ON-LINE CHAIN PARTITIONS OF UP-GROWING SEMI-ORDERS

STEFAN FELSNER, KAMIL KLOCH, GRZEGORZ MATECKI, AND PIOTR MICEK

Abstract. On-line chain partition is a two-player game between Spoiler and Algorithm. Spoiler presents a partially ordered set, point by point. Algorithm assigns incoming points (immediately and irrevocably) to the chains which constitute a chain partition of the order. The value of the game for orders of width w is a minimum number $\mathrm{val}(w)$ such that Algorithm has a strategy using at most $\mathrm{val}(w)$ chains on orders of width at most w. We analyze the chain partition game for up-growing semi-orders. Surprisingly, the golden ratio comes into play and the value of the game is $\lfloor \frac{1+\sqrt{5}}{2}\, w \rfloor$.

1. Introduction

On-line chain partitions of an order can be described as a two-person game between Algorithm and Spoiler. The game is played in rounds. Spoiler presents an on-line order, one point at a time. Algorithm responds by making an irrevocable assignment of the new point to one of the chains of the chain partition. The performance of Algorithm's strategy is measured by comparing the number of chains used with the number of chains of an optimal chain partition. By Dilworth's Theorem the size of an optimal chain partition equals the width of the order. The value of the game for orders of width w, denoted by $\mathrm{val}(w)$, is the least integer n for which some Algorithm has a strategy using at most n chains for every on-line order of width w. Alternatively, it is the largest integer n for which Spoiler has a strategy that forces any Algorithm to use n chains on order of width w.

The study of chain partition games goes back to the early 80's when Kierstead [4] (upper bound) and Szemerédi (lower bound published in [5]) proved the estimates for on-line orders of width w: $\binom{w+1}{2} \leqslant \mathrm{val}(w) \leqslant \frac{5^w-1}{4}$. It took almost 30 years until these bounds had been slightly improved. The story can be found in the survey [2].

The study of on-line chain partition on restricted classes of orders began in 1981 when Kierstead and Trotter [6] proved the following result: when Spoiler is restricted to presenting interval orders of width w, the value of the game is $3w-2$. Among other classes of orders that have been studied thereafter are $(\mathbf{k}+\mathbf{k})$-free orders and semi-orders. Again we refer to [2] for details.

Up-growing on-line orders have been introduced by Felsner [3]. In this variant Spoiler's power is restricted by the condition that the new element has to be a maximal element of the order presented so far. Felsner [3] showed that the value of the chain partition game on up-growing orders is $\binom{w+1}{2}$. The case of up-growing interval orders was resolved by Baier, Bosek and Micek [1]. The value of the game in this variant is $2w-1$.

This paper resolves the on-line chain partition problem for up-growing semi-orders. An order $\mathbf{P}$ is called a semi-order if it has a unit interval representation, i.e., there

Date: February 9, 2011.

Key words and phrases. on-line, chain partition, order, dimension, semi-order.

exists a mapping I of points of the order into unit length intervals on a real line so that $x < y$ in $\mathbf{P}$ iff interval $I(x)$ is entirely to the left of $I(y)$. Alternatively semi-orders are characterized as the $(\mathbf{2}+\mathbf{2})$ and $(\mathbf{3}+\mathbf{1})$-free orders (see Fig. 2).

Considering on-line chain partitions of semi-orders note that the general (not up-growing) case is easy to analyze: First, observe that the number of chains used by Algorithm can be bounded by $2w-1$. Let x be the new point and consider the set $\operatorname{Inc}(x)$ of points incomparable with x. Clearly, the only chains forbidden for x are those used in $\operatorname{Inc}(x)$. Now $\operatorname{width}(\operatorname{Inc}(x)) \leqslant w-1$ since the width of the whole order does not exceed w. Moreover, $\operatorname{height}(\operatorname{Inc}(x)) \leqslant 2$ as the presented order is $(\mathbf{3}+\mathbf{1})$-free. Therefore, $|\operatorname{Inc}(x)| \leqslant 2(w-1) = 2w-2$, proving that x can be assigned to at least one of $2w-1$ legal chains.

It turns out that there is no better strategy for Algorithm. In other words, Spoiler may force Algorithm to use $2w-1$ chains on semi-orders of width w. A strategy for Spoiler looks as follows:

(1) Present two antichains A and B, both consisting of w points in such a way that $A < B$, i.e., all points from A are below all points from B. If Algorithm uses $2w-1$ or more chains, the construction is finished. Otherwise, suppose that k chains $(2 \leqslant k \leqslant w)$ contain elements from A and B, namely let $a_i \in A_i$, $b_i \in B_i$ for $1 \leqslant i \leqslant k$ lie in the same chain.
(2) Present $k-1$ points $x_1, \ldots, x_{k-1}$ in such a way that $\{a_1, \ldots, a_i\} \leqslant x_i \leqslant \{b_{i+1}, \ldots, b_k\}$ and x_i is incomparable to all the rest (the interval representation of the whole order looks as in Fig. 1). It is easy to verify that in such setting Algorithm is forced to use $2w-1$ chains.

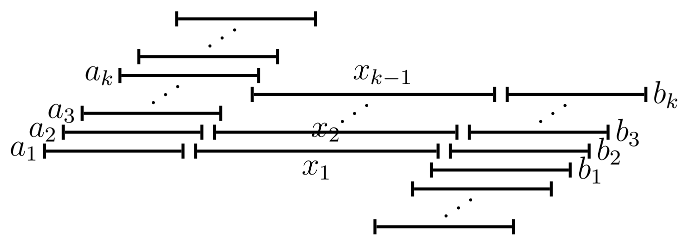

FIGURE 1. Strategy for Spoiler forcing Algorithm to use $2w-1$ chains.

The contribution of this paper is the following theorem.

Theorem 1.1. *The value of the on-line chain partition game for up-growing semi-orders of width w is $\lfloor \frac{1+\sqrt{5}}{2} \cdot w \rfloor$.*

2. Up-growing Semi-orders

2.1. **Outline.** In this section we prove that the value of the on-line chain partition game for up-growing semi-orders equals $\lfloor \varphi \cdot w \rfloor$, where $\varphi = \frac{1+\sqrt{5}}{2}$ is the golden number. First, in Sect. 2.2 we collect some facts about semi-orders. Section 2.3 describes a strategy for Spoiler which forces Algorithm to use at least $\lfloor \varphi \cdot w \rfloor$ chains on a semi-order of width w. This sets the lower bound for the value of the game. In Sect. 2.4 we propose a strategy for Algorithm using at most $\lfloor \varphi \cdot w \rfloor$ chains on semi-orders of width at most w.

The presence of the golden number φ in the result of a chain partition game may seem surprising. In fact, it is the Fibonacci sequence ($F_0 = 0$, $F_1 = 1$ and $F_{i+2} = F_i + F_{i+1}$) which appears in the counting argument of the upper bound and serves as a discrete counterpart of φ.

2.2. **Basic Facts.** For $x, y \in \mathbf{P}$ by $x \parallel_{\mathbf{P}} y$ we mean that x and y are incomparable in $\mathbf{P}$. Let $x{\downarrow}_{\mathbf{P}} = \{y \in P : y < x\}$, called a *down set* of x in $\mathbf{P}$, denote the set of predecessors of x in $\mathbf{P}$. Dually, let $x{\uparrow}_{\mathbf{P}} = \{y \in P : y > x\}$, called an *up set* of x, denote the set of successors of x in $\mathbf{P}$. If the order $\mathbf{P}$ is unambiguous from the context we also write $x{\uparrow}$ instead of $x{\uparrow}_{\mathbf{P}}$ and $x{\downarrow}$ instead of $x{\downarrow}_{\mathbf{P}}$. By $X{\downarrow}$ we mean $\bigcup_{x \in X} x{\downarrow}$.

The maximum and the minimum elements of a chain γ are denoted respectively by $\mathrm{top}(\gamma)$ and $\mathrm{bottom}(\gamma)$.

An order $\mathbf{P}$ is called an *interval order* if it has an interval representation, i.e., there exists a mapping I of points of the order into intervals on a real line so that $x < y$ in $\mathbf{P}$ iff $\max I(x) < \min I(y)$. Interval orders have several nice characterizations, see e.g. [7]. In our context the following two will be used repeatedly:

(1) $\mathbf{P} = (P, \leqslant)$ is an interval order iff the set of down sets (up sets) of elements of $\mathbf{P}$ is linearly ordered with respect to inclusion, i.e., for $p, q \in P$ either $p{\downarrow} \subseteq q{\downarrow}$ or $p{\downarrow} \supseteq q{\downarrow}$. Note that this ordering of down sets corresponds to the order of left endpoints in an interval representation.
(2) $\mathbf{P} = (P, \leqslant)$ is an interval order iff $\mathbf{P}$ is a $(\mathbf{2}+\mathbf{2})$-free order, i.e., $\mathbf{P}$ does not contain elements $a, b, c, d \in P$ such that: $a < b$, $c < d$, $a \parallel d$ and $c \parallel b$ (see Fig. 2).

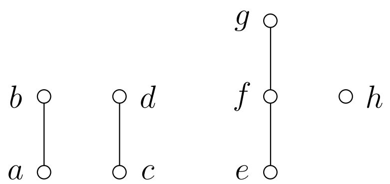

FIGURE 2. $(\mathbf{2}+\mathbf{2})$ and $(\mathbf{3}+\mathbf{1})$ orders.

An interval order $\mathbf{P}$ is called a *semi-order* if it has a unit interval representation, i.e., all intervals are of the same length. Semi-orders are also characterized in terms of forbidden structures: an interval order $\mathbf{P}$ is a semi-order iff $\mathbf{P}$ is a $(\mathbf{3}+\mathbf{1})$-free order, i.e., $\mathbf{P}$ does not contain elements $e, f, g, h \in P$ such that $e < f < g$ and $h \parallel e, f, g$ (see Fig. 2).

2.3. **The Lower Bound.** Fix w and consider the system (I_k) of k linear inequalities

$$x_0 + x_1 + \ldots + x_{j-1} + 2x_j - x_{j+1} \leqslant w, \; j = 0, \ldots, k. \qquad (I_k)$$

From the following two propositions it immediately follows that there exists a strategy for Spoiler which forces Algorithm to use $\lfloor \varphi \cdot w \rfloor$ chains on an up-growing semi-order of width w. This is the lower bound needed for Theorem 1.1.

Proposition 2.1. *If $(x_0, x_1, \ldots, x_k, x_{k+1})$ is an integral solution of (I_k) with $x_0 \geqslant x_1 \geqslant \ldots \geqslant x_k \geqslant x_{k+1} = 0$ then there is a strategy for Spoiler to present an up-growing semi-order of width w and force Algorithm to use at least $w + x_0$ chains.*

Proposition 2.2. *For each w there is an integer k and an integral solution of (I_k) with $x_0 = \lfloor(\varphi - 1) \cdot w\rfloor \geqslant x_1 \geqslant \ldots \geqslant x_k > x_{k+1} = 0$.*

Proof of Proposition 2.1. Fix w, $k > 0$ and an integer solution $(x_0, \ldots, x_k)$ of (I_k) with $x_0 \geqslant x_1 \geqslant \ldots \geqslant x_k \geqslant x_{k+1} = 0$. The strategy for Spoiler induced by $(x_0, \ldots, x_k)$ presents an up-growing semi-order $\mathbf{P} = (P, \leqslant)$ of width w. The height of $\mathbf{P}$ is at most 3. The points of $\mathbf{P}$ are presented in bundles so that the actual presentation sequence has the following structure

$$P = (A, C_0, B_1, C_1, B_2, C_2, \ldots, C_k, B_{k+1}).$$

The set A is exactly the set of minimal elements of $\mathbf{P}$. Points of height 2 lie in $C_0 \cup \bigcup_{i=1}^{k+1} B_i$ and all points from $\bigcup_{i=1}^{k} C_i$ are of height 3.

Throughout the construction Spoiler maintains auxiliary sets $D_1, \ldots, D_k$. Initially $D_i = \emptyset$ for every i. During the construction the following invariant will be kept:

$$D_i \subseteq B_i \text{ and } D_i \text{ does not contain top of any chain used by Algorithm.} \tag{1}$$

Now, we describe the phases of the construction. The proof that the construction has all desired properties will follow thereafter.

Spoiler starts the construction by presenting an antichain A of size w. Algorithm has to use w different chains.

Phase j $(0 \leqslant j \leqslant k)$. In the j-th phase Spoiler builds $x_j - x_{j+1}$ *forcing paths.* The points constituting these paths will go into the set C_j.

The first point q_0 of a forcing path dominates $A \cup \bigcup_{i \leqslant j} B_i$. Now, suppose that the first $i+1$ points of the path have been presented and let q_i be the last of these points. If Algorithm assigned q_i to a new chain or to a chain whose top is in A, then q_i is the last point of the forcing path. Otherwise, q_i was assigned to some chain with a top $b \in B_s$. In this case Spoiler updates $D_s := D_s \cup \{b\}$ and then introduces $q_{i+1} > A \cup B_1 \cup \ldots \cup B_{s-1} \cup D_s$.

Note that the invariant (1) is kept, i.e., D_i does not contain any chain top. Algorithm has to assign q_{i+1} to a new chain or to a chain with top in $A \cup B_1 \cup \ldots \cup B_{s-1}$. This means that if q_{i+1} is assigned to a chain with a top from $B_{s'}$ then $s' < s$. Thus, consecutive points $q_0, q_1, \ldots$ of a forcing path (excluding the last one) are assigned to chains with tops from the B_i's with decreasing indices. This proves that the path is finite.

The intuition is that with each forcing path Spoiler forces Algorithm to produce a *skip chain*, i.e., a chain of height 2 with its bottom in A and its top in C_j (avoiding all the B_i's), or to use a brand new chain.

Assume that Spoiler constructed all the forcing paths and consider the set $A_j \subseteq A$ of bottom points of skip chains with tops in C_j. Clearly, $|A_j| \leqslant x_j - x_{j+1}$. Now, Spoiler introduces a set B_{j+1} consisting of $x_j - x_{j+1}$ points such that $A_j \cup B_j{\downarrow} \subseteq B_{j+1}{\downarrow}$ and $|B_{j+1}{\downarrow}| = x_0 - x_{j+1}$ (put $B_0 = \emptyset$). This means that if $|A_j| < x_j - x_{j+1}$ or $A_j \cap B_j{\downarrow} \neq \emptyset$ then Spoiler completes $B_{j+1}{\downarrow}$ with arbitrarily chosen points from $A - (A_j \cup B_j{\downarrow})$. This is possible as $|B_j{\downarrow}| + |A_j| \leqslant (x_0 - x_j) + (x_j - x_{j+1}) = x_0 - x_{j+1} \leqslant w = |A|$, by (I_k).

To prove that the construction actually works and thus concludes the proof of Proposition 2.1 we have to verify the following three facts.

Fact 2.3. $\mathbf{P}$ *is a semi-order.*

Fact 2.4. *The width of* $\mathbf{P}$ *is* w.

Fact 2.5. *Algorithm has to use at least* $w + x_0$ *chains to cover* $\mathbf{P}$.

Proof of Fact 2.3. We proceed in two steps, first we show that $\mathbf{P}$ is an interval order and then that it is $(\mathbf{3}+\mathbf{1})$-free.

In order to prove that $\mathbf{P}$ is an interval order we show that the down sets of points from $\mathbf{P}$ are linearly ordered with respect to inclusion. Indeed,

(i) $A{\downarrow} = \emptyset$,
(ii) $B_1{\downarrow} \subseteq B_2{\downarrow} \subseteq \ldots \subseteq B_k{\downarrow} \subseteq A$,
(iii) $C_0{\downarrow} = A$,
(iv) if $c \in C_j$ is the starting point of a forcing path, then $c{\downarrow} = A \cup (B_1 \cup \ldots \cup B_j)$,
(v) if $c \in C_j$ is not a starting point, then there is an s with $c{\downarrow} = A \cup (B_1 \cup \ldots \cup B_{s-1} \cup D_s^t)$.

The set D_s^t from the last line refers to the respective set D_s at the moment when c is introduced. Recalling that D_s can only grow over time and $D_s \subseteq B_s$ we can conclude that the down sets of elements of c are linearly ordered with respect to inclusion. Hence $\mathbf{P}$ is an interval order.

To see that $\mathbf{P}$ is a semi-order suppose that $\mathbf{P}$ contains a $(\mathbf{3}+\mathbf{1})$-configuration $d \parallel a$, b, c with $a < b < c$. Since $\mathbf{P}$ has height at most 3 and $A < \bigcup_{i=0}^{k} C_i$, the only option is that $a \in A$, $b, d \in \bigcup_{i=1}^{k+1} B_i$ and $c \in \bigcup_{i=1}^{k} C_i$ (C_0 is excluded as it is incomparable to the B_i's). Let i, j be such that $b \in B_i$ and $d \in B_j$. Then it is easy to see that $a < d$ if $i \leqslant j$ (as in this case $a \in b{\downarrow} \subseteq d{\downarrow}$) and $d < c$ otherwise (as c being an element of a forcing path with $c > b \in B_i$ implies $c > B_1 \cup \ldots \cup B_{i-1} \supseteq B_j$). This contradiction to $d \parallel a, c$ shows that $\mathbf{P}$ is $(\mathbf{3}+\mathbf{1})$-free so it is a semi-order. $\square$

Proof of Fact 2.4. To prove that width$(\mathbf{P}) = w$ consider any antichain X in $\mathbf{P}$. We will show that $|X| \leqslant w$. Let $m \in X$ be the point with a maximal down set among points in X. We distinguish between three cases:

If $m \in A$, then $X \subseteq A$ and $|X| \leqslant |A| = w$. If $m \in B_i$, then $X \subseteq B_1 \cup \ldots \cup B_i \cup (A - B_i{\downarrow})$ and $|X| \leqslant (x_0 - x_1) + \ldots + (x_{i-1} - x_i) + (w - (x_0 - x_i)) = w$.

If $m \in C_0$, then $X \subseteq \bigcup_i B_i$ and $|X| \leqslant |C_0| + |B_1| + \ldots + |B_{k+1}| = (x_0 - x_1) + (x_0 - x_1) + \ldots + (x_k - x_{k+1}) = x_0 + x_0 - x_1 \leqslant w$ by (I_k).

The last and most interesting case is when $m \in C_j$ for $j > 0$. We may write $m{\downarrow} = A \cup (B_1 \cup \ldots \cup B_{j-1} \cup D_j^t)$ where again D_j^t is the set D_j at the moment when m was inserted. When m is the starting point of a forcing path we have $D_j^t = \emptyset$. Clearly, $X \subseteq Y \cup (\bigcup_i B_i - m{\downarrow})$ where $Y = \{c \in \bigcup_i C_i : c{\downarrow} \subseteq m{\downarrow}\}$.

Since the starting points of forcing paths in Y were introduced in phases 0 to $j-1$, their total number is $(x_0 - x_1) + (x_1 - x_2) + \ldots + (x_{j-1} - x_j) = x_0 - x_j$. The introduction of each $c \in Y$ being not a starting point of a forcing path is preceded by an extension of some D_i for $1 \leqslant i \leqslant j$. Therefore the number of non-starting points in Y is bounded by $|D_1| + \ldots + |D_{j-1}| + \left|D_j^t\right|$. To simplify this expression we first prove the following bound:

$$|D_i| \leqslant x_i.$$

For the proof of the inequality note that the set D_i is enlarged only if a point from a forcing path is assigned to a chain with a top from B_i. This can only happen for forcing paths presented in phases following phase i. Each forcing path can contribute at most one point to D_i. There are $(x_i - x_{i+1}) + \ldots + (x_k - x_{k+1})$ forcing paths in phases presented after phase i. Since $x_{k+1} = 0$ this is not greater than x_i, as claimed.

Collecting pieces from above we get $|Y| \leqslant (x_0 - x_j) + |D_1| + \ldots + |D_{j-1}| + |D_i^t| \leqslant (x_0 - x_j) + x_1 + \ldots + x_{j-1} + \left|D_j^t\right|$.

Recall that $\bigcup_i B_i - m{\downarrow} = (B_j - D_j^t) \cup B_{j+1} \cup \ldots B_{k+1}$. All this finally yields:

$$\begin{aligned}
|X| &\leqslant |Y| + \left|\bigcup_i B_i - m{\downarrow}\right| \\
&= [(x_0 - x_j) + x_1 + \ldots + x_{j-1} + \left|D_j^t\right|] \\
&\qquad + [(x_{j-1} - x_j - \left|D_j^t\right|) + (x_j - x_{j+1}) + \ldots + (x_k - x_{k+1})] \\
&= x_0 + x_1 + \ldots + x_{j-1} + (x_{j-1} - x_j)
\end{aligned}$$

From (I_k) we know that this last expression is not greater than w. □

Proof of Fact 2.5. We will prove that Algorithm is forced to use at least $w + x_0$ chains on $\mathbf{P}$. First, we show that

all points in $A - B_{k+1}{\downarrow}$ are tops of the chains to which they are assigned.

For the proof of this statement first consider points in $\mathbf{P}$ dominating $A - B_{k+1}{\downarrow}$. These are exactly the points in $\bigcup_i C_i$. Recall that if Algorithm produced a skip chain and assigned $c \in \bigcup C_i$ to a chain whose top was equal to $a \in A$ then c ends a forcing path. If this forcing path was played in phase j, then Spoiler later presented B_{j+1} in such a way that $B_{j+1} > a$ and therefore $a \in B_{j+1}{\downarrow} \subseteq B_{k+1}{\downarrow}$ so $a \notin A - B_{k+1}{\downarrow}$ and we are done.

Consider the set E of end points of forcing paths presented in the game. The key fact is that all points in $(A - B_{k+1}{\downarrow}) \cup \bigcup_i B_i \cup E$ are covered with distinct chains. Indeed, we have shown that chains in $A - B_{k+1}{\downarrow}$ are tops of the chains. End points of forcing paths are, by definition, in a chain that is not used in $\bigcup_i B_i$. Recall that

(i) $|A - B_{k+1}{\downarrow}| = w - x_0$,
(ii) $|B_1| + \ldots + |B_{k+1}| = (x_0 - x_1) + \ldots + (x_k - x_{k+1}) = x_0$,
(iii) $|E| = (x_0 - x_1) + (x_1 - x_2) + \ldots + (x_k - x_{k+1}) = x_0$.

Hence $|A - B_{k+1}{\downarrow}| + \sum_i |B_i| + |E| = w + x_0$ which gives the lower bound on the number of chains used by Algorithm. □

□

Proof of Proposition 2.2. We will show that for any w there is a solution of (I_k) with $x_0 = \lfloor(\varphi - 1) \cdot w\rfloor$. Consider the following sequence:

$$\begin{aligned}
x_0 &= \lfloor(\varphi - 1) \cdot w\rfloor, \\
x_{j+1} &= \lfloor(\varphi - 1) \cdot (w - x_0 - \ldots - x_j)\rfloor.
\end{aligned}$$

Note that for any $0 \leqslant a \leqslant x$ we have $\lfloor(\varphi-1)(x-a)\rfloor \leqslant (\varphi-1)x - (\varphi-1)a < (\varphi-1)x$ and thus $\lfloor(\varphi-1)(x-a)\rfloor \leqslant \lfloor(\varphi-1)x\rfloor$. It implies that the sequence of x_j's is decreasing. Moreover, it eventually gets to zero since the partial sum $x_0 + \ldots + x_j$ is getting larger. In particular there is a k such that $x_0 \geqslant \ldots \geqslant x_k > x_{k+1} = 0$. It is easy to verify that the sequence is a solution of (I_k), indeed

$$x_j \leqslant (\varphi - 1)(w - x_0 - \ldots - x_{j-1}) = \tfrac{\varphi}{\varphi+1}(w - x_0 - \ldots - x_{j-1}).$$

Multiplying this by $\varphi+1$, moving the term $\varphi\, x_j$ to the right hand side, adding $x_0+\ldots+x_j$ on both sides and $-w+w$ on the right side we get

$$x_0+\ldots+x_{j-1}+2x_j \leqslant (\varphi-1)(w-x_0-\ldots-x_j)+w.$$

The left side is an integer, therefore we may take the floor of the right side without affecting the truth. This results in an inequality from the system (I_k):

$$x_0+\ldots+x_{j-1}+2x_j \leqslant \lfloor(\varphi-1)(w-x_0-\ldots-x_j)\rfloor+w = x_{j+1}+w.$$

Hence $(x_0,\ldots,x_k)$ is indeed a solution of (I_k), concluding the proof of Proposition 2.2. □

2.4. **The Upper Bound.** Consider a semi-order $\mathbf{P}=(P,\leqslant)$ with $P=(a,b,c,d,e)$ and the chain partition $\Gamma : P\setminus\{e\}\to\mathbb{N}$ as shown in Fig. 3. Point e may be covered

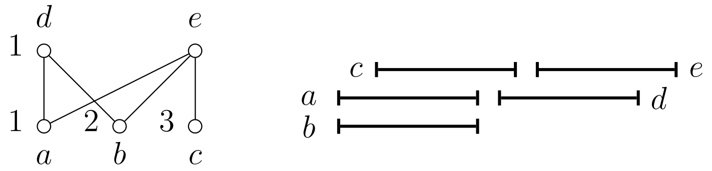

FIGURE 3. Order $\mathbf{P}$ with its unit interval representation and the chain partition Γ of points a,b,c,d.

with a new chain (say, with number 4) or with one of the chains already used. In the latter case Algorithm may choose between 2 and 3. We say that chain α is *valid* for a new point x extending an already partitioned order $\mathbf{P}$ if x dominates all points from α in $\mathbf{P}$. We claim that among the valid chains 2 and 3 defining $\Gamma(e)=3$ is the better choice. Indeed, any future point p presented by Spoiler and dominating c will also dominate b (otherwise, $\mathbf{P}$ would have a $(\mathbf{2}+\mathbf{2})$ configuration which is forbidden in interval orders). On the other hand, Spoiler may play q greater than b but remaining incomparable to c (see Fig. 4). Hence, using the chain of c for e leaves more options for

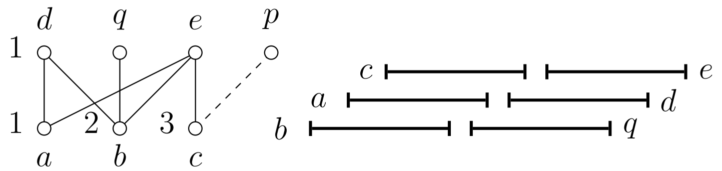

FIGURE 4. Point q may be presented by Spoiler in the future, point p can not.

the future. Whenever the chains of two points x and y are valid and $x{\uparrow} \subsetneq y{\uparrow}$ then it seems safer to use the chain of x. Our Algorithm ALG will go along this intuition.

Suppose that Spoiler introduces a semi-order $\mathbf{P}=(P,\leqslant)$ with a presentation order $P=(p_1,\ldots,p_n)$. We refer to the chains used by ALG as ALG-chains or just chains. With $\mathbf{P}_x$ we denote the order consisting of all points presented prior to x. We say that a chain α is *valid* for point x if $\mathrm{top}(\alpha)$ in $\mathbf{P}_x$ is below x in $\mathbf{P}$. The options of ALG are to put x into some valid chain or into a new chain. We are ready to describe Algorithm's strategy. Let x be a new point presented by Spoiler.

Algorithm ALG**:** If there is a valid chain for x, then put x into a valid chain α such that $\mathrm{top}(\alpha)\!\uparrow \subseteq \mathrm{top}(\beta)\!\uparrow$ in $\mathbf{P}_x$ for all valid chains β. Otherwise, if there is no valid chain, use a new chain for x.

ALG is a greedy algorithm, i.e., it uses a new chain only when it is left with no other option. Note also that ALG has some freedom in choosing a chain for a new point x as there may be many tops of valid chains for x with the same minimum up set in $\mathbf{P}_x$.

The bound on the performance of ALG on width w is stated in the following proposition. This is the upper bound part for Theorem 1.1.

Proposition 2.6. ALG *uses at most $\varphi \cdot w$ chains on any up-growing semi-order of width at most w.*

Suppose Spoiler presents a semi-order $\mathbf{P} = (P, \leqslant)$ with the presentation order $P = (p_1, \ldots, p_n)$ and $\mathrm{width}(\mathbf{P}) \leqslant w$. We may assume that ALG uses a new chain for the last point p_n. If this were not the case Spoiler could stop the game earlier and the number of chains used by ALG would remain the same.

We partition $\mathbf{P}$ into layers. These layers will in some way reflect the preferences of ALG during the game. The point $p \in \mathbf{P}$ is *significant* if p dominates at least one maximal point of $\mathbf{P}_p$. By the linearity of down sets of an interval order $\mathbf{P}$ this is equivalent to the fact that p has the largest down set at the moment of its presentation. Let $e_1, \ldots, e_{m-1}$ be the significant points of $\mathbf{P}$, sorted with respect to the presentation order (If $\mathbf{P}$ has no significant points then $\mathbf{P}$ is an antichain and the thesis is trivial). These points define the partition of $\mathbf{P}$ into layers as follows (put $e_0\!\downarrow = \emptyset$):

$$\begin{aligned} D_i &= e_i\!\downarrow - e_{i-1}\!\downarrow, \quad \text{for } 1 \leqslant i < m, \\ D_m &= P - e_{m-1}\!\downarrow. \end{aligned}$$

Thus, D_i (for $1 \leqslant i < m$) is exactly the set of maximal points of $\mathbf{P}_{e_i}$ covered by e_i. Here is a list of helpful and easy properties of the D_i's.

Fact 2.7.

(i) $d_1\!\uparrow \supsetneq d_2\!\uparrow \supsetneq \ldots \supsetneq d_m\!\uparrow$, *for all* $d_i \in D_i$.
(ii) D_m *is exactly the set of maximal points of* $\mathbf{P}$. *In particular,* $p_n \in D_m$.
(iii) D_i *is an antichain, for every* i.
(iv) *If* $d_i \in D_i$, $p \in P$ *and* $d_i < p$ *then* $D_1 \cup \ldots \cup D_{i-1} \subseteq p\!\downarrow$.
(v) *If* $d_i \in D_i$, $p \in P$ *and* $d_i \not< p$ *then* $p\!\downarrow \subseteq D_1 \cup \ldots \cup D_i$.
(vi) *If* $d_i \in D_i$, $p \in P$ *and* p *is presented prior to* d_i *then* $p\!\downarrow \subseteq D_1 \cup \ldots \cup D_{i-1}$.

Proof. (i) From $e_i \in d_i\!\uparrow$ but $e_i \notin d_j\!\uparrow$ for all $j > i$ and the linear order on the up-sets of elements we obtain $d_i\!\uparrow \supsetneq d_j\!\uparrow$.

(ii) The last significant point e_{m-1} has the largest down set in $\mathbf{P}$. This means that all points outside $e_{m-1}\!\downarrow$, namely $D_m = P - e_{m-1}\!\downarrow$, have empty up sets.

(iii) This follows from (ii) and the fact that D_i is the set of maximal points of $\mathbf{P}_{e_i}$.

(iv) Suppose $i > 1$ since for $i = 1$ the claim is obvious. Clearly, $d_i \not< e_{i-1}$. But if $d_i > e_{i-1}$ then $e_{i-1}\!\downarrow \subsetneq d_i\!\downarrow$. Therefore d_i is the next significant point after e_{i-1} or e_i was presented before d_i. Both cases are impossible as $D_i \subseteq e_i\!\downarrow$, thus $d_i \parallel e_{i-1}$. Now, if $d_i < p$ then by the linearity of down sets we obtain $D_1 \cup \ldots \cup D_{i-1} = e_{i-1}\!\downarrow \subsetneq p\!\downarrow$.

(v) Suppose that $p > d$ for some $d \in D_j$, $j > i$. Then (iv) guarantees that $D_i \subseteq p\!\downarrow$. This implies $d_i < p$, contradicting the assumptions.

(vi) If $i = m$ then the thesis is trivial as D_m is the set of maximal points of $\mathbf{P}$. Thus, suppose $i < m$. Recall that e_i is the first presented point which dominates every point from D_i. Now as p is presented prior to d_i and d_i precedes e_j for $j \geqslant i$, we get that p cannot dominate any point from D_j. □

With the next fact we prove that ALG always chooses a chain whose top is in the highest of the layers which contain tops of valid chains.

Fact 2.8. *Let $d_i \in D_i$, $d_j \in D_j$, $i < j$ and $d_i, d_j < x$. If d_i is an* ALG*-top in $\mathbf{P}_x$ and* ALG *uses its chain on x then d_j is not an* ALG*-top in $\mathbf{P}_x$.*

Proof. At the moment when Spoiler presents x, both d_i and d_j are already introduced ($\mathbf{P}$ is up-growing). Since e_{j-1} was presented before d_j, $d_i \in e_{j-1}{\downarrow}$ and $d_j \not< e_{j-1}$ we conclude that $d_i{\uparrow} \supsetneq d_j{\uparrow}$ in $\mathbf{P}_x$. Hence if both d_i and d_j are valid tops for x then ALG has a preference for using the chain of d_j for x. □

Before we proceed with the proof we introduce the concept of a predecessor and a successor of a point with respect to some fixed chain partition. Let $\mathcal{C}$ be the chain partition of $\mathbf{P} = (P, \leqslant)$. For $p \in P$ with $p \in C$ for a chain $C \in \mathcal{C}$ we define:

(i) The *predecessor* of p in $\mathcal{C}$ is the point preceding p in C (if p is the least point in C then the predecessor of p does not exist).
(ii) The *successor* of p in $\mathcal{C}$ is the point succeeding p in C (if p is the largest point in C then the successor of p does not exist).

We fix an optimal chain partition $\mathcal{O}$ of $\mathbf{P}$. Since $\mathrm{width}(\mathbf{P}) \leqslant w$ this partition consists of at most w chains. With respect to this partition we denote the predecessor and the successor of $p \in P$ by $o^-(p)$ and $o^+(p)$, respectively. Analogously, we refer to the predecessor and the successor of p with respect to the the chain partition constructed by ALG as $\mathrm{alg}^-(p)$ and $\mathrm{alg}^+(p)$.

We arrive at the key concept of the proof: the alternating paths. Each such path starts at the bottom of an ALG-chain. We propose to understand it as a chain of events originating from the starting bottom. By counting the number of such paths we will get a bound on the number of chains used by ALG.

For each ALG-chain α define an *alternating path* $q = (q_0, \ldots)$ as follows:

(i) q_0 is the bottom point of α,
(ii) $q_{2i+1} = o^-(q_{2i})$, if $o^-(q_{2i})$ does exist,
(iii) $q_{2i+2} = \mathrm{alg}^+(q_{2i+1})$, if $\mathrm{alg}^+(q_{2i+1})$ does exist.

We claim that for each path q all the q_{2i}'s are pairwise distinct and so are all the q_{2i+1}'s. Indeed, note that $q_0 \neq q_{2i}$ for $i > 0$ as $q_{2i} = \mathrm{alg}^+(q_{2i-1})$ is not a bottom of an ALG-chain, while q_0 is defined as a bottom. Now suppose that the claim does not hold and consider the least i such that $q_i = q_j$ for some $j > i$ where i and j have the same parity. If i and j are even we get $q_{i-1} = \mathrm{alg}^-(q_i) = \mathrm{alg}^-(q_j) = q_{j-1}$ and if they are odd we get $q_{i-1} = o^+(q_i) = o^+(q_j) = q_{j-1}$. In both cases this contradicts the choice of i. This fact implies that the alternating paths are finite. Note that an alternating path $q = (q_0, \ldots, q_l)$ is uniquely determined by any $q_i \in q$ together with the information whether q_i is an odd or a even element. Altogether we have proven the following fact:

Fact 2.9. *For an alternating path $q = (q_0, \ldots, q_l)$ all the q_i's with the same parity of indices are pairwise distinct, i.e. $q_{2i} \neq q_{2j}$ and $q_{2i+1} \neq q_{2j+1}$ for $i \neq j$. Moreover, each*

$p \in P$ occurs in at most two alternating paths: once with an odd index and once with an even index.

For an alternating path $q = (q_0, \dots, q_l)$ we call the q_{2i}'s the *up-points* of q and the q_{2i+1}'s the *down-points* of q. An alternating path $q = (q_0, \dots, q_l)$ is an *up-path* if its last point is an up-point, otherwise, q is a *down-path*. Note that an up-path connects the bottom of an ALG-chain with the bottom of an $\mathcal{O}$-chain, hence, there are at most w up-paths (see Fact 2.12). The goal is to bound the number of down-paths.

From our perspective the important layers in the partition of $P = D_1 \cup \dots \cup D_m$ will be those containing at least one end point of a down-path. Define

$$I = \{\, i : \text{ there is a down-path ending in } D_i \,\} = \{\, i_0 < i_1 < \dots < i_s \,\}.$$

Note that $m \notin I$ being an end point of a down-path has a non-empty up set, while up sets of all points in D_m are empty (see Fact 2.7.(ii)). This means

$$i_s < m. \tag{2}$$

From the definition of I we immediately obtain:

Fact 2.10. *There is an* ALG*-top in every D_{i_j}, for $0 \leqslant j \leqslant s$.*

The next fact basically states that $D_{i_0} \cup \dots \cup D_m$ induce an order of height at most 3.

Fact 2.11. *$D_{i_0+1} \cup \dots \cup D_{m-1}$ is an antichain.*

Proof. In order to get contradiction suppose that there are d, $d' \in D_{i_0+1} \cup \dots \cup D_{m-1}$ and that $d < d'$. As d' dominates a point from a layer higher than D_{i_0}, we get from Fact 2.7(iv) that $D_{i_0} \subseteq d'{\downarrow}$. On the other hand, since $d' \notin D_m$ there must be some point $d'' > d'$ (see Fact 2.7(ii)). Fix some $t \in D_{i_0}$ that remains an ALG-top throughout the game (such t exists by Fact 2.10). Recall that p_n, the last point presented by Spoiler, is incomparable with t as otherwise p_n would not be assigned to a new ALG-chain. Together this shows that p_n and $t < d' < d''$ form a $(\mathbf{3}+\mathbf{1})$-configuration. This is impossible since $\mathbf{P}$ is a semi-order. □

We now introduce variables that count the number of paths with respect to their end points. Define

x_U the total number of up-paths,
x_j the number of down-paths ending in $\bigcup_{i \geqslant i_j} D_i$, for $j = 0, \dots, s$,
$x_{s+1} = 0$.

In terms of these variables the number of chains used by ALG can be expressed as

$$x_U + x_0.$$

Fact 2.12. *$x_U \leqslant w$.*

Proof. Consider the set U of end points of up-paths. By Fact 2.9 these end points are pairwise distinct and $|U| = x_U$. As each point in U is a bottom point of its chain in $\mathcal{O}$ these points belong to different chains in $\mathcal{O}$. Therefore, $x_U = |U| \leqslant |\mathcal{O}| \leqslant w$. □

Throughout the rest of the paper our efforts are targeted on bounding the number x_0 of down-paths. In the following two lemmas we will present a system of inequalities involving x_0 and the other x_i's. From these inequalities we will derive the desired constraints on the value of x_0.

Lemma 2.13. *$x_0 + x_0 - x_1 \leqslant w$.*

Lemma 2.14. *$x_0 + x_1 + \ldots + x_j + (x_j - x_{j+1}) \leqslant w$, for $j = 1, \ldots, s$.*

We need some preparations for the proofs of the lemmas. In the counting arguments on which the proofs are based we will be often showing that certain sets are disjoint. Below we introduce a criterion that will help us do the bookkeeping.

Point $p \in P$ is a *good* point if $o^-(p)$ exists and $o^-(p)$ is an ALG-top at the moment when p is presented, i.e., $o^-(p)$ is a top of its ALG-chain in $\mathbf{P}_p$. Simple enough, point p is considered *bad* if it is not good.

Fact 2.15 states that the penultimate point of a down-path is always good. Fact 2.16 says that if a down-path ends in D_{i_j} then it contains $j+1$ bad up-points. Fact 2.17 is a technical statement used in the proof of Lemma 2.14.

Fact 2.15. *The penultimate point of a down-path is a good point and lies in D_m.*

Proof. Let $q = (q_0, \ldots, q_{l-1}, q_l)$ be a down-path. We have $q_l = o^-(q_{l-1})$ and since q ends with $q_l = o^-(q_{l-1})$ this point must be an ALG-top in $\mathbf{P}$. This shows that q_{l-1} is good.

To prove that $q_{l-1} \in D_m$ we are going to prove an equivalent condition that the up set of q_{l-1} is empty (see Fact 2.7(ii)). Recall that the last point presented in $\mathbf{P}$, namely p_n, receives a new ALG-chain. This means that there is no valid chain for p_n and in particular, $p_n \not> q_l$. From $p_n\!\uparrow = \emptyset$ we get $p_n \parallel q_l$. Now, if there was a point $x > q_{l-1}$ then p_n, q_l, q_{l-1}, x would form a $(\mathbf{3}+\mathbf{1})$-configuration. This is impossible since $\mathbf{P}$ is a semi-order. □

Fact 2.16. *A down-path q ending in D_{i_j} contains $j+1$ bad up-points $\{y_0, \ldots, y_j\}$ such that $o^-(y_k) \in D_{i_k}$, for $0 \leqslant k \leqslant j$.*

Proof. Fix k and define y_k to be the first up-point in q such that $o^-(y_k) \in D_l$ for some $l \geqslant i_k$. Such point does exist, as the penultimate point in q is a candidate for y_k.

Claim. y_k is a bad point.

Suppose y_k is the first point of q. In this case ALG uses a new chain on y_k. Since ALG is greedy there is no valid chain for y_k at the moment it is presented. In particular, $o^-(y_k)$ is not an ALG-top in $\mathbf{P}_{y_k}$ and therefore y_k is a bad point.

If y_k is not the first point of $q = (\ldots, p, o^-(p), y_k, o^-(y_k), \ldots)$, then we know that p did not qualify for y_k and therefore $\mathrm{alg}^-(y_k) = o^-(p) \in D_1 \cup \ldots \cup D_{i_k-1}$. From Fact 2.8 it follows that $o^-(y_k)$ is not an ALG-top in $\mathbf{P}_{i_k}$ and again y_k is bad. This proves the claim.

It remains to show that $o^-(y_k) \in D_{i_k}$. Fix an ALG-top $t \in D_{i_k}$ (Fact 2.10). Note that $t \not< y_k$ as otherwise ALG would have given preference to the chain of t instead of the one it used for y_k. From Fact 2.7(v) we get $y_k\!\downarrow \subseteq D_1 \cup \ldots \cup D_{i_k}$. From this and the definition of y_k we conclude $o^-(y_k) \in D_{i_k}$. □

Fact 2.17. *For every j $(0 \leqslant j \leqslant s)$ and down-path q there is an up-point $u \in q$ such that one of the following two conditions is true:*

(i) *$u \in D_{i_j+1} \cup \ldots \cup D_m$ and $o^-(u) \in D_{i_0}$, or*
(ii) *u is good and $o^-(u) \in D_{i_0+1} \cup \ldots \cup D_{i_j-1}$.*

Proof. Fix j and a down-path q. Let a be the last up-point in q such that $o^-(a) \in D_k$ for some $k \leqslant i_0$. There is such a point as by Fact 2.16 each down-path has an up-point y with $o^-(y) \in D_{i_0}$.

First we are going to prove that $o^-(a) \in D_{i_0}$. It is trivial if $o^-(a)$ is the last point of q as the first layer with an end of a down-path is D_{i_0}. If $o^-(a)$ is not the last point then let b be the next point of q, i.e. $q = (\ldots, a, o^-(a), b, o^-(b), \ldots)$. From the way point a is chosen we have $o^-(b) \in D_l$, for some $l > i_0$. By Fact 2.7(iv) $b > o^-(b) \in D_l$ implies $D_{i_0} \subseteq b\Downarrow$. Now, recall that there is an ALG-top $t \in D_{i_0}$ (Fact 2.10). In particular, at the moment when b was introduced the chain of $t \in D_{i_0}$ was valid for b. Since ALG has chosen $\mathrm{alg}^-(b) = o^-(a) \in D_k$ we get $k \geqslant i_0$ (Fact 2.8). Since a was chosen such that $o^-(a) \in D_k$ for some $k \leqslant i_0$ we get $k = i_0$.

If $a \in D_{i_j+1} \cup \ldots \cup D_m$ then a fulfills the condition for the u of (i) and we are done. From now on we deal with the case $a \in D_k$ for some $k \leqslant i_j$. As $a > o^-(a) \in D_{i_0}$ point a must be somewhere in $D_{i_0+1} \cup \ldots \cup D_m$. We therefore know that

$$a \in D_{i_0+1} \cup \ldots \cup D_{i_j}. \tag{3}$$

Since $i_j < m$ (see (2)) we have $a \notin D_m$, hence, there is a point $a' \in P$ with $a' > a$. Note that $o^-(a)$ can't be the last point of q. Otherwise, a would be the penultimate point of q which is in D_m (Fact 2.15). Let b be the successor of $o^-(a)$ in q, i.e. $q = (\ldots, a, o^-(a), b, o^-(b), \ldots)$. We claim that Spoiler presents b prior to a in $\mathbf{P}$, i.e. $b \in \mathbf{P}_a$. From the definition of a it follows that $b > o^-(b) \in D_l$, for some $l > i_0$ and hence $D_{i_0} \subseteq b\Downarrow$. (Fact 2.7(v)). At the moment when b is presented there are at least two valid chains for b, the one actually used by ALG with its top in $\mathrm{alg}^-(b) = o^-(a)$ and some chain with top $t \in D_{i_0}$ (Fact 2.10). Recall that the last point p_n presented in $\mathbf{P}$ is put into a new ALG-chain and therefore $p_n \parallel t$. Now, if $a > t$ then p_n together with $t < a < a'$ would form a $(\mathbf{3}+\mathbf{1})$-configuration. Therefore $a \parallel t$ while obviously $a > o^-(a)$. Suppose that a is presented by Spoiler prior to b. This implies that at the moment when b is introduced (i.e. in $\mathbf{P}_b$) we have $t\Uparrow \subsetneq o^-(a)\Uparrow$ and therefore ALG would prefer the chain of t over the chain of $\mathrm{alg}^-(b) = o^-(a)$ to be used for b. With this contradiction we have proved the claim that the order of presentation is $P = (\ldots, b, \ldots, a, \ldots)$.

Let c be the last up-point in q presented by Spoiler prior to a (i.e. in $\mathbf{P}_a$). There is such a point as b is an up-point of q and it is presented prior to a. Since b comes after a on q this also holds for c, i.e.,

$$q = (\ldots, a, o^-(a), \ldots, c, o^-(c), \ldots). \tag{4}$$

The last step in the proof is to show that c fulfills the condition for the u of (ii).

First, we show that c is good. If not, $z = \mathrm{alg}^+(o^-(c))$ would be defined and had to be presented prior to c, so before a as well. But then z would be also an up-point of q which contradicts the choice of c.

It remains to prove that $o^-(c) \in D_{i_0+1} \cup \ldots \cup D_{i_j-1}$. Recall that a is the last up-point of q with $o^-(a) \in D_k$ for $k \leqslant i_0$. Since c comes later than a on q (see (4)) we have $o^-(c) \in D_k$ for some $k \geqslant i_0 + 1$. To bound k from above recall that $a \in D_{i_0+1} \cup \ldots \cup D_{i_j}$ and $o^-(c)$ is presented prior to a. Applying this to Fact 2.7.(vi) we get that $o^-(c) \in \bigcup_{i<i_j} D_i$. This finishes the proof of Fact 2.17. $\square$

With these preparations we are ready for the proofs of Lemmas 2.13 and 2.14.

Proof of Lemma 2.13. We are going to construct an antichain in $\mathbf{P}$ of size $x_0 + x_0 - x_1$. This implies the statement.

By Fact 2.16 a down-path q contains a bad up-point y_q with $o^-(y_q) \in D_{i_0}$. Collect these points in a set $Y = \{y_q :\ q \text{ is a down-path}\}$. Since x_0 is just the number of down-paths and Y contains a point from each down-path we get $|Y| = x_0$. Let Z be the set of penultimate points of down-paths ending in D_{i_0}. The number of such paths is $x_0 - x_1$. hence $|Z| = x_0 - x_1$. From Fact 2.15 we know that points in Z are good. Summarizing:

(i) $|Y| = x_0$, all $y \in Y$ are bad and satisfy $o^-(y) \in D_{i_0}$.
(ii) $|Z| = x_0 - x_1$, all $z \in Z$ are good and satisfy $o^-(z) \in D_{i_0}$.

This implies that Y and Z are disjoint and

$$x_0 + x_0 - x_1 = |Y| + |Z| = |Y \cup Z| = \left|o^-(Y \cup Z)\right| \leqslant |D_{i_0}| \leqslant w,$$

where the last inequality holds as D_{i_0} is an antichain in $\mathbf{P}$ (Fact 2.7(iii)). □

Proof of Lemma 2.14. The basic idea of the proof is similar to the proof Lemma 2.13 but the details are more involved. For fixed j we construct a set consisting of $x_0 + x_1 + \ldots + x_j + x_j - x_{j+1}$ points such that the points of the set belong to different chains in $\mathcal{O}$. As $\mathcal{O}$ contains at most w chains, this implies the inequality with index j.

Fix j. First we construct a set of size $x_1 + \ldots + x_j$. For a down-path q ending in D_{i_k} put $r(q) = \min(k, j)$. By Fact 2.16 each q contains a set Y_q of $r(q)$ bad up-points y with $o^-(y) \in D_{i_1} \cup \ldots \cup D_{i_j}$. Let Y be the union of all these sets Y_q. We claim that that all the Y_q's are pairwise disjoint. This is true because no point occurs in more than one alternating path as an up-point (Fact 2.9). The claim implies $|Y| = \sum_q |Y_q|$. We determine the size of Y as follows

$$\begin{aligned}
|Y| &= \sum_{k=1}^{s} \sum_{\substack{q \text{ ending} \\ \text{in } D_{i_k}}} |Y_q| = \sum_{k=1}^{s} \sum_{\substack{q \text{ ending} \\ \text{in } D_{i_k}}} r(q) = \sum_{k=1}^{s} \min(k, j) \cdot (x_k - x_{k+1}) \\
&= \sum_{k=1}^{j} k \cdot (x_k - x_{k+1}) + \sum_{k=j+1}^{s} j \cdot (x_k - x_{k+1}) = x_1 + \ldots + x_j - j \cdot x_{s+1} \\
&= x_1 + \ldots + x_j.
\end{aligned}$$

For further reference we collect the important properties of Y:

(i) $|Y| = x_1 + \ldots + x_j$.
(ii) All $y \in Y$ are bad and $o^-(y) \in D_{i_1} \cup \ldots \cup D_{i_j} \subseteq D_{i_0+1} \cup \ldots \cup D_{i_j}$.

The second set to consider is:

$$Z = \left\{ z :\ \text{there is a down-path } q = (\ldots, z, o^-(z)) \text{ and } o^-(z) \in D_{i_j} \right\}.$$

This is the set of the penultimate points of down-paths ending in D_{i_j}. The penultimate points of down-paths are up-points and hence all distinct (Fact 2.9). From Fact 2.15 we know that all points in Z are good. Summarizing:

(i) $|Z| = x_j - x_{j+1}$,
(ii) All $z \in Z$ are good and $o^-(z) \in D_{i_j}$

With a help of Fact 2.17 we construct a third set U. Each down-path q contains an up-point u_q satisfying property (i) or (ii) of Fact 2.17. The set U is the collection of all these points. Since no point is an up-point of more than one path all the u_q's are distinct. Since there are x_0 down-paths we have $|U| = x_0$. We partition the set $U = \{u_q : \ q \text{ down-path}\}$ into three parts U_1, U_2 and U_3 as follows:

(i) U_1 is the set of $u \in U$ with $u \in D_{i_j+1} \cup \ldots \cup D_{m-1}$ and $o^-(u) \in D_{i_0}$.
(ii) U_2 is the set of $u \in U$ with $u \in D_m$ and $o^-(u) \in D_{i_0}$.
(iii) U_3 is the set of $u \in U$ such that u is good and $o^-(u) \in D_{i_0+1} \cup \ldots \cup D_{i_j-1}$.

The following properties of sets Y, Z and U_1, U_2, U_3 are crucial:

(1) The sum of sizes of U_1, U_2, U_3, Y and Z is $x_0 + x_1 + \ldots + x_j + (x_j - x_{j+1})$.
(2) U_3, Y and Z are disjoint. Indeed points in Y are bad while points in Z and U_3 are good. The predecessors in $\mathcal{O}$-chains show that Z and U_3 are disjoint.
(3) $o^-(U_3 \cup Y \cup Z) \subseteq D_{i_0+1} \cup \ldots \cup D_{i_j}$ and $U_1 \subseteq D_{i_j+1} \cup \ldots \cup D_{m-1}$.
(4) Points in $U_2 \cup D_{i_0+1} \cup \ldots \cup D_{m-1}$ lie in different $\mathcal{O}$-chains. Indeed, $D_{i_0+1} \cup \ldots \cup D_{m-1}$ is an antichain (Fact 2.11) and $\mathcal{O}$-chains of points from U_2 skip all layers between D_{i_0} and D_m, i.e., they avoid $D_{i_0+1} \cup \ldots \cup D_{m-1}$.

We are now ready to complete the proof:

$$\begin{aligned} x_0 + (x_1 + \ldots + x_j) + (x_j - x_{j+1}) = \\ &\overset{(1)}{=} (|U_1| + |U_2| + |U_3|) + |Y| + |Z| \\ &\overset{(2)}{=} |U_2| + |U_3 \cup Y \cup Z| + |U_1| \\ &= |U_2| + \left|o^-(U_3 \cup Y \cup Z)\right| + |U_1| \\ &\overset{(3)}{\leqslant} |U_2| + |D_{i_0+1} \cup \ldots \cup D_{m-1}| \\ &\overset{(4)}{\leqslant} |\mathcal{O}| \\ &\leqslant w. \end{aligned}$$

□

Lemmas 2.13 and 2.14 provide us with a system of inequalities involving all the x_i's:

$$x_0 + x_1 + \ldots + x_j + x_j - x_{j+1} \leqslant w, \ j = 0, \ldots, k.$$

Note that these are the inequalities used for the lower bound. Here, for the completion of the proof of Proposition 2.6 we need a final lemma to bound the number $x_0 + w$ of chains used by Algorithm.

Lemma 2.18. $x_0 \leqslant (\varphi - 1) \cdot w$.

Proof. We weight the jth inequality with the Fibonacci number $F_{2(k-j)+1}$ and take the sum of weighted inequalities:

$$\sum_{j=0}^{k} \sum_{i=0}^{j} x_i F_{2(k-j)+1} + \sum_{j=0}^{k} (x_j - x_{j+1}) F_{2(k-j)+1} \leqslant w \sum_{j=0}^{k} F_{2(k-j)+1}. \tag{5}$$

Using the well-known Fibonacci identity $\sum_{j=0}^{k} F_{2(k-j)+1} = \sum_{j=0}^{k} F_{2j+1} = F_{2k+2}$ we can simplify the double-sum:

$$\sum_{j=0}^{k} \sum_{i=0}^{j} x_i F_{2(k-j)+1} = \sum_{i=0}^{k} x_i \sum_{j=i}^{k} F_{2(k-j)+1} = \sum_{j=0}^{k} x_j F_{2(k-j)+2}.$$

This and again using the Fibonacci identity on the right hand side allows us to rewrite inequality (5):

$$\sum_{j=0}^{k} x_j F_{2(k-j)+2} + \sum_{j=0}^{k} x_j F_{2(k-j)+1} - \sum_{j=1}^{k+1} x_j F_{2(k-j)+3} \leqslant w F_{2k+2}.$$

Using the Fibonacci recursion and the fact that $x_{k+1} = 0$ this reduces to

$$x_0 F_{2k+2} + x_0 F_{2k+1} \leqslant w F_{2k+2}.$$

This in turn can be rewritten into

$$x_0 \leqslant \frac{F_{2k+2}}{F_{2k+3}} \cdot w \leqslant (\varphi - 1) \cdot w.$$

The last inequality is due to the fact that the sequence $(\frac{F_{2k+2}}{F_{2k+3}})_{k \geqslant 0}$ is monotonically increasing with the limit $\varphi - 1$. □

The statement of Lemma 2.18 was the last piece of the puzzle. The number of chains used by ALG is bounded by

$$x_U + x_0 \leqslant w + (\varphi - 1) \cdot w.$$

This completes the proof of Proposition 2.6 and hence of Theorem 1.1.

References

[1] Patrick Baier, Bartłomiej Bosek, and Piotr Micek. On-line chain partitioning of up-growing interval orders. *Order*, 24(1):1–13, 2007.

[2] Bartłomiej Bosek, Stefan Felsner, Kamil Kloch, Tomasz Krawczyk, Grzegorz Matecki, and Piotr Micek. On-line chain partitions of orders: a survey. to appear in ORDER.

[3] Stefan Felsner. On-line chain partitions of orders. *Theoret. Comput. Sci.*, 175(2):283–292, 1997. Orders, algorithms and applications (Lyon, 1994).

[4] Henry A. Kierstead. An effective version of Dilworth's theorem. *Trans. Amer. Math. Soc.*, 268(1):63–77, 1981.

[5] Henry A. Kierstead. Recursive ordered sets. In *Combinatorics and ordered sets (Arcata, Calif., 1985)*, volume 57 of *Contemp. Math.*, pages 75–102. Amer. Math. Soc., Providence, RI, 1986.

[6] Henry A. Kierstead and William T. Trotter, Jr. An extremal problem in recursive combinatorics. In *Proceedings of the Twelfth Southeastern Conference on Combinatorics, Graph Theory and Computing, Vol. II (Baton Rouge, La., 1981)*, volume 33, pages 143–153, 1981.

[7] Rolf H. Möhring. Computationally tractable classes of ordered sets. In *Algorithms and Order*, pages 105–193. Kluwer Acad. Publ., Dordrecht, 1989.

AG Diskrete Mathematik, Institut für Mathematik, Technische Universität Berlin
Strasse des 17. Juni 136, 10623 Berlin, Germany
E-mail address: felsner@math.tu-berlin.de
URL: http://www.math.tu-berlin.de/~felsner/

Embedded Systems Lab, University of Passau, Innstrasse 43, 94032 Passau, Germany
E-mail address: kamil.kloch@uni-passau.de
URL: http://www.esl.fim.uni-passau.de/~kloch/

Algorithmics Research Group, Faculty of Mathematics and Computer Science
Jagiellonian University, Lojasiewicza 6, 30-348 Kraków, Poland
E-mail address: Grzegorz.Matecki@tcs.uj.edu.pl
URL: http://tcs.uj.edu.pl/Matecki
E-mail address: Piotr.Micek@tcs.uj.edu.pl
URL: http://tcs.uj.edu.pl/Micek