ON-LINE CHAIN PARTITIONS OF ORDERS: A SURVEY

BARTŁOMIEJ BOSEK, STEFAN FELSNER, KAMIL KLOCH, TOMASZ KRAWCZYK, GRZEGORZ MATECKI, AND PIOTR MICEK

Abstract. On-line chain partition is a two-player game between Spoiler and Algorithm. Spoiler presents, point by point, a partially ordered set. 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 val(w) such that Algorithm has a strategy using at most val(w) chains on orders of width at most w. There are many recent results about variants of the general on-line chain partition problem. With this survey we attempt to give an overview over the state of the art in this field. As particularly interesting aspects of the article we see:

– The sketch of the proof for the new sub-exponential upper bound of Bosek and Krawczyk: $\mathrm{val}(w) \leqslant w^{16 \lg(w)}$.
– The new lower bound: $\mathrm{val}(w) \geqslant (2 - o(1))\binom{w+1}{2}$.
– The inclusion of some simplified proofs of previously published results.
– The comprehensive account on variants of the problem for interval orders.
– The new lower bound for 2-dimensional up-growing orders.

1. General problem

Partitioning graphs and orders into simple components is a fundamental combinatorial task. The on-line approach to these problems is receiving much attraction not only because of its natural application flavor but also because of the beautiful mathematics evolving in the area. The classical theorem of Dilworth says that an order of width w can be partitioned into w chains. The dual of Dilworth's theorem is also true: an order of height h admits a partition into h antichains. In the on-line setting the challenge is to construct such a partition of the smallest possible size, assuming that the points constituting the order are revealed one at a time and have to be assigned to a block of the partition immediately.

An on-line algorithm can be understood as an algorithm running without the full knowledge of the input. Instead, the input is revealed piece by piece and for each new piece of the input an irrevocable action must be taken before the next piece is shown. Such a scenario surely applies in many real-world problems. In this paper the setting of on-line algorithms and their performance is discussed in terms of two-person games and strategies for the two players of such games.

The first results about on-line algorithms for problems on orders were obtained in the context of *recursive combinatorics*, this is a logics program aiming for constructive existence proofs of infinite structures. The introduction of Kierstead's survey [16] has a more detailed account. In our terminology [16] could have been entitled *On-line problems for orders*. In the late 1980's the idea of on-line algorithms and competitive analysis

Date: January 19, 2011.
Key words and phrases. on-line, chain partition, order.

became popular in computer science and quickly found its way into combinatorics. Since then most of the work in the area is using this terminology.

Our interest lies primarily in the on-line chain partition problem. We like to describe this problem as a two-player coloring game. The players are named *Spoiler* and *Algorithm*. The game is played in rounds. In each round Spoiler adds one new point to the present order and describes all comparabilities to already existing points. Algorithm responds by making an assignment of the new point to one of the chains in the chain partition that he maintains. Think of the chains as being colored with different colors and of the point to be colored in the color of the chain containing them, it then makes sense to say that a move of Algorithm consists in assigning a color to the new point. The aim of Algorithm is to use few colors while Spoiler tries to force many colors. In Fig. 1 we show the smallest example that can force any on-line algorithm to use more than w chains on an order of width w.

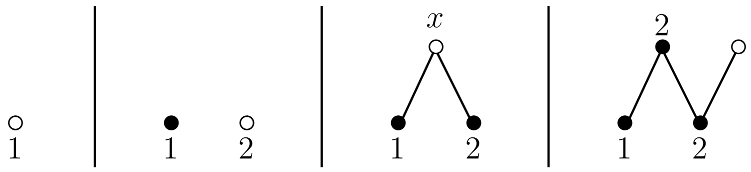

FIGURE 1. Spoiler forces 3 chains on an order of width 2. The white element is the new element of a round. 1, 2, 3 are chains. In the third round Algorithm has three choices for x: The case $x = 3$ is an immediate win for Spoiler, the other two cases are symmetric and lead to Spoiler's win in round 4.

To measure the quality of the on-line algorithm we consider upper bounds on the number of chains in the constructed partition as a function of the width w. This is natural because by Dilworth's theorem an optimal off-line algorithm would produce a chain partition with exactly w chains.

The value $\text{val}(w)$ of the game is the least integer s such that Algorithm has a strategy using at most s chains on any on-line order of width at most w. It can be verified that $\text{val}(w)$ is equivalently the largest s for which Spoiler has a strategy forcing any algorithm to use s chains on an order of width w.

Obviously $\text{val}(1) = 1$ but at first glance it is not clear if $\text{val}(2)$ is well-defined, i.e., that there exists a constant c such that Algorithm can partition any on-line order of width 2 into at most c chains. Leaving out the detailed bounds on $\text{val}(w)$ for Section 3, here we only mention that the best-known lower bound for $\text{val}(w)$ is quadratic while the best-known upper bound is super-polynomial (until 2009, it had been exponential). In our belief, this huge gap represents one of the most intriguing challenges in the whole domain of partially ordered sets.

This paper is organized as follows. In Section 2 we deal with on-line antichain partitions. Since this problem is much easier than on-line chain partitioning this section may serve as a warm-up. Section 3 is concerned with the on-line chain partition problem for general orders. Theorem 3.1 is a lower bound construction due to Szemeredi. Theorem 3.2 is a new lower bound gaining a factor of almost two. The up-growing variant of the on-line chain partition problem is introduced. We show the tight upper bound for this variant in Theorem 3.5. In Subsection 3.1 we sketch the proof of the new sub-exponential upper bound (Theorem 3.4).

In Sections 4 to 6 we discuss variants of the problem on restricted classes of orders. In Section 4 we begin with interval orders and consider variants where the order is given with or without representation and with or without the up-growing restriction. Section 5 is concerned with semi-orders. The value of the up-growing game on this class is based on the golden ratio (Theorem 5.2). For some variants where the order is given with a geometric representation the value of the game remains elusive. In Section 6 we review the state of the art regarding d-dimensional orders. This section also contains a new lower-bound proof for up-growing 2-dimensional orders presented with a realizer (Theorem 6.2).

Playing on special classes of orders, as in the up-growing variant or adding information regarding a representation, is restricting the power of Spoiler. In Sections 7 and 8 we talk about variants where the power of Algorithm is modified. In Section 7 we report on results about First-Fit, this takes all freedom from Algorithm, the color of the new point can be pre-calculated by Spoiler. Still there are situations where First-Fit has its merits. In Section 8 we discuss the adaptive version of the game. In this setting Algorithm may assign a point to several chains and withdraw it from some of these chains in later stages of the game. We conclude in Section 9 with some open problems.

2. On-line antichain partitions

Dilworth's Theorem is much harder to prove than its dual version. The difference in 'hardness' carries over to the on-line setting. As shown below, the value of the on-line antichain partition problem is precisely known while the situation for chains (Section 3) is much more intricate.

The following precise result must be provided with two complementary strategies: for Spoiler and for Algorithm. Algorithm's strategy using at most $\binom{h+1}{2}$ antichains on orders of height h appears in [16] where it is attributed to James Schmerl. Kierstead [16] also describes a strategy for Spoiler that forces any on-line algorithm to use at least $\binom{h+1}{2}$ antichains on an order of height h. The strategy is attributed to Emre Szemerédi. We describe a camouflage version for Szemerédi's lower bound below in Theorem 3.1. To translate the proof of Theorem 3.1 to the antichain problem it has to be noted that the order $\mathbf{P}$ presented by Spoiler has dimension 2 and can be presented together with an on-line realizer. Reverting one of the linear extensions of the realizer yields the conjugate order $\mathbf{P}^c$ which has the property that chains of $\mathbf{P}$ are in bijection to antichains of $\mathbf{P}^c$ and vice versa. (For details on dimension 2 and conjugate orders we refer to [25].)

Theorem 2.1 (Schmerl, Szemerédi)**.** *The value of the on-line antichain partition game for orders of height h is $\binom{h+1}{2}$.*

Proof of the upper bound. Algorithm will maintain an antichain partition using a family of antichains $A_{(a,b)}$ indexed by pairs (a, b) of numbers $1 \leqslant a, b$ and $a + b \leqslant h + 1$. Since there are exactly $\binom{h+1}{2}$ such pairs (a, b) this will prove the theorem.

When Spoiler presents a new point x Algorithm determines the size a of the longest chain in the already presented order that has x as its maximum element and the size b of the longest chain that has x as its minimum element. As the size of any chain in the already presented order is at most h we get $a + b \leqslant h + 1$.

Now, Algorithm inserts x into $A_{(a,b)}$. It has to be shown that $A_{(a,b)}$ remains an antichain. Indeed, suppose that x is comparable with some y that was previously put into $A_{(a,b)}$, and say $x > y$. Membership of y in $A_{(a,b)}$ is certified by chain C of size a with maximum y. Since $C \cup \{x\}$ is a chain of of size $a+1$ with maximal element x we have contradicted $x \in A_{(a,b)}$. In the case where $x < y$, argue with a chain of size b having y as minimum to obtain a similar contradiction. □

3. On-line chain partitions

The material of this section is in the core of the problem. We start with Szemerédi's lower bound and a new one improved by a factor of almost two. Theorem 3.3 is Kierstead's classic upper bound and Theorem 3.4 is a new sub-exponential upper bound, obtained by Bosek and Krawczyk in 2009. The main ideas for that result will be presented at the end of the section in Subsection 3.1. We present the status of the problem for width 2 and 3. After that we introduce the up-growing version of the problem. In this variant the value of the on-line chain partition game is precisely known. We include a proof of the result.

The lower bound $\binom{w+1}{2} \leqslant \mathrm{val}(w)$ is often attributed to Szemerédi (published in [16]) but in fact Szemerédi is the author of the dual construction for the on-line antichain partition game and Saks is the one who translated it for the chain partition game. Although we are going to prove the same bound in a much more restricted setting (see Theorem 6.2) we would like to share this nice and short construction with the reader. Szemerédi's argument can be improved to obtain the result twice as good.

Theorem 3.1 (Szemerédi)**.** *The value of the on-line chain partition game is at least* $\binom{w+1}{2}$.

Proof. We use induction on w and present a strategy $S(w)$ for Spoiler forcing Algorithm to use $\binom{w+1}{2}$ chains on an order of width w. For $w = 1$ it suffices to introduce a single point. Then, indeed, $\binom{1+1}{2} = 1$ chain is forced.

For $w > 1$ the strategy $S(w)$ consists of two steps. First, Spoiler constructs a colorful chain C of size w. Colors used by Algorithm on C will be blocked for further usage. The construction of C goes as follows. Put initially $C = \emptyset$. As long $|C| < w$, Spoiler introduces a new point x greater than all points in C and incomparable with the rest. If Algorithm uses a new color on x then x is incorporated to C. Otherwise, C remains unmodified (see Fig. 2). Note that each color used by Algorithm on C may be used at most once outside C. Therefore the procedure stops with an order consisting of the colorful chain C with $|C| = w$ and a rest $R(w)$ of at most $w-1$ additional points.

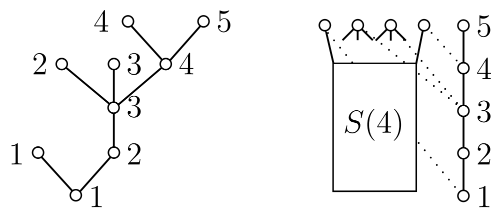

Figure 2. Strategy $S(5)$ for Spoiler.

Now, Spoiler plays $S(w-1)$ in such a way that every new point is incomparable with all elements of C and lies below all elements of $R(w)$. Algorithm is not allowed to reuse the w colors used on C. Using the induction hypothesis for $S(w-1)$ it follows that $w + \binom{w}{2} = \binom{w+1}{2}$ colors are forced in total. As the largest antichain in the order presented by $S(w-1)$ is of size $w-1$ and any such antichain may be extended only by one point (from C), the width of the order is w. □

Theorem 3.2. *The value of the on-line chain partition game is at least* $(2-o(1))\binom{w+1}{2}$.

Proof. The strategy of Spoiler starts by repeating the strategy from Theorem 3.1. The result is an order P_H with an antichain $x_1, \ldots, x_w$ of minimal elements. Each x_i is a bottom of a chain X_i of size i with i different colors used on it. Also, distinct X_i's are totally incomparable. Next, essentially the same strategy is played in an up-side down way below P_H. The result is an order P_L with an antichain $y_1, \ldots, y_w$ of maximal elements such that $x_i > y_j$ for all i, j where each y_i is a top of an i-colorful chain Y_i.

We claim that there is an index i such that $|Y_w|+|X_i| \geqslant 2w-\sqrt{2w}$ (the sizes of X_i and Y_w are the same as the number of different colors on them). Given this index i Spoiler continues recursively with the strategy for width $w-1$ such that all the new points are below each of $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_w$ and their successors, above $y_1, \ldots, y_{w-1}$ and their predecessors but incomparable with X_i and Y_w. It follows that the colors used in $X_i \cup Y_w$ can not be used again. By induction we find that the number of chains forced by Spoiler is

$$\sum_{k=1}^{w}(2k-\sqrt{2k}) \geqslant 2\binom{w+1}{2} - w\sqrt{2w} = (2-o(1))\binom{w+1}{2}.$$

It remains to prove the claim. To minimize the maximum of $|X_i \cup Y_w|$ it is best to have $|X_i \cap Y_w| = k-(w-i)$ for all $i > w-k$. In this case $|X_i \cup Y_w| = 2w-k$ for all $i > w-k$. Of course we have to respect the fact that $|Y_w| = w$ and hence $\sum_{i>w-k}(k-w+i) \leqslant w$. This implies $\binom{k+1}{2} \leqslant w$, i.e., $k^2+k \leqslant 2w$ and finally $k < \sqrt{2w}$. □

Theorem 3.3 (Kierstead [15])**.** *The value of the on-line chain partition game is at most* $\frac{5^w-1}{4}$.

A good outline of the beautiful proof of the theorem is given in Trotter's chapter [32] in the *Handbook of Combinatorics*. The strength of this result may be measured by the fact that no progress has been made for more than 25 years. Only in 2009, Bosek and Krawczyk managed to improve the upper bound. Their new sub-exponential bound is:

Theorem 3.4 (Bosek and Krawczyk)**.** *The value of the on-line chain partition game is at most* $w^{16\lg w}$.

We give a sketch of the quite involved proof further in Section 3.1.

Additionally, in the paper from 1981 Kierstead presented a general lower bound $4w-3 \leqslant \mathrm{val}(w)$. Today the precise value of $\mathrm{val}(w)$ is known only for $w \leqslant 2$, where $\mathrm{val}(2) = 5$ (by Kierstead's lower bound and Felsner's upper bound given in [11]). In the next case $w = 3$ there is still a gap. Recently, Bosek [3] improved the upper bound and the current state of the art is: $9 \leqslant \mathrm{val}(3) \leqslant 16$.

The strategy for Spoiler enforcing 5 colors on orders of width 2 can be plugged into Szemerédi's strategy and with a few ideas from the proof of Theorem 3.2 it can produce a slightly better lower bound for $\mathrm{val}(w)$. The authors claim that $\mathrm{val}(w) \geqslant (\frac{5}{4}-o(1))w^2$. Nevertheless, the latter seems to be pretty far from the best possible result.

Note that partitioning an order of width w into n chains is equivalent to coloring a co-comparability graph of clique size at most w using n colors. Consider, for a moment, an on-line coloring game in which Spoiler introduces a graph and Algorithm maintains a proper coloring. Such a game on coloring graphs is more challenging for Algorithm than the previous on-line chain partition game as a co-comparability graph does not convey the information about the direction of the poset. In particular, Kierstead's algorithm from Theorem 3.3 made explicit use of the orientation of the order relation. These considerations led Schmerl to ask whether there exists a strategy for Algorithm in the on-line coloring game on co-comparability graphs with cliques of size at most w using a certain number of colors bounded in terms of w. Schmerl's question has been answered by Kierstead, Penrice and Trotter in [20]. They show that for every tree T of radius two there exists a function $f_T : \mathbb{N} \to \mathbb{N}$ such that there is a strategy for Algorithm in on-line coloring games on graphs of clique-size at most w and without T as an induced subgraph which uses at most $f_T(w)$ colors. In other words, if Spoiler is not allowed to produce an induced copy of T then there is a reasonable strategy for Algorithm. Let S be the subdivision of $K_{1,3}$. Clearly S is a radius two tree. As co-comparability graphs do not contain an induced S the question posed by Schmerl is answered affirmatively. A more detailed account to on-line coloring games on graphs can be found in the survey [18] by Kierstead which includes a proof that the class of graphs that have no induced S is on-line χ-bounded.

Felsner [11] introduced a variant of the chain partitioning game in which Spoiler's power is limited by the condition that the new element has to be a maximal element of the order presented so far. In other words, a possible comparability of a new element x to an old element y has to be of the form $x > y$. On-line posets with this property are called *up-growing*. Felsner determined the precise value of the game for up-growing orders. In this paper, the lower bound is as a consequence of Theorem 6.2. The following strikingly simple argument for the upper bound is taken from the paper of Agarwal and Garg [1].

Theorem 3.5 (Felsner [11])**.** *The value of the on-line chain partition game for up-growing orders of width w is $\binom{w+1}{2}$.*

Proof of the upper bound. Algorithm maintains a family $F_1, \ldots, F_w$ of sets of chains where F_i contains at most i chains. Together, all the chains form a partition of the present order. Denote by $\text{Tops}(F_i)$ the set of maximum elements (*tops*) of chains from F_i. The invariant maintained by Algorithm is the following:

$$\text{Tops}(F_i) \text{ is an antichain, for every } i.$$

Now, suppose that Spoiler has just introduced a new maximal point x. Let j be the least number such that $|F_j| < j$ or there is a point in $\text{Tops}(F_j)$ which is dominated by x. Such j does exist as otherwise F_w would have to be of size w and x would have to be incomparable with all w points from $\text{Tops}(F_w)$, so that the set $\{x\} \cup \text{Tops}(F_w)$ would form an antichain of size $w + 1$.

If j is determined and x is comparable with the top of some chain $C \in F_j$, then Algorithm adds x to C. Otherwise, if x is incomparable to all elements in $\text{Tops}(F_j)$ but $|F_j| < j$ then Algorithm defines a new chain $C = \{x\}$ and introduces it into F_j.

Since x may have been comparable to several elements in $\mathrm{Tops}(F_j)$ something has to be done to restore the invariant. This can only happen if $j > 1$. In this case Algorithm modifies the families F_{j-1} and F_j as follows (new F_i's are marked with a plus sign):

$$F^+_{j-1} = F_j - \{C\}, \quad F^+_j = F_{j-1} \cup \{C\}.$$

From the choice of j it follows that the invariant is again true. The total number of chains used by Algorithm is bounded by $1 + 2 + \ldots + w = \binom{w+1}{2}$. □

3.1. **The sub-exponential upper bound – sketch of the proof.** We give a brief account of the main ideas of the proof of Theorem 3.4, i.e., the sub-exponential upper bound $\mathrm{val}(w) \leqslant w^{16 \lg(w)}$.

In the first part of the proof, the general chain partition problem is reduced to a family of instances of a more structured problem, called a *regular game*. The second part is a description and analysis of the algorithm for the regular game.

The regular game of width k is an on-line game with players Spoiler and Algorithm. The description is based on the notion of a *regular board*. A regular board after t turns is a poset $(\bigcup_{i=1}^{t} A_i, \leqslant)$ of width k. All the A_i's are antichains of size k. They are pairwise disjoint and linearly ordered with respect to $\sqsubseteq$, where $X \sqsubseteq Y$ if for all $x \in X$ there is $y \in Y$ with $x < y$. Each antichain A_i is introduced by Spoiler during his round as one atomic move. The index i represents the time when the antichain was introduced into the board. The first two antichains A_1, A_2 are fixed to be the borders of the board, i.e., $a_1 < a_2$ for all $a_1 \in A_1, a_2 \in A_2$ and all further antichains are to be presented in between A_1 and A_2 with respect to $\sqsubseteq$. Let $A_{p(i)}$ and $A_{s(i)}$ denote the immediate predecessor and the immediate successor of A_i in the $\sqsubseteq$-order at time i.

- Orders $(A_{p(i)} \cup A_i, \leqslant)$ and $(A_i \cup A_{s(i)}, \leqslant)$ are strong orders; where $(X \cup Y, \leqslant)$ is a *strong order* if for every two comparable points $x \in X$, $y \in Y$ there is a minimum-size chain partition of $(X \cup Y, \leqslant)$ with x, y in the same chain.

The move of Spoiler on the board at round $t \geqslant 3$ begins with a choice of two consecutive antichains in the $\sqsubseteq$-order, they will become $A_{p(t)}$ and $A_{s(t)}$. Next Spoiler presents a new antichain A_t and strong orders $(A_{p(t)} \cup A_t, \leqslant_1)$ and $(A_t \cup A_{s(t)}, \leqslant_2)$ such that the transitive closure of $\leqslant_1 \cup \leqslant_2$ restricted to $A_{p(t)} \cup A_{s(t)}$ is a subset of $\leqslant$. The board after t turns is $(\bigcup_{i=1}^{t} A_i, \leqslant^+)$, where $\leqslant^+$ is the transitive closure of $\leqslant \cup \leqslant_1 \cup \leqslant_2$. In particular, $(\bigcup_{i=1}^{t-1} A_i, \leqslant)$ is the induced suborder of $(\bigcup_{i=1}^{t} A_i, \leqslant^+)$ exactly as one should expect in an on-line setting (see Fig. 3).

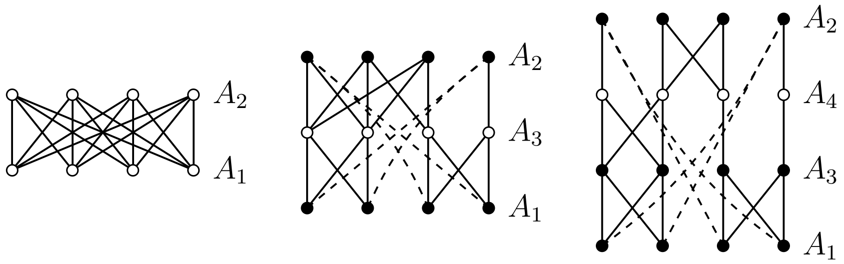

FIGURE 3. The first two moves of Spoiler in a regular game of width 4.

The reply of Algorithm is a coloring of the elements of A_t such that all points in the game with the same color form a chain.

The reduction from the general chain partition problem to the regular game is done in two steps. First the order P is split into a sequence $P_1, \ldots, P_w$ of suborders such that the width of $P_1 \cup \ldots \cup P_i$ is at most i. This is done on-line by assigning the new point x to the first P_i where it does not violate the width constraint. Each P_i is then used to construct a regular game of width i such that the coloring produced in this regular game yields a chain partition of P_i. The essence of the reduction is captured in the following proposition:

Proposition 3.6. *If Algorithm has a strategy which uses at most* $\operatorname{reg}(v)$ *chains on a regular game of width* v *then there is a strategy for Algorithm which uses at most* $\sum_{v \leqslant w} \operatorname{reg}(v) \leqslant w \cdot \operatorname{reg}(w)$ *chains in the general on-line chain partition game for orders of width* w*.*

Now, we sketch the strategy of Algorithm for the regular game. During round t, just before coloring the points of an incoming antichain, Algorithm assigns a color to each comparability edge (x, y) of the incoming strong orders, i.e., $x <_1 y$ in $(A_{p(t)} \cup A_t, \leqslant_1)$ or $x <_2 y$ in $(A_t \cup A_{s(t)}, \leqslant_2)$, in such a way that

($\star$) the set of all points incident to edges colored with γ is a chain in $\leqslant$.

The next step is easy. To $x \in A_t$, Algorithm assigns a color of any edge incident to the vertex x. Condition ($\star$) guarantees that all points with the same color lie in one chain. Therefore, in the following we focus on coloring new edges of the incoming strong orders.

Algorithm's edge-coloring strategy is based on the idea of a node. A *node* is a connected component (in the comparability graph of the order) of one of the strong orders presented by Spoiler during the game. From the definition of strong order and because width is at most k it simply follows that a node has the same number of minimal and maximal points. Also, each edge belongs to exactly one node. The essential property of nodes is:

– The set of all nodes of strong orders in a regular game can be organized in a tree T, called the *tree of the game* (see Fig. 4). The root of T is the node (A_1, A_2).

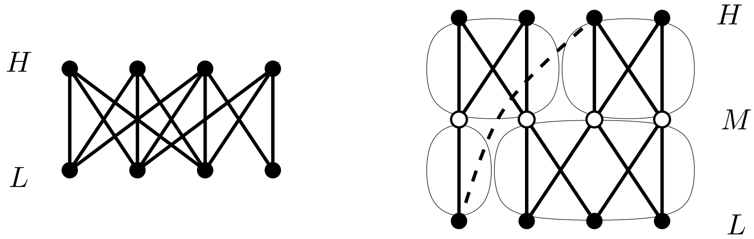

FIGURE 4. A node in the strong order (L, H) and its four sons in strong orders (L, M) and (M, H).

The *characteristics* of a node $N = (L, H)$, where L is the lower and H is the higher level of N, consists of its $\operatorname{width}(N) = |L| = |H|$ and its *surplus* $s(N)$ which is the largest k such that for all non-empty $X \subseteq L$ we have $|\operatorname{succ}(X)| \geqslant \min\{|X| + k, |H|\}$, where $\operatorname{succ}(X)$ denotes the successors in H of elements of X. For N being a complete bipartite graph the condition is true for every k and we put $s(N) = \infty$. Note that by the definition of strong order $s(N) \geqslant 1$ for every node N. A useful property of the characteristics is that if N' is a descendant of N in the tree T then $\text{width}(N') \leqslant \text{width}(N)$ and $\text{width}(N') = \text{width}(N)$ implies $s(N') \leqslant s(N)$, i.e., the pairs (width, surplus) of characteristics are weakly decreasing with respect to the lexicographic order along paths in the tree.

A *cross* of a node $N = (L, H)$ is a set $\{x_1, x_2, y_1, y_2\}$ with $x_1, x_2 \in L$, $y_1, y_2 \in H$, with the four relations $x_i < y_j$, such that there is an extension of chains $\{x_1, y_1\}$, $\{x_2, y_2\}$ to a minimum-size chain partition of N. A node is *vital* if it contains a cross. For each vital node N a representative cross $X(N)$ is fixed.

A vital node N with characteristics (u, s) is called *active* if it has no ancestor in the tree with the same characteristics. On the set of active nodes with characteristics (u, s) define an order $P(u, s)$ by the rule that $N <_{(u,s)} N'$ iff there is a maximum $y \in X(N)$ and a minimum $x' \in X(N')$ with $y \leqslant x'$. The key property of $P(u, s)$ is:

– The width of $P(u, s)$ is at most $w/2$.
  Algorithm recursively generates an on-line chain partition of $P(u, s)$.

For a chain C in $P(u, s)$ consider the set $\mathcal{E}(C)$ of all edges of nodes in C, i.e, $\mathcal{E}(C) = \{(x, y) :\ x \in L,\ y \in H,\ x < y \text{ and } (L, H) \in C\}$. On the set of these edges define the order relation $<_E$ where $(x, y) <_E (x', y')$ if and only if $y \leqslant x'$. The key properties of $(\mathcal{E}(C), <_E)$ are:

– $(\mathcal{E}(C), <_E)$ is $(\mathbf{2w-1}) + (\mathbf{2w-1})$-free and its width is at most w^3.
  Hence, First-Fit can partition this order on-line using at most $3(2w-1)(w^3)^2$ chains (cf. Section 7).

There are only w^2 possible characteristics (u, s). Suppose that Algorithm can partition on-line orders of width $v < w$ into $\mathsf{alg}(v)$ chains. Then we can summarize the result of this part as:

Proposition 3.7. *There is a strategy for Algorithm to color the edges of all active nodes with at most* $\lambda(w) = 3(2w-1)(w^3)^2 \cdot w^2 \cdot \mathsf{alg}(\frac{w}{2})$ *colors in such a way that* $(\star)$ *is preserved.*

It remains to take care of non-active nodes, or more precisely, of edges lying in non-active nodes of strong orders presented by Spoiler in the regular game. With an active node N we associate a set $D(N)$ of *dependent nodes*. It is the set of nodes N' such that N is the first active node on the path from N' to the root of T. Since (A_1, A_2), the root of T, is active, the set $\{D(N) :\ N \text{ is active}\}$ forms a partition of all nodes in T.

The basic idea is to replace each of the $\lambda(w)$ colors used for the edges of active nodes (Proposition 3.7) by a bundle of μ colors. Then the colors in the bundles associated with the edges of an active node N are used to color the edges of all nodes $N' \in D(N)$.

An easy but important property of non-vital nodes is:

– All descendants of a non-vital node are also non-vital and therefore if N' is a non-vital node in $D(N)$ then all descendants of N' are also in $D(N)$.

Although a non-vital $N' \in D(N)$ may have a lot of descendants, the fact that it does not contain a cross results in:

– There is a greedy strategy that extends an edge coloring of a non-vital node N' to an edge coloring with property $(\star)$ of all edges of descendants of N'. This extension does not require additional colors.

Now, in order to color all the edges in $D(N)$ it remains to deal with the edges of vital nodes and of first non-vital children of N in the tree of the game (we briefly call them *first-non-vital nodes*). Unless N represents a complete bipartite graph we have:

– All vital nodes in $D(N)$ have the same characteristics as N and they form a path in the tree T (see Fig. 5).

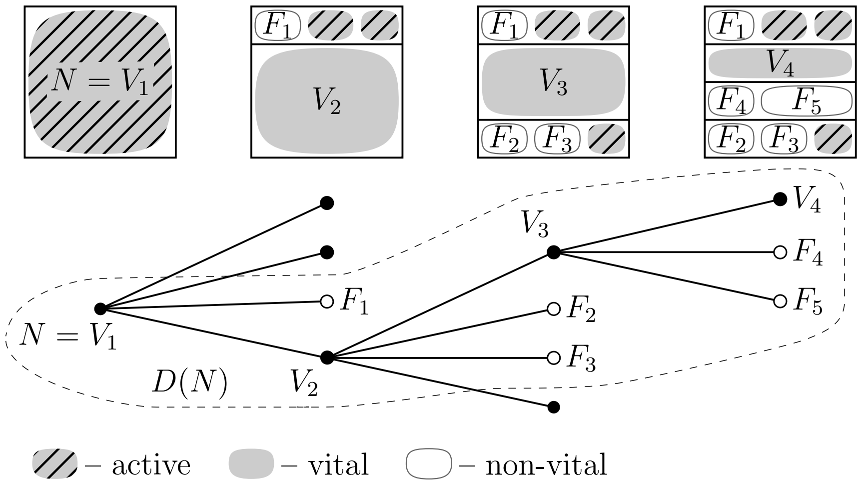

FIGURE 5. The tree-structure and the path of vital nodes in $D(N)$.

Note that consecutive vital nodes on a path in $D(N)$ split first-non-vital nodes in $D(N)$ into two regions. Let V be the last vital node on this path and let $A(N)$ respectively $B(N)$ be the edges of first-non-vital nodes above respectively below V in the sense of $<_E$. Define the order relation $<_E$ (the same as previously) on the edges of $A(N)$ and $B(N)$. Now, consider the orders $(A(N), <_E)$, $(B(N), <_E)$ as on-line orders:

– The orders $(A(N), <_E)$ and $(B(N), <_E)$ are down-growing and up-growing orders of width at most w^3, respectively. Hence, each of these orders can be partitioned on-line into at most $\binom{w^3+1}{2}$ chains (cf. Theorem 3.5).

To an edge $z < u$ in $A(N) \cup B(N)$ we want to assign a color that is used on some edge $x < y$ of N such that property $(\star)$ is preserved. That is we need $x \leqslant z < u \leqslant y$. Such a color assignment is certainly possible if only every edge $x < y$ of N has $2\binom{w^3+1}{2}$ colors in its bundle.

It remains to color edges of vital nodes and possibly first-non-vital nodes that appear as sons of the last vital node in $D(N)$. To take care of all these edges it is sufficient to have two additional colors in the bundle of every edge $x < y$ in N.

The case where N represents a complete bipartite graph can be handled with similar ideas.

Proposition 3.8. *If each edge of an active node N is colored with a bundle of $2\binom{w^3+1}{2}+2$ colors then Algorithm can color the edges of nodes in $D(N)$ using only colors from the bundles on edges of N.*

The combination of the previous three propositions yields the following:

$$\begin{aligned} \mathsf{alg}(w) &\leqslant w \cdot (2\binom{w^3+1}{2} + 2) \cdot 3(2w-1)(w^3)^2 \cdot w^2 \cdot \mathsf{alg}(\frac{w}{2}) \\ &\leqslant \mathsf{poly}(w) \cdot w^{16 \lg w}. \end{aligned}$$

For more details we send the reader to [4].

4. Interval orders

An order $\mathbf{P} = (X, \leqslant)$ is an *interval order* if there is a function I which assigns to each $x \in X$ a closed interval $I(x) = [l_x, r_x]$ on the real line $\mathbb{R}$ so that $x < y$ in $\mathbf{P}$ if and only if $I(x) < I(y)$, i.e., $r_x < l_y$. The function I is called a representation of $\mathbf{P}$; see Fig. 6 for an example. Fishburn [13] characterized interval orders as the orders without induced $(\mathbf{2} + \mathbf{2})$, i.e., without four elements a, b, c, d such that $a < b$ and $c < d$ are the only comparabilities.

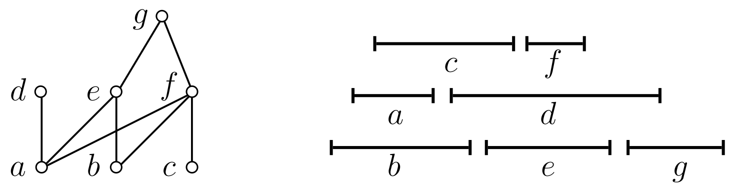

Figure 6. Interval order $\mathbf{P} = (\{a, b, c, d, e, f, g\}, \leqslant)$ with its representation.

We start off with a result for antichain partitioning. Gyárfás and Lehel [14] proved that any chordal graph $\mathbf{G}$ can be covered on-line with $2\alpha(\mathbf{G}) - 1$ cliques (where $\alpha(\mathbf{G})$ is the maximum size of an independent set in $\mathbf{G}$). This immediately implies that the value of the on-line antichain partition game for interval orders of height h is at most $2h - 1$, and this bound is tight (see [21]).

The value of the on-line chain partition game for interval orders was settled in the early 80's by Kierstead and Trotter. Like all other results at that time it was expressed in the language of recursive combinatorics. Several years later Chrobak and Ślusarek proved the same result, this time using the terminology of on-line algorithms.

Theorem 4.1 (Kierstead, Trotter [23]; Chrobak, Ślusarek [9]). *The value of the on-line chain partition game for interval orders of width w is $3w - 2$.*

There is one subtle issue distinguishing the two results. In the on-line games considered so far Spoiler always presented an on-line order as a set of points. Interval orders can be presented in a new way: not as points, but as intervals. In this new variant of the game Spoiler adds some extra information to the order. The task for Algorithm remains the same, i.e., assign colors to intervals in such a way that two intersecting intervals have always a different color. The corresponding notion for the width of the poset is the clique-size – the maximum size of the set of mutually intersecting intervals. This new variant of the game is called a variant *with representation*. Kierstead and Trotter analyzed the variant without representation. Chrobak and Ślusarek analyzed the variant with representation. Below we recall Spoiler's strategy for intervals and Algorithm's strategy for the case without representation.

The same argument works also for a game in which Spoiler presents arcs on a circle and Algorithm colors them avoiding monochromatic intersections. Ślusarek [30] showed that the value of this game remains $3w - 2$, here w denotes the maximum size of a set of arcs sharing a point on the circle. The proof does not work when Spoiler presents a circular arc graph without underlying arc representation. The problem is that not all cliques in a circular arc graph admit a representation with a non-empty common intersection and the argument relies on such cliques.

Proof of Theorem 4.1. First, for the lower bound, we provide a strategy $S(w)$ for Spoiler forcing Algorithm to use at least $3w-2$ colors on a collection of intervals of clique-size at most w. The strategy $S(1)$ is trivial, it suffices to present a single interval. For strategy $S(k+1)$ Spoiler plays many strategies $S(k)$ on disjoint areas of the real line. On each copy of $S(k)$ Algorithm has to use at least $3k-2$ colors (by induction). If $3k+1$ or more colors are used in total, we are done. Otherwise, Algorithm only has $\binom{3k}{3k-2}$ possible selections of colors for a single copy of $S(k)$. When the number of $S(k)$'s is large enough, Spoiler forces four of them, say C_1, C_2, C_3, C_4 (read from left to right) to use the same set of $3k-2$ colors. Now, Spoiler introduces two intervals: the first covers all intervals from C_1 and is disjoint with the rest, the other covers C_4, again being disjoint from the rest (see Fig. 7). On both of these intervals (and also on all the following ones) Algorithm has to use colors that have not been used on the C_i's. If the same color is used for both new intervals then Spoiler introduces the next two as in the left part of Fig. 8. Otherwise, if Algorithm uses two different colors, then the third color is forced by presenting an interval as shown in the right part of Fig. 8.

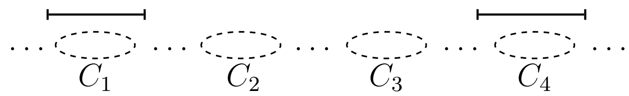

FIGURE 7. $S(k+1)$: Two intervals intersecting C_1 and C_4.

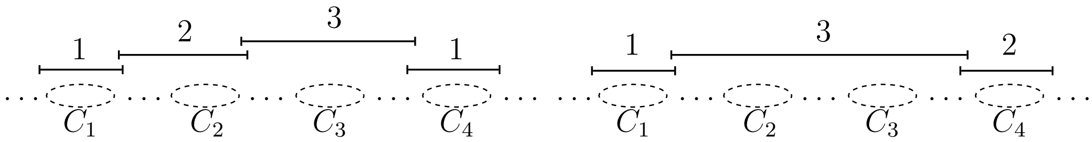

FIGURE 8. $S(k+1)$: Algorithm has to use three different colors.

If all intersections between the new intervals are restricted to the gaps between consecutive $S(k)$'s, then the clique size of the resulting collection of intervals is at most $k+1$. Since Spoiler forced at least $(3k-2)+3$ colors we are done.

In order to prove the upper bound we present a strategy for Algorithm using at most $3w-2$ chains on any interval order of width at most w. We use induction and assume that strategies $A(k)$ that handle interval orders of width $k < w$ with $3k-2$ colors exist. Strategy $A(1)$ has to color an order of width one with one color.

Strategy $A(w)$ maintains a partition of the order into two sets G and R such that the width of G is bounded by $w-1$. A new point x is put into G if it does not violate the width condition for $G \cup \{x\}$. Otherwise, x is put into R. To deal with points in G algorithm $A(w)$ recursively calls $A(w-1)$. By induction at most $3w-5$ colors are used on G. It suffices to show that points in R may be colored on-line using 3 chains.

To visualize the argument we fix an interval representation and identify points with their intervals. Each interval $r \in R$ belongs to clique of size w together with $w-1$ elements $g_1, \ldots, g_{w-1}$ from G. Let $\gamma(r)$ be any point on the real axis in the intersection $g_1 \cap \ldots \cap g_{w-1} \cap r$ of intervals. For $r' \in R - \{r\}$ we note that $\gamma(r) \in r'$ would prove the existence of an antichain of size $w+1$ (see Fig. 9). Hence, no two intervals from R

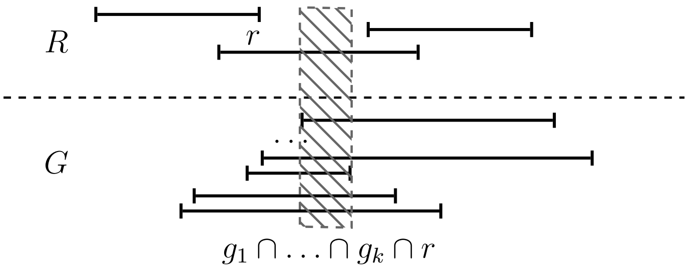

FIGURE 9. $\gamma(r) \notin r'$ for all $r' \in R - \{r\}$.

contain each other. In particular, every $r' \in R$ intersecting r must contain an endpoint of r. Moreover, no interval from R is contained in the union of all other intervals from R. This implies that there is no point on the real axis that is contained in three or more intervals from R.

From these considerations it follows that the co-comparability graph of the order induced on R is a subgraph of a path. A greedy strategy can color such graphs on-line with three colors. A coloring corresponds to a partition of R into three chains. □

The up-growing variant of the game for interval orders has also been explored. When Spoiler presents points (not intervals) then the value is $2w - 1$, proved in [2]. Below we present an argument for an upper-bound which is much shorter than the original one. The observation is that the on-line algorithm from Theorem 4.1 is also optimal in the up-growing setting.

Theorem 4.2 (Baier, Bosek, Micek [2])**.** *The value of the on-line chain partition game for up-growing interval orders of width w is $2w - 1$.*

Proof of the upper bound. The strategy $A(w)$ for Algorithm is the same as in the proof of Theorem 4.1. We are going to induct that $A(w)$ uses at most $2w - 1$ chains on any up-growing interval order of width at most w. Assume that this is true for all naturals up to $k < w$ and consider $A(w)$.

The set G is recursively covered by $A(w-1)$ using $2w-3$ chains and all we have to do is to show a way to cover points in R with only 2 chains. Suppose that x is a maximal point just introduced by Spoiler and it is put in R.

For visualization purposes we again fix an interval representation and identify points with their intervals.

Following the proof of Theorem 4.1 all intervals $r \in R$ intersecting x must contain one of x's endpoints. In fact, due to the up-growing restriction, more is true:

– every $r \in R$ intersecting x has to contain the left endpoint of x.

To see this recall that by the definition of $\gamma(r) \in r$ the intervals containing $\gamma(r)$ form an antichain of size w. If r would contain the right endpoint of x, then $\gamma(r)$ would also be to the right of x. Since there is no antichain of size $w + 1$ this implies that one of the intervals containing $\gamma(r)$ is completely to the right of x. This is impossible since the new element x has to be a maximal element.

From the proof of Theorem 4.1 we know that there is at most one interval in R that contains the left endpoint of x, i.e., x is incomparable to at most one element of R. It follows that Algorithm can use the obvious greedy strategy to cover R with two chains. □

The up-growing case with representation does not pose a big challenge for Algorithm. In this setting there is enough information to make even the Nearest-Fit algorithm use the optimal (off-line) number of colors.

Theorem 4.3 (Broniek [8]). *The value of the on-line chain partition game for up-growing interval orders of width w presented with representation is w.*

Proof. The strategy for Algorithm: from all legal colors (i.e., colors not used for intervals intersecting the new interval) choose the closest one used rightmost (in other words, choose a legal color used on an interval with the right endpoint nearest to the new interval). We prove that this strategy (called the Nearest-Fit algorithm) uses no more colors than the clique-size of the presented collection of intervals.

Let top(γ) denote the top element of the γ-chain, this is the rightmost interval colored with γ. Let $x = [l_x, r_x] = \text{top}(\alpha)$ be the interval with the leftmost right endpoint from all top elements.

We claim that r_x is contained in an interval of each color used by Algorithm. Consider any color β used during the game. The right endpoint of top(β) is to the right of r_x. If top(β) contains r_x then it is the interval we are looking for. Otherwise, $\beta \neq \alpha$ and top(β) is completely to the right of r_x. Now, let y be the leftmost interval among those colored with β and completely to the right of r_x (see Fig. 10). By the up-growing property, x must have been presented prior to y. The Nearest-Fit algorithm colored y with $\beta \neq \alpha$ while α was also legal for y. This means that there is an interval z of color β to the left of y but with $r_z > r_x$. Our choice of y implies that z contains r_x.

Hence, for all the β's we have found an interval colored with β and containing r_x. Therefore, the number of intervals containing r_x is at least the number of colors used by Algorithm. As these intervals form an antichain, the proof is finished. □

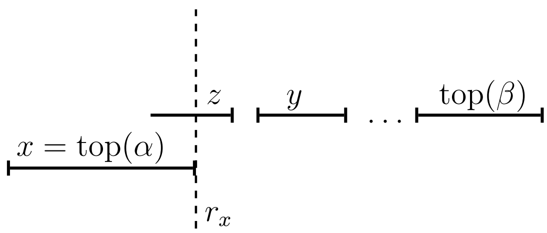

FIGURE 10. Each color is used on an interval containing r_x.

5. Semi-orders

An order $\mathbf{P} = (X, \leqslant)$ is a semi-order if there is a function I assigning to each point $x \in X$ a closed, unit-length interval $I(x) = [l_x, l_x + 1]$ of the real line $\mathbb{R}$ so that for all $x, y \in X$ we have $x < y$ in $\mathbf{P}$ iff $l_x + 1 < l_y$. In other words, an interval order is a semi-order if it has a representation formed by unit-length intervals. An interval representation is *proper* if there is no inclusion between intervals. Proper interval orders are the interval orders admitting a proper representation. It is a well known theorem of Roberts [28], that the classes of proper interval orders and semi-orders coincide. A representation-free characterization of semi-orders is due to Scott-Suppes [29]: an orders is a semi-order exactly if it has no induced $(\mathbf{2}+\mathbf{2})$ and no induced $(\mathbf{3}+\mathbf{1})$, this is a four element order on a, b, c, d such that $b < c < d$ are the only comparabilities.

The on-line chain partition game for semi-orders in its most general form is relatively easy to analyze.

Proposition 5.1. *The value of the on-line chain partition game for semi-orders of width w is $2w-1$.*

Proof. The strategy for Spoiler forcing $2w-1$ chains on a semi-order of width w is as follows:

Phase 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 $(2 \leqslant k \leqslant w)$ are used twice, once in A on $a_1, \dots, a_k$ and once in B, on $b_1, \dots, b_k$ respectively so that a_i and b_i have the same color.

Phase 2. Present $k-1$ incomparable points $x_1, \dots, x_{k-1}$ such that the only comparabilities are $a_1, \dots, a_i < x_i < b_{i+1}, \dots, b_k$. Their interval representation may look as in Fig. 11. The width of the resulting order is w. It is easy to verify that Algorithm is forced to use $2w-1$ chains as each x_i has to go into a new chain.

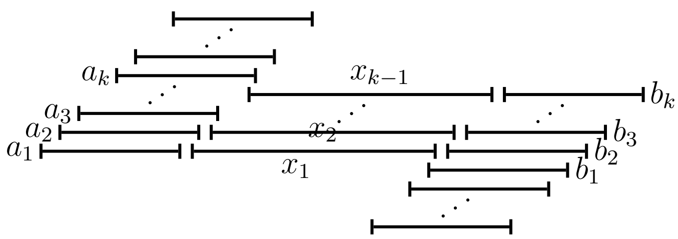

FIGURE 11. Strategy for Spoiler forcing Algorithm to use $2w-1$ chains on a semi-order of width w.

To prove the upper bound we show that a greedy strategy for Algorithm never needs more than $2w-1$ chains. This fact will become quite obvious with a little help of geometry. Fix a proper representation of the order and identify points with their intervals. Let x be the new point and let $\mathrm{Inc}(x)$ denote the set of points incomparable with x. The only chains forbidden for x are those used in $\mathrm{Inc}(x)$. If $y \in \mathrm{Inc}(x)$ then intervals x and y intersect. Moreover, since y cannot lie in the interior of x, it must contain one of the endpoints of x. The number of intervals sharing a common point does not exceed the width of the order w. This implies that $|\mathrm{Inc}(x)| \leqslant 2(w-1) = 2w-2$, proving that at least one out of a set of $2w-1$ chains is legal for x. □

The analysis of the up-growing case turned out to be much more involved. The result is shown in the next theorem. The proof can be found in an independent paper [12].

Theorem 5.2 (Felsner, Kloch, Matecki, Micek [12])**.** *The value of the on-line chain partition game for up-growing semi-orders of width w is $\lfloor \frac{1+\sqrt{5}}{2} w \rfloor$.*

We now turn to the variants where the semi-order is presented together with a representation. There are two variants:

(i) $\mathbf{P}$ is presented with unit intervals.

(ii) **P** is presented with a proper representation.

In both cases the value of the game is still unknown. Rather unsatisfactory bounds are given in Proposition 5.3.

Proposition 5.3. *The value of the on-line chain partition game for semi-orders of width w presented with representation (unit-length or proper) is at least $\lfloor \frac{3}{2}w \rfloor$ and at most $2w - 1$.*

Proof. The upper bound is valid for any greedy algorithm (see Proposition 5.1). For the lower bound we present a strategy for Spoiler which forces $3k$ colors on a collection of unit-length intervals of clique-size $2k$. The strategy is as follows:

Phase 1. Present a clique A of k identical, unit-length intervals. Let $1, \ldots, k$ be the colors used by Algorithm.

Phase 2. Present a clique B of $2k$ unit-length intervals in such a way that:

(i) their left endpoints lie within a unit distance from the right border of A,
(ii) Algorithm is forced to use new colors on the k leftmost intervals from B.

We now explain how to build B satisfying (i) and (ii). Present the first interval so that the distance between its left endpoint and the right border of A is $\frac{1}{2}$. For the rest of the construction maintain a partition of B into two (possibly empty) sets $B_0 \cup B_1$, where B_0 (B_1, respectively) contains intervals with a new (old) color, and additionally all left ends of intervals from B_0 lie to the left of all left ends of intervals from B_1 (see Fig. 12). Introduce any further interval into the gap between B_0 and B_1, i.e., put it slightly to the right of all left ends of B_0 and slightly to the left of all left ends of B_1. Depending on the color used by Algorithm, the new interval extends either B_0 or B_1. Since B has $2k$ intervals and there are at most k old colors used on B, we indeed get $|B_0| \geqslant k$, which is exactly condition (ii).

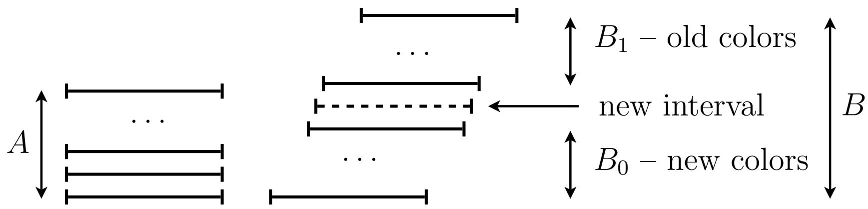

FIGURE 12. Construction of $B = B_0 \cup B_1$.

Phase 3. Present k identical unit-length intervals intersecting A and the k leftmost intervals in B.

The k intervals presented in Phase 3 need new colors. Therefore $3k$ colors have to be used in total but the largest antichain only has size $2k$. □

The upper bound $2w - 1$ is tight for greedy strategies of Algorithm, i.e., strategies using a new color only when they have to (noted in [9]). To force $2w - 1$ colors Spoiler presents two cliques of intervals: $a_1, \ldots, a_w$ and $b_1, \ldots, b_w$, where $l_{a_1} < \ldots < l_{a_w} < r_{a_1} < \ldots < r_{a_w} < l_{b_1} < \ldots < l_{b_w} < r_{b_1} < \ldots < r_{b_w}$. The order of presentation is: $a_1, b_1, \ldots, a_w, b_w$. Clearly, a greedy Algorithm assigns i-th color to a_i and b_i. Now, Spoiler presents $x_1, \ldots, x_{w-1}$ (exactly as in the proof of Proposition 5.1) such that the only comparabilities are $a_1, \dots, a_i < x_i < b_{i+1}, \dots, b_w$ and Algorithm must use $w-1$ new colors. The presented collection is of clique-size w and can be realized by unit-length intervals.

6. d-dimensional orders

An *extension* of an order $\mathbf{P} = (X, P)$ is an order $\mathbf{Q} = (X, Q)$ such that $x \leqslant y$ in $\mathbf{P}$ implies $x \leqslant y$ in $\mathbf{Q}$. If an extension $\mathbf{Q}$ of $\mathbf{P}$ is a linear order, it is called a *linear extension.* A set of linear extensions of $\mathbf{P}$ intersecting to $\mathbf{P}$ is called a *realizer* of $\mathbf{P}$. The *dimension* of $\mathbf{P}$, denoted by $\dim(\mathbf{P})$, is the least number n such that there is a realizer of $\mathbf{P}$ consisting of n linear extensions. This definition is due to Dushnik and Miller in [10]. Clearly, an order is of dimension 1 if and only if it is a chain. The dimension of an antichain A with $|A| > 1$ is exactly 2. Indeed, for any linear extension L of A the set $\mathcal{R} = \{L, L^*\}$ is a realizer, here L^* denotes the reverse of L. For a comprehensive account on the topic and an extensive bibliography we refer the reader to Trotter's monograph [31].

A geometric interpretation of the dimension (and justification of the term) is the following. Denote by $\mathbb{R}^d$ the standard Cartesian product of real numbers, partially ordered by inequality on each coordinate: $(x_1, \dots, x_d) \leqslant (y_1, \dots, y_d)$ if and only if $x_i \leqslant y_i$ for each $1 \leqslant i \leqslant d$. Let $\mathcal{R} = \{L_1, \dots, L_d\}$ be a realizer of a finite poset $\mathbf{P} = (X, \leqslant)$. With every element $x \in X$ we associate the point $(x_1, \dots, x_d)$ so that x_i is the position of x in the linear extension L_i. Such a mapping of X into $\mathbb{R}^d$ defines an embedding of the poset $\mathbf{P}$ into $\mathbb{R}^d$. Conversely, projections of such an embedding onto d coordinates give d linear extensions yielding a realizer of $\mathbf{P}$. An example of such an embedding of a poset into a 2-dimensional grid is shown in Fig. 13.

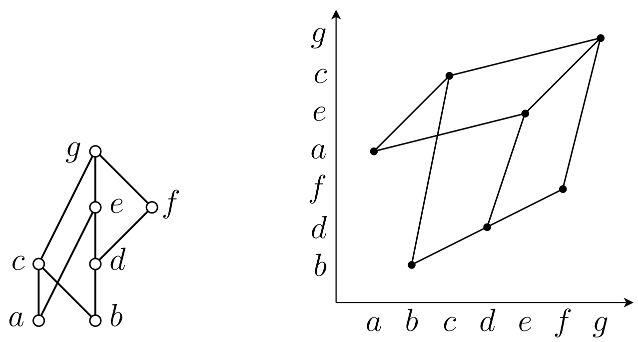

Figure 13. Poset embedded into a 2-dimensional grid.

The analysis of the on-line chain partition game restricted to d-dimensional orders appears to be as hard as the general problem (even for $d = 2$). No better bound, specific for this class, is known. On the other hand there is a nice result of Kierstead, McNulty and Trotter for the game in which Spoiler introduces a d-dimensional order via its embedding into $\mathbb{R}^d$ or equivalently, by providing on-line a realizer of size d.

Theorem 6.1 (Kierstead, McNulty, Trotter [19])**.** *The value of the on-line chain partition game for d-dimensional orders of width w presented with representation is at most $\binom{w+1}{2}^{d-1}$.*

Proof. The workhorse of the proof is the following fact: Y is a chain in a 2-dimensional order $\mathbf{P}$ with realizer $\{L_1, L_2\}$ if and only if Y is an antichain in $\mathbf{P}^*$ defined by a realizer $\{L_1, L_2^*\}$. In particular, an antichain partition of $\mathbf{P}^*$ (obtained e.g. by Theorem 2.1) is a chain partition of $\mathbf{P}$.

We describe the strategy of Algorithm witnessing the upper bound using induction on d. For $d = 1$ Spoiler presents a chain and Algorithm covers it optimally using $1 = \binom{1+1}{2}^0$ chain.

Fix $d > 1$. Let $\mathbf{P}$ be the presented order and $\{L_1, \ldots, L_d\}$ be its realizer given by Spoiler. Consider $\mathbf{P}^* = L_1 \cap \ldots \cap L_{d-1} \cap L_d^*$. Note that every chain in $\mathbf{P}^*$ is an antichain in $\mathbf{P}$ and so height$(\mathbf{P}^*) \leqslant$ width$(\mathbf{P}) \leqslant w$. On the other hand if Y is an antichain in $\mathbf{P}^*$ then the order induced by Y in $\mathbf{P}$ is just $(L_1 \cap \ldots \cap L_{d-1})|_Y$ and therefore Y a subposet of $\mathbf{P}$ with dimension at most $d - 1$.

During the game, Algorithm uses Schmerl's algorithm (see Theorem 2.1) to generate an on-line antichain partition of $\mathbf{P}^*$ of size at most $\binom{h(\mathbf{P}^*)+1}{2} \leqslant \binom{w+1}{2}$. Each antichain A in the partition of $\mathbf{P}^*$ is a suborder of $\mathbf{P}$. Its width is at most w and $L_1|_A, \ldots, L_{d-1}|_A$ is a $(d-1)$ realizer. Therefore, it can recursively be partitioned into $\binom{w+1}{2}^{d-2}$ chains. Altogether the Algorithm uses at most $\binom{w+1}{2}^{d-2+1}$ chains. □

The next theorem deals with up-growing orders presented with a 2-realizer. The motivation to consider such a restricted setting comes from the results of Szemerédi and Felsner (see Theorems 3.1 and 3.5). The poset constructed in the proof of Theorem 3.1 is 2-dimensional but not up-growing. On the other hand, the up-growing order as used in the original proof of Theorem 3.5 was not 2-dimensional. The following result shows that the value $\binom{w+1}{2}$ remains a lower bound even if we consider on-line orders which are both: up-growing and 2-dimensional.

Theorem 6.2. *The value of the on-line chain partition game for 2-dimensional up-growing orders of width w presented with representation is at least $\binom{w+1}{2}$.*

Proof. The argument is inspired by the proof of the lower bound from Theorem 3.5 from [11]. However, we have to take care that all construction steps preserve the dimension. This is achieved by restricting the operations used by the Spoiler's strategy to only very elementary ones. For the description of the operations we need an easy fact about 2-dimensional orders.

Claim 6.3. If $\mathbf{P}$ is a 2-dimensional order with a realizer L_1, L_2 and the maximal elements of $\mathbf{P}$ are ordered as in L_1, i.e., $\max(\mathbf{P}) = \{x_1, \ldots, x_w\}$ and $x_1 <_{L_1} \ldots <_{L_1} x_w$, then their order is reversed in L_2, i.e., $x_w <_{L_2} \ldots <_{L_2} x_1$. We call $(x_1, \ldots, x_w)$ the *sorted antichain* of maximal elements of $\mathbf{P}$.

Given the sorted antichain $(x_1, \ldots, x_w)$ of maxima and two indices $1 \leqslant i \leqslant j \leqslant w$ we introduce the following operations extending the order in an up-growing way:

above$_{i,j}$ Add a new element y with relations $x_i < y, x_{i+1} < y, \ldots, x_j < y$ and all relations implied by transitivity but no others.

left$_{i,j}$ Always preceded by above$_{i,j}$. Add a set $y_{i+1}, y_{i+2}, \ldots, y_j$ of twin elements such that each y_s from this set has relations $x_{i+1} < y_s, \ldots, x_j < y_s$ and all relations implied by transitivity but no others. The index of the element y introduced by the preceding move above$_{i,j}$ is i, i.e., $y_i = y$.

right$_{i,j}$ Always preceded by above$_{i,j}$. Add a set $y_i, y_{i+1}, \dots, y_{j-1}$ of twin elements such that each y_s from this set has relations $x_i < y_s, \dots, x_{j-1} < y_s$ and all relations implied by transitivity but no others. The index of the element y introduced by the preceding move above$_{i,j}$ is j, i.e., $y_j = y$.

The combination of a move above$_{i,j}$ followed by a move left$_{i,j}$ is illustrated in Fig. 14. Throughout the strategy Spoiler repeatedly makes a move of type above$_{i,j}$ and depending on the color given to the new element y, Spoiler completes the operation with a move of type either left$_{i,j}$ or right$_{i,j}$.

Claim 6.4. If L_1, L_2 is a realizer of $\mathbf{P}$ and $\mathbf{P}^+$ is obtained by a move above$_{i,j}$ followed by left$_{i,j}$ then L_1, L_2 can be extended to a 2-realizer L_1^+, L_2^+ of $\mathbf{P}^+$. The same holds if above$_{i,j}$ is followed by right$_{i,j}$. In other words: the operations preserve the 2-dimensionality of the order.

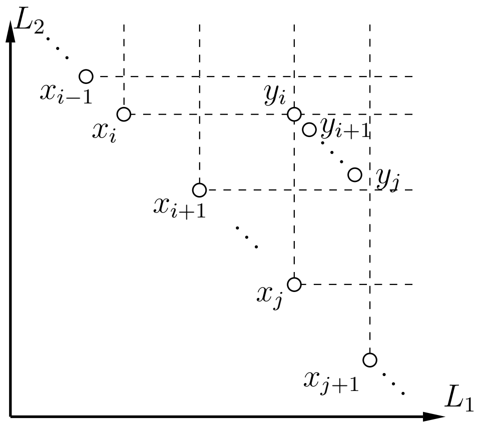

FIGURE 14. Combination of above$_{i,j}$ followed by left$_{i,j}$.

Recall that top(α) is the top element of the α-chain. If x is a maximal element of an order partitioned into chains then private(x) is the set of chains α with top(α) $\leqslant x$ and top(α) $\nleqslant y$ for all maximal elements $y \neq x$. The general idea is to keep track of the number of private chains for the consecutive maxima and make Algorithm produce a large number of them. The workhorse for the proof of the theorem is the following proposition.

Proposition 6.5. *Fix a number $Z \in \mathbb{N}$. Let $\mathbf{P}$ be a 2-dimensional order of width w with sorted antichain $(x_1, \dots, x_w)$ of maximal elements and let a chain partition of $\mathbf{P}$ be given. There is a strategy $S(i,j)$, for all $i \leqslant j$, which extends $\mathbf{P}$ in an up-growing way by using only the three operations described above such that the width remains w and every on-line chain partitioning algorithm has to tolerate one of the following two results for the sorted antichain of maximal elements $(z_1, \dots, z_w)$ of the resulting order:*

(i) $|\text{private}(z_r)| \geqslant r - i + 1$ *for all $r = i, \dots, j$, or*
(ii) *the algorithm has used more than Z colors.*

Moreover for all $s \notin \{i, i+1, \dots, j\}$ we have $z_s = x_s$ and private(x_s) *was not affected by the play of $S(i,j)$.*

Proof. The proof is by induction on $j - i$. For $j = i$ we are in case (i) without doing anything, just observe that the color of the chain to which x_i has been assigned is an element of private(x_i), hence $|\,\text{private}(x_i)| \geqslant 1$.

For the induction step we begin with strategy $S(i+1, j)$ which may result in case (ii) so that we can stop. In the interesting case $S(i+1, j)$ ends with a sorted antichain of maximal elements $(y_1, \ldots, y_w)$ such that $|\,\text{private}(y_r)| \geqslant r - (i+1) + 1 = r - i$ for $r = i+1, \ldots, j$. The next step is a move of type above$_{i,j}$. Let the new element y be assigned to chain γ. We distinguish two cases:

(a) If $\gamma \notin \text{private}(y_j)$ then a move right$_{i,j}$ follows. This results in a new sorted antichain $(y_1, \ldots, y_w)$ of maximal elements with $|\,\text{private}(y_j)| \geqslant j - i + 1$. Playing $S(i, j-1)$ results in one of the two outcomes claimed for $S(i, j)$.

(b) If $\gamma \in \text{private}(y_j)$ then continue with a move left$_{i,j}$. This results in a sorted antichain $(y_1, \ldots, y_w)$ of maximal elements with one more chain γ in the set private(y_i) than before. Continue with another iteration of strategy $S(i+1, j)$. This or one of the following iterations of $S(i+1, j)$ may result in case (a). If case (a) is avoided, then after Z iterations we have $|\,\text{private}(y_i)| \geqslant Z$ and, hence, state (ii) of the proposition. □

To prove the theorem we fix $Z > \binom{w+1}{2}$. Starting with an initial antichain $(x_1, \ldots, x_w)$ apply strategy $S(1, w)$. After completion of $S(1, w)$ we either have reached Z colors, or, the final sorted antichain $(z_1, \ldots, z_w)$ of maximal elements has the property that the private colors of the elements obey $|\,\text{private}(z_i)| \geqslant i$ for each $1 \leqslant i \leqslant w$. Hence, the total number of chains used is at least $1 + 2 + \ldots + w = \binom{w+1}{2}$. □

7. First-Fit

Probably, the simplest strategy for Algorithm in the on-line chain partition game is First-Fit, a strategy assigning the new point to a chain with the smallest possible number. Spoiler can make First-Fit use arbitrarily many chains already on orders of width 2. An example of Kierstead [16], see Fig. 15, shows how to force $3, 4, 5, \ldots$ chains.

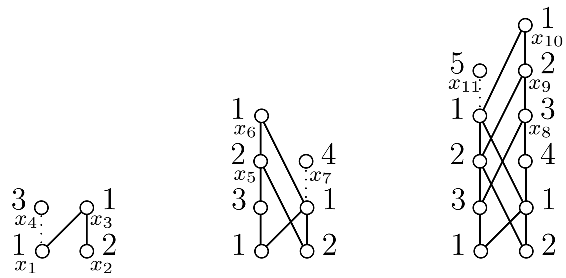

Figure 15. First-Fit forced to use 5 chains on an order of width 2.

Recently, Bosek, Krawczyk and Szczypka [5] proved that First-Fit uses at most $3kw^2$ chains on $(\mathbf{k}+\mathbf{k})$-free orders of width w, i.e., orders with no two incomparable chains of size k. It is likely that indeed First-Fit uses only $O(w)$ chains on $(\mathbf{k}+\mathbf{k})$-free orders (see Problem 3). Note that the case $k = 2$ deals with interval orders.

Several papers investigate the performance of First-Fit for interval orders. This proved to be an exciting and a challenging problem. The upper bound for the number of chains used by FF on interval orders has a long history: $O(w^2)$ by Woodall, $40w$ [17], $26w$ [21], $10w$ [27], and $8w$ [7], [26]. The paper of Pemmaraju and Raman [27] introduced a completely new and elegant technique called the column construction method. The authors achieved an upper bound of $10w$, overlooking one detail leading later to $8w$. From the other side, Chrobak and Ślusarek [9] showed that FF can be forced to use $4.4w - c$ chains on a collection of intervals with clique-size w, for some constant c. Kierstead and Trotter [22] have recently improved this to $4.99w - c$.

8. On-line adaptive chain partitions

On-line adaptive chain partitioning is a variant of the game where Algorithm is stronger than in the standard game. In the adaptive variant Algorithm may assign a non-empty set of colors to the new point. The choice of the set is restricted by the condition that the set of all points containing γ in their set must form a chain. Before coloring an incoming point, Algorithm may remove colors from the sets of some older points. Of course at least one color has to remain for each point. Figure 16 shows an example of an adaptive game.

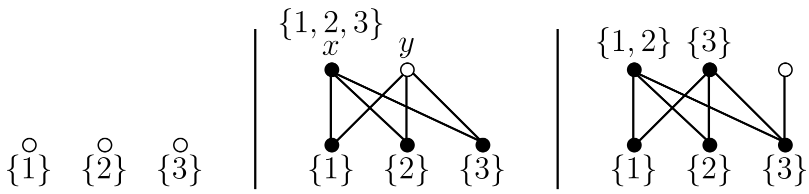

FIGURE 16. Spoiler forces 4 colors on the order of width 3. If Algorithm sticks to three colors, either x or y has only one color upon the arrival of y. In both cases Spoiler may present a point forcing the fourth color.

The value $\text{adapt}(w)$ of the on-line adaptive chain partition game is the least integer s such that Algorithm has a strategy using at most s colors on any on-line order of width at most w. This variant of the game was introduced in [11] in the up-growing variant. The motivation was that the value of this game equals the on-line dimension of up-growing orders.

Very little is known about $\text{adapt}(w)$. In particular, no strategy using substantially less colors than in the original chain partition game is known for Algorithm. Theorem 8.1 gives the best-known lower bound for $\text{adapt}(w)$. We expect that this bound is far from the best possible. Theorem 8.2 is a recent precise result for up-growing orders of height 2.

Theorem 8.1 (Bosek, Micek [6]). *The value of the on-line adaptive chain partition game for up-growing orders of width w is at least* $(2 - o(1))w$.

Theorem 8.2 (Kozik, Matecki [24]). *The value of the on-line adaptive chain partition game for up-growing orders of height at most* 2 *and width w is* $(1 + \pi/\cosh(\frac{\sqrt{3}}{2}\pi) - o(1))w \approx 1.41w$.

9. Open problems

Despite the recent progress, the big challenge in the field of on-line chain partitions remains to lower the gap between upper and lower bound in the unrestricted setting.

Hopefully we have convinced the reader that considering variants and restricted versions of this problem can also lead to interesting structures and beautiful mathematics. We feel that together with the restrictions to special classes of orders, two types of restriction which reduce the power of Spoiler are interesting:

(i) the up-growing case,
(ii) the case where Spoiler has to present the order with a geometric representation which certifies the membership of the order in a given class.

Below is a table of related results and open problems. Columns **U** and **R** of the table indicate whether Spoiler has to play up-growing and with a geometric representation, respectively. In particular it would be very interesting to answer the following questions:

Problem 1. What is the value of the on-line coloring game in which Spoiler presents unit-length/proper intervals? It is likely that the values of these two variants of the chain partition game for semi-orders with representation are different. The best known lower and upper bound is $\frac{3}{2}w$ and $2w-1$, respectively. Moreover, any greedy on-line algorithm may be forced to use $2w-1$ chains.

Problem 2. What is the value of the on-line chain partitioning game for 3-dimensional orders with geometrical representation? In this case the lower and upper bound are $\binom{w+1}{2}$ and $\binom{w+1}{2}^2$, respectively.

Problem 3. Does the First-Fit algorithm use $O(w)$ chains on $(\mathbf{k}+\mathbf{k})$-free orders?

Problem 4. What is the strict bound for the number of colors (chains) used by the First-Fit algorithm on a collection of intervals with clique-size at most w? The current lower bound is $4.99w$ and upper bound is $8w$. Trotter conjectures it to be $5w$.

Problem 5. Is $\text{adapt}(w)$ bounded from above by a polynomial of w? The linear lower bound $(2-o(1))w$ is rather weak.

References

[1] Anurag Agarwal and Vijay K. Garg. Efficient dependency tracking for relevant events in concurrent systems. *Distrib. Comput.*, 19(3):163–183, 2007.

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

[3] Bartłomiej Bosek. *On-line chain partitioning approach to scheduling.* PhD thesis, Jagiellonian University, 2008.

[4] Bartłomiej Bosek and Tomasz Krawczyk. The subexponential upper bound for an on-line chain partitioning problem. Proceedings of the 51th Symposium on Foundations of Computer Science, FOCS10, pp. 347-354, 2010.

[5] Bartłomiej Bosek, Tomasz Krawczyk, and Edward Szczypka. First-fit algorithm for on-line chain partitioning problem. *SIAM Journal on Discrete Mathematics*, 23:1992–1999, 2010.

[6] Bartłomiej Bosek and Piotr Micek. On-line dimension of up-growing orders. unpublished.

[7] Graham Brightwell, Henry A. Kierstead, and William T. Trotter. unpublished result.

[8] Przemysław Broniek. On-line chain partitioning as a model for real-time scheduling. In *Proceedings of the Second Workshop on Computational Logic and Applications (CLA 2004)*, volume 140 of *Electron. Notes Theor. Comput. Sci.*, pages 15–29 (electronic), Amsterdam, 2005. Elsevier.

[9] Marek Chrobak and Maciej Ślusarek. On some packing problem related to dynamic storage allocation. *RAIRO Inform. Théor. Appl.*, 22(4):487–499, 1988.

[10] Ben Dushnik and E. W. Miller. Partially ordered sets. *Amer. J. Math.*, 63:600–610, 1941.

	class	U	R	value	remarks
1	all orders			?	$(2-o(1))\binom{w+1}{2} \leqslant ? \leqslant w^{16\lg(w)}$; Theorem 3.2 and [4]
2	all orders	+		$\binom{w+1}{2}$	[11]
3	interval orders			$3w-2$	[23]
4	interval orders		+	$3w-2$	[23], [9]
5	interval orders	+		$2w-1$	[2]
6	interval orders	+	+	w	[8]
7	semi-orders			$2w-1$	Proposition 5.1
8	semi-orders		+	?	$\frac{3}{2}w \leqslant ? \leqslant 2w-1$; Proposition 5.3
9	semi-orders	+		$\lfloor \frac{1+\sqrt{5}}{2} w \rfloor$	[12]
10	semi-orders	+	+	w	from line 6
11	2-dimensional			?	$\binom{w+1}{2} \leqslant ? \leqslant w^{16\lg(w)}$; from lines 1 and 14
12	2-dimensional		+	$\binom{w+1}{2}$	[19]
13	2-dimensional	+		$\binom{w+1}{2}$	from lines 2 and 14
14	2-dimensional	+	+	$\binom{w+1}{2}$	from line 2 and Theorem 6.2
15	d-dimensional		+	?	$\binom{w+1}{2} \leqslant ? \leqslant \binom{w+1}{2}^{d-1}$; [19]

Table 1. On-line chain partitions of orders: known results and open problems.

[11] Stefan Felsner. On-line chain partitions of orders. *Theoret. Comput. Sci.*, 175(2):283–292, 1997.

[12] Stefan Felsner, Kamil Kloch, Grzegorz Matecki, and Piotr Micek. On-line chain partitions of up-growing semi-orders. submitted.

[13] Peter C. Fishburn. Intransitive indifference with unequal indifference intervals. *J. Mathematical Psychology*, 7:144–149, 1970.

[14] András Gyárfás and Jenö Lehel. On-line and first fit colorings of graphs. *J. Graph Theory*, 12(2):217–227, 1988.

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

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

[17] Henry A. Kierstead. The linearity of first-fit coloring of interval graphs. *SIAM J. Discrete Math.*, 1(4):526–530, 1988.

[18] Henry A. Kierstead. Coloring graphs on-line. In *Online algorithms*, volume 1442 of *Lecture Notes in Comput. Sci.*, pages 281–305. Springer, Berlin, 1998.

[19] Henry A. Kierstead, George F. McNulty, and William T. Trotter. A theory of recursive dimension for ordered sets. *Order*, 1(1):67–82, 1984.

[20] Henry A. Kierstead, Stephen G. Penrice, and William T. Trotter. On-line coloring and recursive graph theory. *SIAM J. Discrete Math.*, 7(1):72–89, 1994.

[21] Henry A. Kierstead and Jun Qin. Coloring interval graphs with First-Fit. *Discrete Math.*, 144(1-3):47–57, 1995.
[22] Henry A. Kierstead and William T. Trotter. First-fit coloring of interval graphs. in preparation.
[23] Henry A. Kierstead and William T. Trotter. An extremal problem in recursive combinatorics. *Congr. Numer.*, 33:143–153, 1981.
[24] Jakub Kozik and Grzegorz Matecki. in preparation.
[25] Rolf H. Möhring. Computationally tractable classes of ordered sets. In Ivan Rival, editor, *Algorithms and Order*, volume 255 of *NATO ASI Series C*. Kluwer Academic Publishers, 1989.
[26] N. S. Narayanaswamy and R. Subhash Babu. A note on first-fit coloring of interval graphs. *Order*, 25(1):49–53, 2008.
[27] Sriram V. Pemmaraju and Rajiv Raman. Approximation algorithms for the max-coloring problem. In *Automata, languages and programming*, volume 3580 of *Lecture Notes in Comput. Sci.*, pages 1064–1075. Springer, Berlin, 2005.
[28] Fred S. Roberts. Indifference graphs. In *Proof Techniques in Graph Theory*, pages 139–146. Academic Press, New York, 1969.
[29] Dana Scott and Patrick Suppes. Foundational aspects of theories of measurement. *J. Symb. Logic*, 23:113–128, 1958.
[30] Maciej Ślusarek. Optimal on-line coloring of circular arc graphs. *RAIRO Inform. Théor. Appl.*, 29(5):423–429, 1995.
[31] William T. Trotter. *Combinatorics and partially ordered sets: Dimension theory*. Johns Hopkins Series in the Mathematical Sciences. Johns Hopkins Univ. Press, Baltimore, MD, 1992.
[32] William T. Trotter. Partially ordered sets. In *Handbook of combinatorics, Vol. 1*, pages 433–480. Elsevier, Amsterdam, 1995.

Algorithmics Research Group, Faculty of Mathematics and Computer Science
Jagiellonian University, Lojasiewicza 6, 30-348 Kraków, Poland
E-mail address: `Bartlomiej.Bosek@tcs.uj.edu.pl`
URL: `http://tcs.uj.edu.pl/Bosek`
E-mail address: `Tomasz.Krawczyk@tcs.uj.edu.pl`
URL: `http://tcs.uj.edu.pl/Krawczyk`
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`

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/`