots, n\}$ such that $s_i \preccurlyeq s_j \Rightarrow t_{\pi(i)} \leq t_{\pi(j)}$, by asking questions of the form: “is $t_i \leq t_j$?”.

The PARTIAL ORDER PRODUCTION problem generalizes many fundamental problems (see Figure 1), corresponding to specific families of posets $P := (S, \preccurlyeq)$. It amounts to sorting by comparisons when P is a chain. The selection [17] and multiple selection [8] problems are special cases in which P is a weak order¹, that is, has a layered structure (with $a \preccurlyeq b$ iff a is on a lower

*This work was supported by the “Actions de Recherche Concertées” (ARC) fund of the “Communauté française de Belgique”, NSERC of Canada, and the Canada Research Chairs Programme. G.J. and R.J. are Postdoctoral Researchers of the “Fonds National de la Recherche Scientifique” (F.R.S.-FNRS). A preliminary version of the work appeared in [5].

[†]Université Libre de Bruxelles (ULB), Brussels, Belgium. E-mail: {jcardin,sfiorini,gjoret}@ulb.ac.be

[‡]Université catholique de Louvain (UCL), Louvain-la-Neuve, Belgium. E-mail: raphael.jungers@uclouvain.be

[§]University of Waterloo, Waterloo, Ontario, Canada. E-mail: imunro@uwaterloo.ca

¹Most of the terms that are not defined in the introduction are defined in Sections 2 and 3

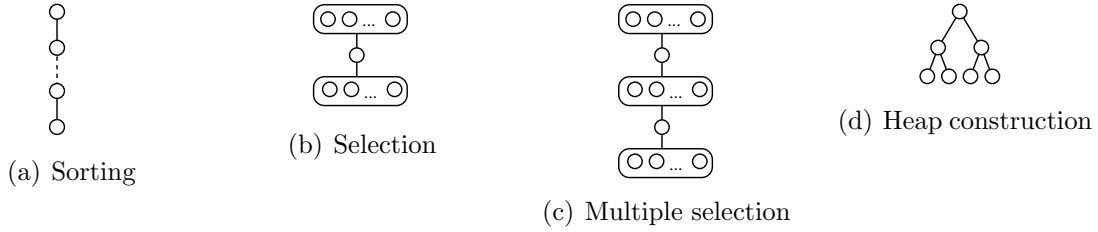


Figure 1: Special cases of the PARTIAL ORDER PRODUCTION problem.

layer than b). When the Hasse diagram of P is a complete binary tree, the problem boils down to heap construction [6].

We assume that the target poset P is part of the input and represented by its Hasse diagram. Hence the size of the input can be $\Omega(n^2)$, whereas sorting the n elements of T takes $O(n \log n)$ time. In other words, reading the input could take more time than necessary to solve problem, provided a topological sorting of P is known.

To cope with this paradoxical situation, we consider algorithms that proceed in two phases: a *preprocessing phase* during which an ordering strategy is determined (for instance, in the form of a decision tree, or any more efficient description, if possible), on the basis of the structure of P , and an *ordering phase* during which all comparisons between elements of T are performed. Accordingly, we distinguish the *preprocessing complexity* and the *ordering complexity* of the algorithm, the latter being essentially proportional to the number of comparisons performed.

As noted before, we expect the overall complexity of the algorithm to be dominated by its preprocessing complexity. Thus it is desirable to perform the preprocessing phase only once, and then use the resulting ordering strategy on several data sets.

Lower bound on the number of comparisons We denote by $e(P)$ the number of linear extensions of the target poset P . Feasible permutations π are in one-to-one correspondence with the linear extensions of P , thus the number of feasible permutations is exactly $e(P)$. On the other hand, the total number of permutations is $n!$. We have thus the following information-theoretic lower bound (logarithms are base 2):

Theorem 1 ([1, 28, 30]). *Any algorithm solving the PARTIAL ORDER PRODUCTION problem for an n -element poset P requires*

$$ITLB := \log n! - \log e(P)$$

comparisons between elements of T in the worst case and on average.

Note that we can assume without loss of generality that P is connected, hence we also have a lower bound of $n - 1$.

Problem history and contribution The PARTIAL ORDER PRODUCTION problem was first proposed in 1976 by Schönhage [28]. It was studied five years after by Aigner [1]. Another four years passed and the problem simultaneously appeared in two survey papers: one by Saks [27] and the other by Bollobás and Hell [2]. In his survey, Saks conjectured that the PARTIAL ORDER PRODUCTION problem can be solved by performing $O(ITLB) + O(n)$ comparisons in the worst case.

Four years later, in 1989, Yao proved Saks' conjecture [30]. He gave an algorithm solving the PARTIAL ORDER PRODUCTION problem in at most $c_1 ITLB + c_2 n$ comparisons, for some constants

c_1 and c_2 . However, the preprocessing phase of Yao’s algorithm seems difficult to implement efficiently. In fact, in the last section of his paper [30], Yao asked whether, assuming P is part of the input (as is the case here), there exists a polynomial-time algorithm for the problem that performs $O(ITLB) + O(n)$ comparisons.

Our main contribution is an algorithm that solves the PARTIAL ORDER PRODUCTION problem and performs at most $ITLB + o(ITLB) + O(n)$ comparisons in the worst case. The preprocessing complexity of our algorithm is $O(n^3)$. Hence we answer affirmatively the question of Yao [30] mentioned above. Moreover, we also significantly improve the ordering complexity, since Yao’s constants c_1 and c_2 are quite large.

Further references, focussing mainly on lower bounds for the problem and generalizations of it include Culberson and Rawlins [12], Chen [9] and Carlsson and Chen [7].

Main ideas underlying our approach We reduce the PARTIAL ORDER PRODUCTION problem to the multiple selection problem. Instead of solving the problem for the given target poset P we solve it for a larger (more constrained) poset that has a simpler structure, namely, a weak order W extending P (a weak order is a set of antichains with a total ordering between these antichains). This approach works because, as we show below, it is possible to find such a weak order W whose corresponding information-theoretic lower bound $ITLB$ is not too large compared to that of P .

Unfortunately, computing $ITLB$ exactly is $\#P$ -hard, because computing the number of linear extensions of a poset is $\#P$ -complete, a result due to Brightwell and Winkler [3]. The analysis is made possible because there exists a quantity, depending on the structure of the target poset, that can be computed in polynomial time and provides a good estimate of $ITLB$. This quantity is nH , where H denotes the entropy of the considered target poset. (The entropy of a graph is defined in the next section, and the entropy of a poset is defined as the entropy of its comparability graph.) It was Körner who introduced the notion of the entropy of a graph, in the context of source coding [21]. The idea of estimating an information-theoretic lower bound by means of the entropy of a poset was used before by Kahn and Kim in their inspiring work on sorting with partial information [18], see below.

Related problems In 1971 Chambers [8] proposed an algorithm for the PARTIAL SORTING problem, defined as follows: given a vector V of n numbers and a set $I \subseteq \{1, 2, \dots, n\}$ of indices, rearrange the elements of V so that for every $i \in I$, all elements with indices $j < i$ are smaller or equal to V_i , and elements with indices $j > i$ are bigger or equal to V_i . For the indices $i \in I$, the elements V_i in the rearranged vector have rank exactly i , hence this problem is also called *multiple selection*. The PARTIAL SORTING problem is a special case of PARTIAL ORDER PRODUCTION in which the partial order is also a weak order.

The algorithm proposed by Chambers is similar to Hoare’s “find” algorithm [17], or QuickSelect. It has been refined and analyzed by Dobkin and Munro [13], Panholzer [25], Prodinger [26], and Kaligosi, Mehlhorn, Munro, and Sanders [20]. For our purposes, the key result is that of Kaligosi *et al.* [20] in which it is shown that multiple selection can be done within a lower order term of the information theoretic lower bound, plus a linear term.

Another generalization of the sorting problem, called SORTING WITH PARTIAL INFORMATION, was studied by Kahn and Kim [18]:

Given an unknown linear order \leq on a set $T = \{t_1, \dots, t_n\}$, together with a subset \preceq of the relations $t_i \leq t_j$ forming a partial order, determine the complete linear order \leq by asking questions of the form: “is $t_i \leq t_j$?”.

This problem is equivalent to sorting by comparisons if \preceq is empty. The information-theoretic lower bound for that problem is $\log e(Q)$, where $Q := (T, \preceq)$. The problem is complementary to the PARTIAL ORDER PRODUCTION problem in the sense that sorting by comparisons can be achieved by first solving a PARTIAL ORDER PRODUCTION problem, then solving the SORTING WITH PARTIAL INFORMATION problem on the output.

A proof that there exists a decision tree achieving the lower bound up to a constant factor has been known for some time (see in particular Kahn and Saks [19]). This is related to the $1/3$ - $2/3$ conjecture of Fredman [14] and Linial [22]. Kahn and Kim [18] provided a polynomial time algorithm that finds the actual comparisons. They show that choosing the comparison that causes the entropy of Q to increase the most leads to a decision tree that is near-optimal in the above sense.

Overview In Section 2, we study the entropy of perfect graphs. We show that it is possible to approximate the entropy of a perfect graph G using a simple greedy coloring algorithm. More precisely, we prove that any such approximation is at most $H(G) + \log(H(G) + 1) + O(1)$, where $H(G)$ denotes the entropy of graph G .

Section 3 explains how to apply this result to solve the PARTIAL ORDER PRODUCTION problem algorithmically. We begin the section by remarking that entropy is bound to play a central role for the problem since $nH(P) - n \log e \leq ITLB \leq nH(P)$, where $H(P)$ denotes the entropy of poset P .

The preprocessing phase of our algorithm starts by applying the greedy coloring algorithm studied in Section 2 to the comparability graph of P . We then modify this coloring (we “uncross” the colors) in order to obtain an extension of P which is an interval order I . Another application of the greedy coloring algorithm, this time on the comparability graph of I , yields a weak order W extending I . Using our result on perfect graphs, we prove that the entropy of W is not much larger than that of P , that is, $H(W) \leq H(P) + 2 \log(H(P) + 1) + O(1)$.

The ordering phase of the algorithm simply runs then a multiple selection algorithm based on the weak order W . We use a multiple selection algorithm from Kaligosi *et al.* [20] that performs a number of comparisons close to the information-theoretic lower bound.

We conclude the section by proving that the preprocessing complexity of our algorithm is $O(n^3)$.

Finally, in Section 4, we discuss the number of comparisons and study the existence of an algorithm solving the PARTIAL ORDER PRODUCTION problem in $ITLB + O(n)$ comparisons. We give an example showing that such an algorithm cannot always reduce the problem to the case where the target poset is a weak order. More specifically, we exhibit a family of interval orders with entropy at most $\frac{1}{2} \log n$, any weak order extension of which has entropy at least $\frac{1}{2} \log n + \Omega(\log \log n)$.

2 Entropy of Perfect Graphs

We recall that a subset S of vertices of a graph is a *stable set* (or *independent set*) if the vertices in S are pairwise nonadjacent. Also, a graph G is *perfect* if $\omega(H) = \chi(H)$ holds for every induced subgraph H of G , where $\omega(H)$ and $\chi(H)$ denote the clique and chromatic numbers of H , respectively.

Let us recall similarly that the *stable set polytope* of an arbitrary graph G with vertex set V and order n is the n -dimensional polytope

$$\text{STAB}(G) := \text{conv}\{\chi^S \in \mathbb{R}^V : S \text{ stable set in } G\},$$

where χ^S is the characteristic vector of the subset S , assigning the value 1 to every vertex in S , and 0 to the others. The *entropy* of G is defined as (see [11, 21])

$$H(G) := \min_{x \in \text{STAB}(G)} -\frac{1}{n} \sum_{v \in V} \log x_v. \quad (1)$$

For example, if $G = (V, E)$ is the graph with $V := \{a, b, c\}$ and $E := \{bc\}$, then $H(G) = 2/3$ and the minimum in (1) is attained for $x = (x_a, x_b, x_c) = (1, 1/2, 1/2)$.

Note that graph entropy was originally defined with respect to a given probability distribution on V . However, for our purposes we can take the uniform distribution, as in [18]. In this case we obtain Equation (1).

An upper bound on $H(G)$ can be found as follows: First, use the greedy coloring algorithm that removes iteratively a maximum stable set from G , giving a sequence S_1, S_2, \dots, S_k of stable sets of G . If G is perfect, this can be done in polynomial time (see, e.g., Grötschel, Lovász and Schrijver [16]). Next, let $\tilde{x} \in \mathbb{R}^V$ be defined as

$$\tilde{x} := \sum_{i=1}^k \frac{|S_i|}{n} \cdot \chi^{S_i}.$$

By definition, $\tilde{x} \in \text{STAB}(G)$. We call any such point \tilde{x} a *greedy point*. The value of the objective function in the definition of $H(G)$ for \tilde{x} is $\sum_{i=1}^k \frac{|S_i|}{n} \log \frac{n}{|S_i|}$. We refer to the latter quantity simply as the entropy of \tilde{x} . It turns out that this gives a good approximation of $H(G)$ when G is a perfect graph.

Theorem 2. *Let G be a perfect graph on n vertices and denote by \tilde{g} the entropy of an arbitrary greedy point $\tilde{x} \in \text{STAB}(G)$. Then*

$$\tilde{g} \leq \frac{1}{1-\delta} \left(H(G) + \log \frac{1}{\delta} \right)$$

for all $\delta > 0$, and in particular

$$\tilde{g} \leq H(G) + \log(H(G) + 1) + O(1).$$

A key tool in our proof of Theorem 2 is a min-max relation of Csiszár, Körner, Lovász, Marton, and Simonyi [11] relating the entropy of a perfect graph G to the entropy of its complement \bar{G} :

Theorem 3 ([11]). *If G is a perfect graph on n vertices, then $H(G) + H(\bar{G}) = \log n$.*

We now turn to the proof of Theorem 2.

Proof of Theorem 2. Let S_1, S_2, \dots, S_k be the sequence of stable sets of G selected by the greedy algorithm (in the order the algorithm removes them). So S_1 is a maximum stable set in G , S_2 is a maximum stable set in $G - S_1$, and so on. The outline of the proof is as follows: We first use the sets S_1, S_2, \dots, S_k to define a point $z \in \mathbb{R}^V$, where V is the vertex set of G . We then show that z belongs to the stable set polytope of the *complement* \bar{G} of G , that is, $z \in \text{STAB}(\bar{G})$. Finally, we derive the desired inequality by combining the upper bound on $H(\bar{G})$ implied by z with Theorem 3.

Fix $\delta > 0$. For each vertex $v \in V$ we let $m = m(v)$ be the unique index in $\{1, \dots, k\}$ such that $v \in S_m$. We define z by letting, for each vertex v of G ,

$$z_v := \frac{\delta}{n} \left(\frac{1}{\tilde{x}_v} \right)^{1-\delta} = \frac{\delta}{n} \left(\frac{n}{|S_{m(v)}|} \right)^{1-\delta} = \frac{\delta}{n^\delta} \left(\frac{1}{|S_{m(v)}|} \right)^{1-\delta}.$$

We claim that for every stable set S of G :

$$\sum_{v \in S} z_v \leq 1. \quad (2)$$

Write the stable set S as $S = T_1 \cup T_2 \cup \dots \cup T_\ell$, where T_i is the i th subset of S taken by the greedy algorithm during its execution. For every $v \in T_1$, we have $S_{m(v)} = S_1$, and $|S_{m(v)}| \geq |S|$, since the greedy algorithm could have selected the set S when it took $S_{m(v)}$. More generally, for every $i \in \{1, 2, \dots, \ell\}$ and $v \in T_i$, we have $|S_{m(v)}| \geq |S| - \sum_{j=1}^{i-1} |T_j|$. It follows in particular that we can enumerate the points of S as v_1, v_2, \dots, v_s in such a way that

$$|S_{m(v_i)}| \geq |S| - i + 1 \quad \forall i \in \{1, 2, \dots, s\}.$$

We thus have

$$\begin{aligned} \sum_{v \in S} z_v &\leq \frac{\delta}{n^\delta} \left(\left(\frac{1}{|S|} \right)^{1-\delta} + \left(\frac{1}{|S|-1} \right)^{1-\delta} + \dots + 1 \right) \\ &\leq \frac{\delta}{n^\delta} \left(\int_0^{|S|} \frac{1}{x^{1-\delta}} dx \right) \\ &\leq 1. \end{aligned}$$

Equation (2) follows.

Two classical results on perfect graphs are that the stable set polytope is completely described by the non-negativity and clique inequalities, that is,

$$\text{STAB}(G) = \{x \in \mathbb{R}_+^V : \sum_{v \in K} x_v \leq 1 \quad \forall K \text{ clique in } G\}$$

(see Chvátal [10]), and that the complement \bar{G} of G is also a perfect graph (Lovász [23]). Combining these two results with (2) shows that $z \in \text{STAB}(\bar{G})$. Using Theorem 3, we then deduce

$$\begin{aligned} H(G) &= \log n - H(\bar{G}) \\ &\geq \log n + \frac{1}{n} \sum_{v \in V} \log z_v \\ &= \log n + \frac{1}{n} \sum_{v \in V} \log \left(\frac{\delta}{n} \left(\frac{1}{\tilde{x}_v} \right)^{1-\delta} \right) \\ &= -\frac{1-\delta}{n} \sum_{v \in V} \log \tilde{x}_v - \log \frac{1}{\delta} \\ &= (1-\delta)\tilde{g} - \log \frac{1}{\delta}. \end{aligned}$$

Hence, $\tilde{g} \leq \frac{1}{1-\delta} (H(G) + \log \frac{1}{\delta})$, for all $\delta > 0$. By choosing $\delta = 1/2$ if $H(G) \leq 1$, and $\delta = 1/(H(G) + 1)$ otherwise, we obtain $\tilde{g} \leq H(G) + \log(H(G) + 1) + O(1)$. \square

3 An Algorithm for Partial Order Production

We denote by $G(P)$ the comparability graph of a poset $P = (V, \leq_P)$, and let $H(P) := H(G(P))$. Note that a stable set in $G(P)$ is an antichain in P , that is, a set of mutually incomparable elements. Note also that $G(P)$ is perfect, a basic result that is dual to Dilworth's theorem, see, e.g., [15]. The relevance of the notion of graph entropy in the context of sorting was first observed by Kahn and Kim [18]. Using the fact that the volume of $\text{STAB}(G(P))$ equals $e(P)/n!$ (see Stanley [29]), they proved the following result.

Lemma 1 ([18]). *For any poset P of order n ,*

$$-nH(P) \leq \log e(P) - \log n! \leq n \log n - \log n! - nH(P).$$

When written as

$$2^{-nH(P)} \leq \frac{e(P)}{n!} \leq 2^{-nH(P)} \cdot \frac{n^n}{n!},$$

the above inequalities become intuitively clear, since $2^{-nH(P)}$ is the (maximum) volume of a box contained in $\text{STAB}(G(P))$, $e(P)/n!$ is the volume of $\text{STAB}(G(P))$, and $2^{-nH(P)} \cdot n^n/n!$ is the volume of a simplex containing $\text{STAB}(G(P))$. The lemma directly implies the following equality for every poset P :

$$ITLB = \log n! - \log e(P) = nH(P) + O(n). \quad (3)$$

We recall that a poset is said to be a *weak order* whenever its comparability graph is a complete k -partite graph, for some k . Such a poset $W = (V, \leq_W)$ can be partitioned into k maximal antichains A_1, \dots, A_k , the *layers* of W , such that $v <_W w$ whenever there exist indices i and j such that $v \in A_i$, $w \in A_j$ and $i < j$. When restricted to weak orders, the PARTIAL ORDER PRODUCTION problem resembles the PARTIAL SORTING problem, with $I = \{\sum_{j=1}^i |A_j| : i = 1, \dots, k-1\}$.

Our key idea is to show that, using (twice) the greedy coloring algorithm presented in the previous section, we can efficiently extend² the given poset P to a weak order W whose entropy is close to that of P . The reason why we have to use twice the greedy algorithm is that the obtained coloration might not be "ordered" (might not represent the stable sets of a weak order). However, we describe below how to uncross this coloring in order to extend P to an interval order without increasing too much the entropy. We show that applying our greedy coloring to an interval order provides an "ordered" coloring, which allows us to run a second time our greedy algorithm, providing an extension which is a weak order.

We then simply run an efficient multiple selection procedure, with W as input. We show that, because replacing P by W does not increase the entropy too much, the resulting number of comparisons is close to $ITLB$.

The preprocessing phase is composed of three steps, each of which can be performed in polynomial time. In the first step, we apply the greedy coloring procedure to $G(P)$, to obtain a greedy point \tilde{x} . This step makes use of an auxiliary network defined from P . Then, in the second step, using again the auxiliary network, we extend P to an interval order I whose entropy is not larger than that of \tilde{x} . This allows us to "uncross" the antichains used in \tilde{x} . (An alternative way of obtaining the interval order I is to apply Kahn and Kim's [18] laminar decomposition lemma to \tilde{x} .) Finally, in the third step, we apply the greedy coloring procedure again, this time on $G(I)$, to obtain the weak order W . See Figure 2 for an illustration of steps 1 and 2.

²A poset Q *extends* a poset P if they have the same ground set V and $v \leq_P w$ implies $v \leq_Q w$, for all $v, w \in V$.



in P). Then

$$\begin{aligned}
\sum_{v \in C} x_v &= (y_{v_1^+} - y_{v_1^-}) + \cdots + (y_{v_c^+} - y_{v_c^-}) \\
&\leq (y_{v_1^-} - y_s) + (y_{v_1^+} - y_{v_1^-}) + (y_{v_2^-} - y_{v_1^+}) + \cdots + (y_{v_c^+} - y_{v_c^-}) + (y_t - y_{v_c^+}) \\
&= y_t - y_s = 1.
\end{aligned}$$

It follows that $x \in \text{STAB}(G)$.

For necessity, consider $x \in \text{STAB}(G)$. For $v \in V$, we let y_{v+} be the maximum total weight of a chain of P whose maximum with respect to \leq_P is v , when each vertex w is given the weight x_w , and $y_{v-} := y_{v+} - x_v$. Then we let $y_s := 0$ and $y_t := 1$. As is easily verified, y is a potential for x . \square

It follows that $H(P)$ is the optimum value of the following convex minimization problem with a polynomial number of variables and constraints:

$$\begin{aligned}
(\text{H-potential}) \quad \min \quad & -\frac{1}{n} \sum_{v \in V} \log x_v \\
\text{s.t.} \quad & x_v = y_{v+} - y_{v-} \quad \forall v \in V \\
& y_p \leq y_q \quad \forall (p, q) \in A(D) \\
& y_s = 0 \\
& y_t = 1.
\end{aligned}$$

We remark that this formulation shows that $H(P)$ can be computed to within any fixed precision in strongly polynomial time, using interior point methods (see for instance [24]). However, approximating $H(P)$ using a greedy point will be enough for our purposes, and will moreover give a better upper bound on the complexity of our algorithm.

Greedy extensions Let \tilde{x} be a greedy point in $\text{STAB}(G)$, as defined in Section 2. Consider the potential $\tilde{y} \in \mathbb{R}^{N(D)}$ defined from \tilde{x} as in the proof of Lemma 2: For $v \in V$, we let \tilde{y}_{v+} be the maximum (total) weight of a chain of P ending in v , where each vertex w has weight \tilde{x}_w , and $\tilde{y}_{v-} := \tilde{y}_{v+} - \tilde{x}_v$. Let also $\tilde{y}_s := 0$ and $\tilde{y}_t := 1$.

From this potential \tilde{y} , we compute an interval order I extending P whose entropy is not larger than that of \tilde{x} . The ground set of I is V . We let $v \leq_I w$ whenever $\tilde{y}_{v+} \leq \tilde{y}_{w-}$. Thus the open intervals (y_{v-}, y_{v+}) (for $v \in V$) provide an interval representation of I . Because $v \leq_P w$ implies $\tilde{y}_{v+} \leq \tilde{y}_{w-}$, which in turn implies $v \leq_I w$, the interval order I extends P . The entropy of I is not larger than that of \tilde{x} because (\tilde{x}, \tilde{y}) remains feasible for the minimization problem (H-potential) defined above, after P is replaced by I .

Apply again the greedy coloring algorithm, but now on $G(I)$. Let A_1, \dots, A_k denote the antichains of I produced by the greedy coloring algorithm. Because I is an interval order, we can find a permutation σ of $\{1, \dots, k\}$ such that $v <_I w$, $v \in A_{\sigma(i)}$ and $w \in A_{\sigma(j)}$ imply $i < j$. Thus, the weak order W with ground set V obtained by setting $v <_W w$ whenever $v \in A_{\sigma(i)}$ and $w \in A_{\sigma(j)}$ with $i < j$ is an extension of I . Such a weak order W is said to be a *greedy extension* of the original poset P .

Lemma 3. *Let P be a poset and W one of its greedy extensions. Then*

$$H(W) \leq \frac{1}{1-\delta} \left(H(P) + 2 \log \frac{1}{\delta} + 2 \right)$$

for all $\delta > 0$, and in particular

$$H(W) \leq H(P) + 2\log(H(P) + 1) + O(1).$$

Proof. Let $\delta' := \delta/2$. Let I denote the intermediate interval order used to obtain W . Theorem 2 implies

$$\begin{aligned} H(P) &\geq (1 - \delta')H(I) - \log(1/\delta') \\ &\geq (1 - \delta')((1 - \delta')H(W) - \log(1/\delta')) - \log(1/\delta') \\ &\geq (1 - \delta)H(W) - 2\log(1/\delta) - 2. \end{aligned}$$

In addition to Theorem 2, for the first inequality we used the fact that $H(I) \leq \tilde{g}$, and for the second one, we used the fact that the greedy coloring of I directly gives the unique decomposition of W in maximal stable sets. This shows the first part of the claim. For the second part, again take $\delta = 1/2$ if $H(P) \leq 1$, and $\delta = 1/(H(P) + 1)$ otherwise. \square

Algorithm and complexity The above results directly suggest the following algorithm: compute a greedy extension W of P , and run a multiple selection procedure on T with respect to W . In terms of the number of comparisons between elements of T , we only incur a controlled penalty.

Theorem 4. *The PARTIAL ORDER PRODUCTION problem can be solved in polynomial time using at most*

$$ITLB + o(ITLB) + O(n) \tag{4}$$

comparisons between elements of T in the worst case.

Proof. The weak order extension W can be computed in polynomial time. Let us denote by A_1, \dots, A_k its layers. We run the multiple selection algorithm on the elements of T , with the ranks $r_i := \sum_{j=1}^i |A_j|$ (for $i = 1, \dots, k-1$). Kaligosi *et al.* [20] give a multiple selection algorithm that requires only $B + o(B) + O(n)$ comparisons in the worst case, where $B := \log n! - \log e(W)$ is the information-theoretic lower bound for W . Thus

$$\begin{aligned} B &= nH(W) + O(n) && \text{(from Eqn. (3))} \\ &\leq nH(P) + 2n\log(H(P) + 1) + O(n) && \text{(from Lemma 3)} \\ &= ITLB + 2n\log\left(\frac{ITLB}{n} + 1\right) + O(n) && \text{(from Eqn. (3))} \\ &= ITLB + o(ITLB) + O(n). \end{aligned}$$

Hence $B + o(B) + O(n) = ITLB + o(ITLB) + O(n)$, and the theorem follows. \square

We conclude the section by discussing the preprocessing complexity of our algorithm.

The first execution of the greedy coloring algorithm can be done in time $O(mn)$, where m is the number of arcs in the network $D := D(P)$ (notice $m \geq n$ and $m = O(n^2)$), as we now briefly explain. The algorithm finds maximal antichains in the graph by decrementing a flow on the auxiliary network. This flow has to satisfy lower bounds on the arcs.

Let $X := \emptyset$, $i := 1$, and put a lower bound of $\ell_a := 1$ on each arc a of the form (v^-, v^+) with $v \in V$, of $\ell_a := 0$ on every other arc a of D . Start with an arbitrary integer s - t flow ϕ of value n such that $\phi_a \geq \ell_a$ for every arc $a \in A(D)$. Let Y be the set of nodes of D that can be reached from s following a *decrementing path*, namely, a path $v_0 v_1 \dots v_k$ with $v_0 := s$ such that, for every

$i \in \{1, 2, \dots, k\}$, either $(v_{i-1}, v_i) \in A(D)$ and $\phi_{(v_{i-1}, v_i)} > \ell_{(v_{i-1}, v_i)}$, or $(v_i, v_{i-1}) \in A(D)$. Now, there are two cases: (1) $t \in Y$. Thus there exists a decrementing s - t path. We then decrement by 1 the flow value of ϕ using the latter path. (2) $t \notin Y$. Observe that no arc of D enters the set Y and that the arcs a going out of Y satisfy $\phi_a = \ell_a$. It follows that

$$A_i := \{v \in V \mid (v^-, v^+) \in \delta^+(Y), \phi_{(v^-, v^+)} = 1\}$$

is an antichain of $P - X$. (Here, $\delta^+(Y)$ denotes the set of arcs of D going out of Y .) Moreover, since the flow value of ϕ equals $|A_i|$, the antichain A_i is maximum among the antichains of $P - X$. This is because, by definition of our lower bounds, the flow value is at least $|A|$, for every antichain A contained in $P - X$. We then let $\ell_{(v^-, v^+)} := 0$ for every $v \in A_i$, set $X := X \cup A_i$, increment i by 1, and repeat the above steps, until $X = V$. Computing the set Y , decrementing the flow, and finding the antichain A_i are steps that can be done in time $O(m)$. Since we go through the main loop at most $2n$ times, this implementation of the greedy algorithm runs in time $O(nm)$.

The greedy point \tilde{x} can be computed in time $O(n)$. The corresponding potential \tilde{y} can be found in $O(m)$ using a simple dynamic program. The second execution of the greedy coloring algorithm can be done in time $O(n^2)$, using the fact that the comparability graph of the interval order I is a co-interval graph. Finally, a bound on the complexity of the multiple selection procedure is $O(n^2)$. So the whole algorithm runs in $O(nm) = O(n^3)$.

4 Tightness

A natural question is whether there exists an algorithm for PARTIAL ORDER PRODUCTION that does at most $ITLB + O(n)$ comparisons between elements of T . We show in this section that every algorithm that first extends the target poset to a weak order and then solves the problem on the weak order can be forced to make $ITLB + \Omega(n \log \log n)$ comparisons, both in the worst case and the average case. This is a consequence of the following theorem:

Theorem 5. *There exists a constant $c > 0$ such that, for all $n \geq 1$, there is a poset P on n elements satisfying $H(W) \geq H(P) + c \log \log n$ for every weak order W extending P .*

In order to prove Theorem 5, we define a family $\{G_k\}$ ($k \geq 1$) of interval graphs inductively as follows:

- G_1 consists of a unique vertex, and
- for $k \geq 2$, the graph G_k is obtained by first taking the disjoint union of $K_{2^{k-1}}$ (the “central clique”) with two copies of G_{k-1} , and then making half of the vertices of the central clique adjacent to all vertices in the first copy, and the other half to all those in the second copy.

It is easily seen that G_k is indeed an interval graph, as is suggested in Figure 3. The graph G_k has $k2^{k-1}$ vertices. The complement \bar{G}_k of G_k is the comparability graph of the interval order I_k defined by an interval representation of G_k .

Lemma 4. $H(I_k) \leq (k+1)/2$.

Proof. By construction, the maximal stable sets of the graph \bar{G}_k all have 2^{k-1} elements, and there are $2^k - 1$ such maximal stable sets. We define a point $x^{(k)}$ of the stable set polytope $\text{STAB}(\bar{G}_k)$ as follows:

$$x^{(k)} := \sum_{i=1}^{2^k-1} \frac{1}{2^k-1} \chi^{S_i},$$

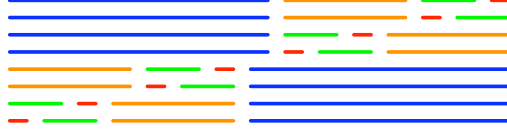


Figure 3: An interval representation of G_4 (colors highlight the different levels of the construction).

where $S_1, S_2, \dots, S_{2^{k-1}}$ are the maximal stable sets of \bar{G}_k . Observe that, for every $\ell \in \{0, \dots, k-1\}$, there are 2^{k-1} vertices in \bar{G}_k that belong to exactly 2^ℓ different maximal stable sets (that is, there are 2^{k-1} intervals of each different length in the interval representation suggested in Figure 3). We thus obtain the following upper bound on the entropy of I_k :

$$H(I_k) \leq -\frac{1}{k2^{k-1}} \sum_{\ell=0}^{k-1} 2^{k-1} \log \frac{2^\ell}{2^k - 1} = \log(2^k - 1) - \frac{k-1}{2} \leq (k+1)/2.$$

The lemma follows. \square

We proceed by showing that every weak order extension of I_k has relatively large entropy compared to I_k . We first introduce some definitions. Consider an arbitrary graph G and a coloring C_1, \dots, C_ℓ of its vertices. Similarly as how greedy points are defined (see Section 2), one can associate an entropy to the latter coloring, namely, the entropy of the probability distribution $\{|C_i|/n\}_{i=1, \dots, \ell}$:

$$-\sum_{i=1}^{\ell} \frac{|C_i|}{n} \log \frac{|C_i|}{n}.$$

The minimum entropy of a coloring is known as the *chromatic entropy* of G , and is denoted by $H_\chi(G)$. The chromatic entropy can be thought of as a constrained version of the graph entropy, in which the stable sets involved in the definition of $H(G)$ are required to form a partition of the vertices of G .

Lemma 5. *Let G be the comparability graph of a poset P . Then any weak order extension W of P has entropy $H(W) \geq H_\chi(G)$.*

Proof. The maximal antichains of W are pairwise disjoint, hence they correspond to a coloring of G . The entropy of W is equal to the entropy of the latter coloring, and thus is at least $H_\chi(G)$. \square

Lemma 5 suggests finding a (good) lower bound on $H_\chi(\bar{G}_k)$, the chromatic entropy of \bar{G}_k . To achieve that, we make use of the following result of [4] (see Corollary 1 in that paper).

Theorem 6 ([4]). *Let G be an arbitrary graph. Then the entropy of any coloring of G produced by the greedy coloring algorithm is at most $H_\chi(G) + \log e$.*

We can therefore restrict ourselves to analyzing the entropy of greedy colorings of \bar{G}_k . Recall that all maximal stable sets in \bar{G}_k have the same cardinality 2^{k-1} . Consider the greedy coloring of \bar{G}_k defined recursively as follows: take first the stable set of cardinality 2^{k-1} that corresponds to the central clique in G_k , and then, if $k \geq 2$, recurse on the two copies of \bar{G}_{k-1} that are left. Let \tilde{g}_k denote the entropy of the resulting coloring of \bar{G}_k .

Lemma 6. $\tilde{g}_k = (k-1)/2 + \log k$.

Proof. The greedy coloring defined above consists of 2^{i-1} color classes of cardinality 2^{k-i} , for $i = 1, 2, \dots, k$. Hence, its entropy is

$$\begin{aligned}\tilde{g}_k &= - \sum_{i=1}^k 2^{i-1} \cdot \frac{2^{k-i}}{k2^{k-1}} \log \frac{2^{k-i}}{k2^{k-1}} \\ &= \frac{1}{k} \sum_{i=1}^k \log \frac{k2^{k-1}}{2^{k-i}} \\ &= \frac{1}{k} \sum_{i=1}^k (\log k + (i-1)) \\ &= \log k + \frac{k-1}{2},\end{aligned}$$

as claimed. □

We may now turn to the proof of Theorem 5.

Proof of Theorem 5. Let $k \geq 1$ and consider the interval order I_k defined above, of order $n := k2^{k-1}$. Let also W be an arbitrary weak order extending I_k . Combining Lemmata 4, 5 and 6 with Theorem 6 gives

$$\begin{aligned}H(W) - H(I_k) &\geq H_\chi(\bar{G}_k) - H(I_k) \\ &\geq \left(\frac{k-1}{2} + \log k - \log e \right) - \frac{k+1}{2} \\ &= \log k - \log e - 1 \\ &= \Omega(\log \log n),\end{aligned}$$

as claimed. □

Acknowledgments The authors wish to thank Sébastien Collette, François Glineur and Stefan Langerman for useful discussions, and the anonymous referees for their comments on an earlier version of the paper.

References

- [1] M. Aigner. Producing posets. *Discrete Math.*, 35:1–15, 1981.
- [2] B. Bollobás and P. Hell. Sorting and graphs. In *Graphs and order, Banff, Alta., 1984*, volume 147 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 169–184, Dordrecht, 1985. Reidel.
- [3] G. Brightwell and P. Winkler. Counting linear extensions. *Order*, 8(3):225–242, 1991.
- [4] J. Cardinal, S. Fiorini, and G. Joret. Tight results on minimum entropy set cover. *Algorithmica*, 51(1):49–60, 2008.
- [5] J. Cardinal, S. Fiorini, G. Joret, R. M. Jungers, and J. I. Munro. An Efficient Algorithm for Partial Order Production. To appear in *Proceedings of STOC 09, Bethesda (Maryland), United States*, 2009.

- [6] S. Carlsson and J. Chen. The complexity of heaps. In *Proceedings of the third annual ACM-SIAM symposium on discrete algorithms (SODA '92), Orlando (Florida), United States*, pages 393–402, Philadelphia, PA, USA, 1992. Society for Industrial and Applied Mathematics.
- [7] S. Carlsson and J. Chen. Some lower bounds for comparison-based algorithms. In *Proc. 2nd European Symposium on Algorithms (ESA '94), Utrecht, The Netherlands*, volume 855 of *Lecture Notes in Computer Science*, pages 106–117. Springer-Verlag, 1994.
- [8] J. M. Chambers. Partial sorting (algorithm 410). *Commun. ACM*, 14(5):357–358, 1971.
- [9] J. Chen. Average cost to produce partial orders. In *Proc. 5th International Symposium on Algorithms and Computation (ISAAC '94), Beijing, P. R. China*, volume 834 of *Lecture Notes in Computer Science*, pages 155–163. Springer-Verlag, 1994.
- [10] V. Chvátal. On certain polytopes associated with graphs. *J. Combinatorial Theory Ser. B*, 18:138–154, 1975.
- [11] I. Csiszár, J. Körner, L. Lovász, K. Marton, and G. Simonyi. Entropy splitting for antiblocking corners and perfect graphs. *Combinatorica*, 10(1):27–40, 1990.
- [12] J. C. Culberson and G. J. E. Rawlins. On the comparison cost of partial orders. Technical Report TR88-01, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada T6G 2E8, 1988.
- [13] D. P. Dobkin and J. I. Munro. Optimal time minimal space selection algorithms. *J. ACM*, 28(3):454–461, 1981.
- [14] M. L. Fredman. How good is the information theory bound in sorting? *Theor. Comput. Sci.*, 1(4):355–361, 1976.
- [15] M. C. Golumbic. *Algorithmic graph theory and perfect graphs. 2nd ed.*, volume 57 of the *Annals of Discrete Mathematics*. Elsevier, Amsterdam, 2004.
- [16] M. Grötschel, L. Lovász and A. Schrijver. *Geometric algorithms and combinatorial optimization. 2nd corr. ed.*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1993.
- [17] C. A. R. Hoare. Find (algorithm 65). *Commun. ACM*, 4(7):321–322, 1961.
- [18] J. Kahn and J. H. Kim. Entropy and sorting. *J. Comput. Syst. Sci.*, 51(3):390–399, 1995.
- [19] J. Kahn and M. E. Saks. Balancing poset extensions. *Order*, 1:113–126, 1984.
- [20] K. Kaligosi, K. Mehlhorn, J. I. Munro, and P. Sanders. Towards optimal multiple selection. In *Proc. International Conference on Automata, Languages, and Programming (ICALP'05), Lecture Notes in Computer Science*, pages 103–114. Springer-Verlag, 2005.
- [21] J. Körner. Coding of an information source having ambiguous alphabet and the entropy of graphs. In *Transactions of the 6th Prague Conference on Information Theory*, pages 411–425, 1973.
- [22] N. Linial. The information-theoretic bound is good for merging. *SIAM J. Comput.*, 13(4):795–801, 1984.

- [23] L. Lovász. Normal hypergraphs and the perfect graph conjecture. *Discrete Math.*, 2(3):253–267, 1972.
- [24] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13 of *SIAM Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.
- [25] A. Panholzer. Analysis of multiple quickselect variants. *Theor. Comput. Sci.*, 302(1-3):45–91, 2003.
- [26] H. Prodinger. Multiple quickselect – Hoare’s find algorithm for several elements. *Inf. Process. Lett.*, 56:123–129, 1995.
- [27] M. E. Saks. The information theoretic bound for problems on ordered sets and graphs. In *Graphs and order, Banff, Alta., 1984*, volume 147 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 137–168, Dordrecht, 1985. Reidel.
- [28] A. Schönhage. The production of partial orders. In *Journées algorithmiques, École Norm. Sup., Paris, 1975*, pages 229–246. Astérisque, No. 38–39. Soc. Math. France, Paris, 1976.
- [29] R. P. Stanley. Two poset polytopes. *Discrete Comput. Geom.*, 1:9–23, 1986.
- [30] A. C. Yao. On the complexity of partial order productions. *SIAM J. Comput.*, 18(4):679–689, 1989.