Online Interval Coloring with Packing Constraints

Leah Epstein[1,⋆] and Meital Levy[2]

[1] Department of Mathematics, University of Haifa, 31905 Haifa, Israel. Email:lea@math.haifa.ac.il
[2] School of Computer Science, Tel-Aviv University, Israel.
Email: levymeit@post.tau.ac.il

Abstract. We study online interval coloring problems with bandwidth. We are interested in some variants motivated by bin packing problems. Specifically we consider open-end coloring, cardinality constrained coloring, coloring with vector constraints and finally a combination of both the cardinality and the vector constraints. We construct competitive algorithms for each of the variants. Additionally, we present a lower bound of 24/7 for interval coloring with bandwidth, which holds for all the above models, and improves the current lower bound for the standard interval coloring with bandwidth problem.

1 Introduction

We study variants of the online interval coloring with bandwidth problem. In these coloring problems, the intervals are presented one by one and the online algorithm must assign each interval a color before the next interval arrives. In the classical problem, the intervals do not have bandwidth and two intersecting intervals cannot be colored by the same color. We are interested in the case where every interval has an associated bandwidth in (0,1]. This problem (standard coloring of intervals with bandwidth) was introduced by Adamy and Erlebach [1]. A set of intervals can be assigned the same color c, if for any point p on the real line, the sum of the bandwidths of intervals colored c and containing p, does not exceed 1. We refer to a coloring satisfying the above condition as a *proper coloring.*

Online coloring of intervals with bandwidth is a simultaneous generalization of two major problems. The first one is online bin packing, the study of which dates back to the works of Johnson and Ullman in the early 1970's [13, 22], see also [7] for a survey. If all the presented intervals intersect, colors correspond to bins. The second problem is the classical online interval graph coloring problem, introduced by Kierstead and Trotter [16].

As mentioned in [1], the problem of coloring intervals with bandwidth arises in many applications. Most of these applications come from the field of networks. Consider a network with a line topology that consists of links, where each link has channels of constant capacity. This can be either an all-optical WDM (wavelength-division multiplexing) network or an optical network supporting SDM (space-division multiplexing). A connection request is from one network node a to another node b, and has a bandwidth associated with it. The set of requests assigned to a channel must not exceed the capacity of the channel on any of the links on the path $[a, b]$. The goal is to minimize the number of channels (colors) used. A connection request from a to b corresponds to an interval $[a, b]$ with the respective bandwidth requirement and the goal is to minimize the number of required channels to serve all requests. Another network related application is one where requests

⋆ A preliminary version of this paper appeared in Proc. of 30th Mathematical Foundations of Computer Science (MFCS 2005), LNCS 2618, pages 295–307.

have constant duration c, and we have to serve all requests as fast as possible. With respect to the online interval coloring problem, the colors correspond to time slots, and the total number of colors corresponds to the schedule length. The last example comes from scheduling, a requested job has a duration and resource requirement during its execution. Jobs (intervals) arrive online and must be assigned to a machine (color) immediately. All the machines have the same capabilities and the objective is to minimize the number of machines used.

The unweighted (classical) problem is equivalent to coloring an interval graph, where each interval corresponds to a node and an edge between two nodes appears if the corresponding intervals intersect. Interval graphs are perfect, therefore the chromatic number of the graph is the maximum clique size [12]. In the case of interval graphs, the maximum clique size represents a point where the largest number of intervals intersect.

We study online coloring problems in terms of competitive analysis, that is, in terms of the *absolute competitive ratio* and the *asymptotic competitive ratio*. Thus we compare an online algorithm to an optimal offline algorithm OPT that knows the complete sequence of intervals in advance.

Let $B(\sigma)$ (or B, if the sequence σ is clear from the context), be the cost of algorithm B on the request sequence σ. An algorithm A is $\mathcal{R}$-competitive (with respect to the absolute competitive ratio) if for every sequence σ, $A(\sigma) \le \mathcal{R} \cdot OPT(\sigma)$. The absolute competitive ratio of an algorithm is the infimum value of $\mathcal{R}$ such that the algorithm is $\mathcal{R}$-competitive.

The asymptotic competitive ratio for an online algorithm A is defined to be

$$\mathcal{R}_A^\infty = \limsup_{n\to\infty} \sup_\sigma \left\{ \frac{A(\sigma)}{OPT(\sigma)} \middle| OPT(\sigma) = n \right\}.$$

All results given in this paper apply to both the absolute and the asymptotic competitive ratios.

Coloring of interval graphs has been intensively studied. Kierstead and Trotter [16] gave upper and lower bounds of 3 on the competitive ratio. Much research has been done analyzing the performance of the simple First Fit algorithm for the unweighted problem. Upper bounds on the competitive ratio of 40, 25.72 and 10 were given in [14, 15, 20] respectively. Chrobak and Slusarek [6] showed a lower bound close to 4.5 on the competitive ratio of First Fit. See [21] for recent developments.

The interval coloring problem with bandwidth was first posed in 2003 in [1] by Adamy and Erlebach. They presented an online algorithm with a competitive ratio of at most 195. Azar et al. [2] presented a new algorithm with a competitive ratio of 10. In [9] we studied several extensions of this problem including coloring of unit length intervals.

Motivated by the well known bin packing problem, we investigate four variants studied in the past with respect to bin packing. Namely, Open-end bin packing, Vector packing, Cardinality constrained packing and Vector packing with cardinality constraints. Open-end online bin packing (also called the Ordered open-end problem) was introduced by Yang and Leung [23]. Online vector packing was studied by Garey et al. as a scheduling problem with resource constraints [11]. This problem was studied also in [17, 10, 4]. Cardinality constrained bin packing was first studied by Krause, Shen and Schwetman [18, 19]. It was also studied in [3, 8]. The vector packing problem with cardinality constraints was mentioned in [5]. In that paper it is treated as a special case of the vector packing problem.

We make adjustments to these variants to suit the interval coloring with bandwidth problem in the following way.

Open-end interval coloring: Given a point p and color c, we remove the restriction that all intervals intersecting point p colored with c should have total bandwidth of at most 1. Instead, we require that if the last interval which received color c and intersects p is removed, then the total bandwidth of all such other intervals, is strictly less than 1. A possible application of this model is the situation where the decision on the color of a new interval does not depend on the exact value of its bandwidth, but on the current load of each color. This is consistent with our algorithms which use a partition into classes of bandwidth rather than using the exact bandwidth to classify a new interval and to assign it a color.

Interval coloring with vector constraints: Instead of one dimensional bandwidths, the intervals are associated with d-dimensional vectors. This is a generalization of the standard interval coloring with bandwidth problem. Here each interval has d distinct weights and each color has d corresponding unit capacities. An interval can receive color c if the assignment is valid according to all d components. This variant models a multiple number of available resources that each request needs and all requests must share, rather than a single resource as in the standard problem.

Cardinality constrained interval coloring: The cardinality constrained coloring, also called the k-bounded interval coloring with bandwidth problem, additionally imposes the constraint that at each point p, at most k intersecting intervals are allowed to use one color. This variant models applications where only a limited number of requests can be satisfied simultaneously, a restriction that occurs in addition to the bandwidth constraints. We assume $k > 1$, otherwise the problem is equivalent to standard online interval coloring [16].

Cardinality and vector constrained interval coloring: This is a combination of the two previous variants. Each interval is associated with a d-dimensional vector of d distinct bandwidths and each color has d corresponding capacities. Additionally at most k intersecting intervals are allowed in one color at each point. We assume $k > 1$, otherwise the problem would also reduce to standard online interval coloring.

Our Results: We present competitive online algorithms for each of the variants. We use ideas which are extensions of the algorithm in [2]. For the open-end coloring model we present an algorithm with competitive ratio of at most 12. For the cardinality constrained variant we suggest an algorithm with competitive ratio of $\min\{10 + 2 \cdot \frac{k}{k-1}, k + 3\}$, for odd k and $\min\{12, k + 3\}$, for even k.

We design a $10d$-competitive algorithm for the vector constrained model and an algorithm of competitive ratio at most $\min\{10d + 2, 3k\}$, for even k and $\min\{10d + 2\frac{k}{k-1}, 3k\}$ for odd k, for the combined model of both vector and cardinality constraints.

We also present a lower bound of $\frac{24}{7} \approx 3.428571$, an improvement of the previously known lower bound of 3.26 for standard interval coloring with bandwidth presented in [9]. The latter lower bound does not apply for most variants studied in the current paper. For cardinality constrained coloring, a simplification of that lower bound can be applied only to very large values of k. It is also not valid for the open-end model. However the lower bound of Kierstead and Trotter [16] can be used in both these models. By using intervals of bandwidth 1, as done in the construction of [16], two intersecting intervals can not receive the same color in any of our models. Therefore the best lower bound previously known for these cases is 3. Our lower bound can be easily modified and applied to all the variants considered in this paper by a simple change of parameter.

Our results are summarized in Table 1.

	Lower Bound	Upper Bound
Interval Coloring With Bandwidth	24/7	10 [2]
Open-End Coloring	24/7	12
Cardinality Constraints	24/7	$\min\{12, k+3\}$ (even k) $\min\{10+2\frac{k}{k-1}, k+3\}$ (odd k)
Vector Constraints	24/7	$10d$
Vector and Cardinality Constraints	24/7	$\min\{10d+2, 3k\}$ (even k) $\min\{10d+2\frac{k}{k-1}, 3k\}$ (odd k)

Table 1. Results obtained in this paper and previous work. The results are given in terms of competitive ratio.

2 Preliminaries

A weighted interval graph G of a set of intervals S, is a graph where each node corresponds to an interval. The weight of the node is the bandwidth of the interval in S related to it. If two intervals intersect, there is an edge between their related nodes in G. Recall that we denote the optimal coloring of the offline algorithm by OPT.

Let $\omega(G)$ or $\omega(S)$ denote the size of the maximum cardinality clique in G (ω for short), *i.e.*, ignoring the weights. Let $\omega^*(G)$ or $\omega^*(S)$ (ω^* for short) denote the largest weighted clique in G. A weighted clique is the sum of the weights of the vertices in a clique. Note that for the interval coloring problem with bandwidth we have $OPT \geq \lceil \omega^* \rceil$

Below we give the generalized presentation of the algorithm of Kierstead and Trotter [16] presented in [9]. For convenience we include the full presentation and list four relevant lemmas from [9] and their proofs.

Let $\sigma = v_1, \ldots, v_n$ be the list of vertices of G, in the order of arrival. Algorithm $KT_{l,b}$ is defined for inputs σ such that, $b(v_i) \in (0, b]$. The algorithm partitions the intervals (*i.e.* the vertices of G) into sets A_m (for integer values of m, such that $m \geq 1$). We use C_m to denote the set of colors dedicated to A_m. Every set A_m is colored using First Fit, independently of other sets. Therefore the colors have the property $C_x \cap C_y = \emptyset$ for $x \neq y$.

A *critical point*, q, in interval $v_i \in A_{m(v_i)}$, is a point where $\omega_i^*(G_{m(v_i)-1}(v_i) \cup \{v_i\}) > (m(v_i)-1) \cdot l$. Since $v_i \in A_{m(v_i)}$, there is at least one such point for every interval in $A_{m(v_i)}$.

Lemma 1. *Given an interval v_i, let $m = m(v_i)$. For the set A_m and every critical point $q \in v_i$, the total bandwidth at q of intervals in A_m does not exceed $b + l$.*

Proof. Proof by contradiction, assume that there is a critical point, $q \in v_i \in A_m$, where the size of the weighted clique of intervals in A_m is strictly larger than $b + l$. Consider all intervals, v_t, where $v_t \in A_m$ and $q \in v_t$. Let v_k be the last such interval (in the order of presentation). Clearly $i \leq k$ and after v_k is presented, the total bandwidth of A_m at point q does not increase.

Algorithm 1 $KT_{l,b}$

On a new interval v_i:

1: For every integer $m \geq 1$, let $V_m(v_i)$ and $E_m(v_i)$ be the following sub-sets of $V(G)$ and $E(G)$ respectively.
$V_m(v_i) = \{v_j \in V(G) : j < i, m(v_j) \leq m\}$;
$E_m(v_i) = \{(u,v) \in E(G) : u, v \in V_m(v_i)\}$;
Let $G_m(v_i) = G(V_m(v_i), E_m(v_i))$
$G_m(v_i) \cup \{v_i\} = G(V_m(v_i) \cup \{v_i\}, E_m(v_i) \cup \{(u, v_i) \in E(G) : u \in V_m(v_i)\})$
For an arbitrary interval graph H, let $\omega_i^*(H)$ denote the size of the maximum weighted clique among cliques containing the interval v_i in graph H

2: Let $m(v_i)$ be the smallest m such that $\omega_i^*(G_m(v_i) \cup \{v_i\}) \leq m \cdot l$.

3: $A_{m(v_i)} \leftarrow A_{m(v_i)} \cup \{v_i\}$

4: Color v_i considering only the intervals of $A_{m(v_i)}$ using First Fit on colors of $C_{m(v_i)}$.

Since $b(v_i) \leq b$, and q is a critical point of v_i, the total bandwidth at point q for intervals in $G_{m-1}(v_i)$ is greater than $(m-1) \cdot l - b$. Since $m(v_k) = m$ we have $\omega_k^*(G_m(v_k)) \leq m \cdot l$, in particular, the total bandwidth at point $q \in v_i \cap v_k$ does not exceed $m \cdot l$.

Since $\{v_j \in V(G) : j < i, m(v_j) < m, q \in v_j\} \subseteq \{v_j \in V(G) : j < k, m(v_j) < m, q \in v_j\}$, the bandwidth of intervals of $G_{m-1}(v_k)$ containing q (at the time v_k arrives) is more than $(m-1) \cdot l - b$. Also by the assumption, the total bandwidth at point q in A_m is more that $b + l$. We combine the total bandwidth at point q in $G_m(v_k) \cup \{v_k\}$ and get more than $((m-1) \cdot l - b) + (l + b) = m \cdot l$. This contradicts the fact that $m(v_k) = m$ □

Lemma 2. *For every m, $\omega^*(A_m) \leq 2(b + l)$.*

Proof. Proof by contradiction, assume that there is a weighted clique of more than $2(b + l)$ in A_m obtained at point p_j. By the previous lemma, this point is not a critical point of any interval in A_m. For every interval $v_i \in A_m$, where $p_j \in v_i$, there is a critical point either to the right of p_j or the left of p_j or both. Denote the closest critical point (of any interval of A_m which contains p_j) to the left of p_j, q_l and the closest critical point to the right of p_j, q_r. Since there is at least one critical point for every v_i, either $q_l \in v_i$ or $q_r \in v_i$ or both for every $v_i \in A_m$ (since the critical point of v_i cannot be in the interval (q_l, q_r)). But this means that either q_l or q_r has a total bandwidth of more than $b + l$. By Lemma 1, this is not possible. Note that either q_l or q_r or both must exist. If one of q_r or q_l does not exist, we get the contradiction at the point that exists. □

Lemma 3. *If all intervals have the same bandwidth, b, and l is divisible by b, for every m, $\omega^*(A_m) \leq 2l$.*

Proof. Similarly to Lemma 1, we show the following. Given an interval v_i let $m = m(v_i)$. For the set A_m and every critical point $q \in v_i$, the total bandwidth at q of intervals in A_m does not exceed l. To prove this claim we use the same notations given in the proof of Lemma 1. Unlike in the previous case, here the total bandwidth at point q for intervals in $G_{m-1}(v_i)$ is exactly $(m-1) \cdot l$, since l is divisible by b. By contradiction we assume that the total bandwidth of point q exceeds l. Therefore, the total bandwidth at point q in $G_m(v_k) \cup \{v_k\}$ is more than $(m-1) \cdot l + l = m \cdot l$. This contradicts the fact that $m(v_k) = m$. By the same argument given in Lemma 2 we get $\omega_k^*(A_m) \leq 2l$. □

Lemma 4. *(i) The largest value of m ever used in $KT_{l,b}$ is $\lceil \frac{\omega^*}{l} \rceil$*
(ii) The number of colors used by $KT_{l,b}$ is at most $\lceil \frac{\omega^*}{l} \rceil (\max_m FF(A_m))$, where $FF(A_m)$ denotes the number of colors used by First Fit on the set A_m of intervals that were presented online.

Proof. **(i):** For a maximum weighted clique of ω^* and for every interval $v_i \in \sigma$, $\omega^*(G_{\lceil \frac{\omega^*}{l} \rceil}(v_i) \cup \{v_i\}) \leq \omega^* \leq \lceil \frac{\omega^*}{l} \rceil \cdot l$ **(ii):** By (i) the largest value of m is at most $\lceil \frac{\omega^*}{l} \rceil$. For each m, A_m is colored by First Fit using the related colors of C_m (last step of the algorithm). □

Note that $KT_{1,1}$ without bandwidth is equivalent to the original algorithm of Kierstead and Trotter [16]. In their algorithm every layer can be colored by First Fit with at most 3 colors. The number of layers equals the size of the maximum cardinality clique. Therefore the number of colors used is at most $3OPT$.

3 Upper Bounds

In this section we present algorithms for different models. We denote the optimal offline algorithm for a specific variant A, by OPT_A, *e.g.*, for the open-end model we denote the optimal offline algorithm that follows the restrictions of the model by $OPT_{Open-End}$. When we write simply OPT, we refer to the minimum number of colors required to color the input if the considered variant is standard online coloring with bandwidth.

3.1 Open-End Coloring

In the Open-End version, colors can consist of intersecting intervals with a total bandwidth of more than 1. However, for any given point, the removal of the last interval colored with a specific color must bring the color's level back to strictly below 1 at that point.

Theorem 1. *There exists an online algorithm with competitive ratio of at most* 12 *for the open-end interval coloring.*

Proof. **Algorithm.** Perform an online partition of the intervals into three disjoint sub-sequences S_1, S_2, and S_3 according to the bandwidth of the intervals. The sub-sequences are defined as follows. For an interval I,

- $I \in S_1$ if $b(I) \leq \frac{1}{4}$
- $I \in S_2$ if $\frac{1}{4} < b(I) < 1$
- $I \in S_3$ if $b(I) = 1$

Each sub-sequence is colored by a different set of colors. The colors to be assigned are split into three disjoint classes C_1, C_2, and C_3. Each class is designated to intervals of one sub-sequence, *i.e.*, C_1 for S_1, C_2 for S_2 and C_3 for S_3.

The classes of colors are built dynamically, when a new color is required, the first unused color is assigned. When a color is assigned to one of the three classes, it can no longer be assigned to any of the other classes.

Run in parallel (i.e., independently) the following three sub-algorithms:

Sub-Algorithm A_{S_1}. Use $KT_{\frac{1}{4},\frac{1}{4}}$ on the intervals of S_1 ignoring the open-end option.

Sub-Algorithm A_{S_2}. Use a variant of $KT_{1,1}$ without bandwidth on S_2. In lines 1-3 of the algorithm $KT_{1,1}$, treat all intervals as if they have bandwidth of exactly 1. The change is made in line 5 of the algorithm. Instead of using at most 3 colors for each A_m, use only one color.

Sub-Algorithm A_{S_3}. Use the Algorithm of Kierstead and Trotter, *i.e.*, $KT_{1,1}$ without bandwidth.

Lemma 5. *(i) A_{S_1} uses at most $5 \cdot OPT_{Open-End}(S_1)$ colors;*
(ii) A_{S_2} uses at most $4 \cdot OPT_{Open-End}(S_2)$ colors;
(iii) A_{S_3} uses at most $3 \cdot OPT_{Open-End}(S_3)$ colors;

Proof. **Sub-Algorithm A_{S_1}.** According to Lemma 4 part (i), the number of colors used by $KT_{\frac{1}{4},\frac{1}{4}}$ is at most $\lceil \frac{\omega^*}{\frac{1}{4}} \rceil \max_m FF(A_m)$, where $FF(A_m)$ denotes the number of colors used by First Fit on the set A_m of intervals that were presented online. By Lemma 2, for every m, $\omega^*(A_m) \leq 1$. Therefore $\max_m FF(A_m) = 1$ and we get that the number of colors used is at most $\lceil 4\omega^* \rceil$. In the open-end version, the total bandwidth for each color may exceed 1. Since all the intervals in S_1 have a maximum bandwidth of $\frac{1}{4}$, $OPT_{Open-End}$ can use each color for a total bandwidth of at most $\frac{5}{4}$. Therefore any algorithm needs at least $\lceil \frac{4}{5}\omega^* \rceil$ colors. Hence, we get, $A_{S_1}(S_1) \leq \lceil 4\omega^* \rceil = \lceil 5 \cdot \frac{4}{5}\omega^* \rceil \leq 5 \cdot \lceil \frac{4}{5}\omega^* \rceil \leq 5OPT_{Open-End}(S_1)$.

Sub-Algorithm A_{S_2}. Note that in this variant of $KT_{1,1}$ without bandwidth, all the sets of intervals A_m should contain the same intervals as if we had used the regular $KT_{1,1}$ without bandwidth. The only difference is the coloring of the intervals within these sets.

First we claim that this variant results in a proper coloring. By Lemma 3, the cardinality clique is at most 2 in each A_m. In the Open-End variant, two intersecting intervals each of bandwidth strictly less than 1 can be colored by the same color. Since every interval $I \in S_2$ satisfies $\frac{1}{4} < b(I) < 1$ every A_m can be colored by a single color, and the claim is proved.

Next we show that the number of colors used by $A_{S_2}(S_2)$ is at most $4 \cdot OPT_{Open-End}(S_2)$. Algorithm A_{S_2} uses at most ω colors, where ω is the largest cardinality clique and not the largest weight clique. Since for every $I \in S_2$, $b(I) > \frac{1}{4}$, $OPT_{Open-End}(S_2)$ can use at most four intersecting intervals in a single color. Therefore $OPT_{Open-End}(S_2)$ uses at least $\frac{\omega}{4}$ colors. Thus we get that A_{S_2} uses at most $4 \cdot OPT_{Open-End}(S_2)$ colors.

Sub-Algorithm A_{S_3}. According to the analysis of the algorithm of Kierstead and Trotter, we have the following bound on the number of colors used, $A_{S_3}(S_3) \leq 3OPT(S_3)$. Since for every $I \in S_3$, $b(I) = 1$, every two intersecting intervals cannot receive the same color. Thus $OPT(S_3) = OPT_{Open-End}(S_3)$.

□

Since the algorithms run obliviously of each other, we need to sum their competitive ratios. By combining the competitive ratios of the sub-algorithms of A_{S_1}, A_{S_2} and A_{S_3}, we get a competitive ratio of 12 for the complete algorithm.

□

3.2 Coloring with Vector Constraints

In the vector constrained variant, each interval has a vector of d distinct weights and each color has d corresponding unit capacities. Given a point p, the total bandwidth of each of the d components in intervals intersecting p, which receive the same color must not exceed the unit capacity of the corresponding component of their color. This constraint must hold for each of the d components simultaneously. The d-dimensional coloring problem we study here is a generalization of the problem of interval coloring with bandwidth, the latter problem can be viewed as a 1-dimensional vector constrained problem.

Denote by $b_i(I)$, the bandwidth of the i'th coordinate of the vector of bandwidths of interval I

Theorem 2. *There exists an online algorithm for vector constrained interval coloring with competitive ratio of at most* $10d$*, where* d *is the dimension of the vector constraints.*

Proof. Perform an online partition of the intervals into d disjoint sub-sequences $S_1, \ldots, S_d$ according to the bandwidth vector of the intervals. The sub-sequences are defined as follows. For every interval I, $I \in S_i$ if $1 \leq i \leq d$ is the smallest number satisfying that $b_i(I) \geq b_j(I)$ for every $j \neq i$.

Each sub-sequence is colored by a different set of colors. The colors to be assigned are split into d disjoint classes $C_1, \ldots, C_d$. Each class is designated to color intervals of one sub-sequence, *i.e.*, C_1 for S_1, C_2 for S_2 and so on.

For every i run an instance of the algorithm of [2] on S_i using the colors of C_i. Any interval $I \in S_i$ is treated by this algorithm as an interval of the one-dimensional bandwidth $b_i(I)$. From the definition of S_i, $b_i(I) \geq b_j(I)$ for every $j \neq i$, therefore the resulting coloring is proper. The algorithm of [2] can be summarized by the general presentation of Kierstead and Trotter given in the preliminaries as follows.

For an interval I,

- $I \in B_1$ if $b(I) \leq \frac{1}{4}$
- $I \in B_2$ if $\frac{1}{4} < b(I) \leq \frac{1}{2}$
- $I \in B_3$ if $\frac{1}{2} < b(I) < 1$

Algorithm of [2].

- Use $KT_{\frac{1}{4},\frac{1}{4}}$ (taking bandwidth into account) on the intervals of B_1.
- Use a variant of $KT_{1,1}$ without bandwidth on B_2. In lines 1-3 of the algorithm $KT_{1,1}$, treat all intervals as if they have bandwidth of exactly 1. The change is made in line 4 of the algorithm. For each A_m use only one color.
- Use $KT_{1,1}$ without bandwidth on B_3.

This algorithm has competitive ratios of 4, 3 and 3 for the classes S_1, S_2 and S_3, respectively. Since we run in parallel d instances of the algorithm of [2], each of which has a competitive ratio of at most 10, we get the combined competitive ratio of at most $10d$. □

3.3 Coloring with Cardinality Constraints

In the cardinality constrained, or the k-bounded interval coloring with bandwidth problem there is an additional restriction. In this variant, for each point p and color c, at most k intersecting intervals colored using color c and intersecting point p may exist.

Note that the cardinality constraint can be expressed by giving vector constraints of dimension $d = 2$. The first component can be defined to be the bandwidth requirement, whereas the second component would be defined to be $\frac{1}{k}$ for all intervals. This would result in an algorithm of competitive ratio of at most 20 using the algorithm of [2]. A slightly more careful analysis results in a competitive ratio of 14 for the same algorithm, since all intervals which would be colored according to their second component would be assigned to the same subclass of intervals. We design an improved algorithm with competitive ratio as follows.

Theorem 3. *There exists an online algorithm for cardinality constrained interval coloring with a competitive ratio of at most* $\min\{10 + 2 \cdot \frac{k}{k-1}, k+3\}$ *for odd* k*, and of at most* $\min\{12, k+3\}$ *for even* k*, where* k *is the cardinality constraint.*

Proof. **Algorithm.** If $\min\{10 + 2 \cdot \frac{k}{k-1}, k+3\} = 10 + 2 \cdot \frac{k}{k-1}$ for odd k or $\min\{12, k+3\} = 12$ for even k, use the algorithm described in case 1. Otherwise use case 2.

Case 1. Perform an online partition of the intervals into two disjoint sub-sequences S_1 and S_2 according to the bandwidth of the intervals. The sub-sequences are defined as follows.
For an interval I,
- $I \in S_1$ if $b(I) \leq \frac{1}{k}$
- $I \in S_2$ if $\frac{1}{k} < b(I) < 1$

Sub-Algorithm A_{S_1}

Even k. Take the bandwidth of every interval in S_1 to be exactly $\frac{1}{k}$ and use algorithm $KT_{\frac{1}{2},\frac{1}{k}}$.

Odd k. Take the bandwidth of every interval in S_1 to be exactly $\frac{1}{k-1}$ and use algorithm $KT_{\frac{1}{2},\frac{1}{k-1}}$, with the following change. The sets A_1 and A_2 are considered together in the application of First Fit in line 4.

Sub-Algorithm A_{S_2} Use the algorithm presented in [2] on S_2. See details of this algorithm in the proof of Theorem 2).

Case 2. Perform an online partition of the intervals into two disjoint sub-sequences R_1 and R_2 according to the bandwidth of the intervals. The sub-sequences are defined as follows.
For an interval I,
- $I \in R_1$ if $b(I) \leq \frac{1}{2}$
- $I \in R_2$ if $\frac{1}{2} < b(I) < 1$

Sub-Algorithm A_{R_1}**.** Use a variant of $KT_{1,1}$ without bandwidth on R_1. In lines 1-3 of the algorithm $KT_{1,1}$, treat all intervals as if they have bandwidth of exactly 1. The change is made in line 4 of the algorithm. Instead of using at most 3 colors for each A_m, use only one color.

Sub-Algorithm A_{R_2}**.** Use $KT_{1,1}$ without bandwidth, treating every interval as if its bandwidth is exactly 1.

We next analyze the competitive ratio.

Case 1.

Sub-Algorithm A_{S_1}

Even k**.** Since each color can be used for a total of k intersecting intervals, we can treat all intervals in S_1 as if they have bandwidth of exactly $\frac{1}{k}$. The value ω^* is computed using this assumption.

The number of colors used by $KT_{\frac{1}{2},\frac{1}{k}}$ is at most $\lceil \frac{\omega^*}{\frac{1}{2}} \rceil \max_m FF(A_m)$, according to Lemma 4 part (ii). Since now all intervals have the same bandwidth $\frac{1}{k}$, and since $\frac{1}{2}$ is divisible by $\frac{1}{k}$, by Lemma 3, we get that for every m, $\omega^*(A_m) \le 1$. Therefore $\max_m FF(A_m) = 1$ and we get that the number of colors used by our algorithm is at most $\lceil 2\omega^* \rceil \le 2\lceil \omega^* \rceil$.

Odd k. In this case, the value ω^* is computed based on the assumption that every interval has bandwidth $\frac{1}{k-1}$.

Similarly to the previous case, we would get a competitive ratio of 2 if an optimal algorithm could use only $k-1$ intersecting intervals for every color. However $OPT_{k-Bounded}$ can use k intersecting intervals for each color. Therefore the $OPT_{k-Bounded}$ uses at least $\frac{k-1}{k}\omega^*$ colors. If the algorithm uses a single color, then the competitive ratio is 1, since $OPT_{k-Bounded} \ge 1$, for an input with at least one interval. Otherwise, the sets A_1, A_2 are colored together using First Fit, and since the size of the largest weighted clique in $A_1 \cup A_2$ is at most 1, a single color is used for $A_1 \cup A_2$. Therefore, the algorithm uses $\lceil \omega^* \rceil - 1 \le \omega^*$ colors. Hence we get a competitive ratio of $2 \cdot \frac{k}{k-1}$.

Sub-Algorithm A_{S_2}. The algorithm presented in [2] has a competitive ratio of at most 10 on intervals in (0,1]. Since for every $I \in S_2$, $b(I)$ in (0,1], the competitive ratio for this part is also at most 10.

Combining the competitive ratio of this case we get at most $10 + \frac{2k}{k-1}$ for odd k and at most 12 for even k.

Case 2.

Sub-Algorithm A_{R_1}. Note that in this variant of $KT_{1,1}$ without bandwidth, all the sets of intervals A_m contain the same intervals as if we had used the regular $KT_{1,1}$ without bandwidth. The only difference is the coloring of the intervals within these sets.

First we claim that this variant results in a proper coloring. By Lemma 3, the cardinality clique is at most 2 in each A_m. Since every interval $I \in R_1$ satisfies $b(I) \le \frac{1}{2}$ every A_m can be colored by a single color and the claim is proved.

We next show that the number of colors used by $A_{R_1}(R_1)$ is at most $k \cdot OPT_{k-Bounded}(R_1)$. Algorithm A_{R_1} uses at most ω colors, where ω is the cardinality clique of the set R_1 and not the weighted clique. In the cardinality constrained variant $OPT_{k-Bounded}$ can only color k intersecting intervals with the same color. Therefore it uses at least $\frac{\omega}{k}$ colors. Thus we get that $A_{R_1}(R_1)$ uses at most $k \cdot OPT_{k-Bounded}(R_1)$ colors.

Sub-Algorithm A_{R_2}. We have $A_{R_2}(R_2) \le 3OPT(R_2)$ and $OPT(R_2) = OPT_{k-Bounded}(R_2)$. The last equality is valid since for every $I \in R_2$, $b(I) > \frac{1}{2}$, so no two intersecting intervals can receive the same color.

Combining the competitive ratio of this case we get $k+3$.

To complete the analysis, since k is known in advance, the algorithm uses the best option for a specified k, thus getting the minimum competitive ratio out of the two cases. □

3.4 Coloring with Vector and Cardinality Constraints

This case is a combination of the previous two variants. There is a d-dimensional vector for each interval with d distinct bandwidths and each color has d corresponding capacities. Additionally at most k intersecting interval are allowed in one color as required in the cardinality constraint. Note that the cardinality constraint can be expressed using an additional ($(d+1)$-th)component of the vector constraint, which is defined to be $\frac{1}{k}$ for all intervals. This would result in an algorithm of competitive ratio of at most $10(d+1)$ using the algorithm of Section 3.2. A slightly more careful analysis results in a competitive ratio of $10d+4$ for the same algorithm, since all intervals which would be colored according to their last component would be assigned to the same subclass of intervals. We design an improved algorithm with competitive ratio as follows.

Theorem 4. *There exists an online algorithm for interval coloring with vector and cardinality constraints with a competitive ratio of at most* $\min\{10d+2, 3k\}$ *for even* k *and of at most* $\min\{10d+2\frac{k}{k-1}, 3k\}$ *for odd* k*, where* k *is the cardinality constraint and* d *is the dimension of the vector constraints.*

Proof. **Algorithm.** If $\min\{10d+2, 3k\} = 10d+2$ (for even k) or $\min\{10d+2\frac{k}{k-1}, 3k\} = 10d+2\frac{k}{k-1}$ (for odd k) use the algorithm described in case 1. Otherwise use case 2.

Case 1. The algorithm is similar to the algorithm presented for interval coloring with vector constraints. We introduce a new sub-sequence S_0. Interval $I \in S_0$ if for every $1 \leq i \leq d$, $b_i(I) \leq \frac{1}{k}$. The other sub-sequences are defined as follows.
For every interval I, $I \in S_i$ if $1 \leq i \leq d$ is the smallest number satisfying $b_i(I) \geq b_j(I)$ for every $j \neq i$ and $b_i(I) > \frac{1}{k}$.
For every $i \geq 1$ run an instance of the algorithm of [2] on S_i using the colors of C_i (for the details of this algorithm see the proof of Theorem 2).

Even k. Take the bandwidth of every coordinate in the bandwidth vector of the intervals in S_0 to be exactly $\frac{1}{k}$ and use algorithm $KT_{\frac{1}{2},\frac{1}{k}}$.
Odd k. Take the bandwidth of every coordinate in the bandwidth vector of the intervals in S_0 to be exactly $\frac{1}{k-1}$ and use algorithm $KT_{\frac{1}{2},\frac{1}{k-1}}$, with the change that the sets A_1 and A_2 are considered together in the application of First Fit in line 4.

Case 2. Take the bandwidth of every coordinate in the bandwidth vector of all the intervals to be exactly 1 and use algorithm $KT_{1,1}$ without bandwidth.

Analysis of the competitive ratio.
Denote by $OPT_{VP-k-Bounded}$ an optimal offline algorithm for vector packing with cardinality constraints.

Case 1. The algorithm presented in [2] has a competitive ratio of 10 on intervals with bandwidth in (0,1]. Since for every $I \in S_i$, $b_i(I) \in (0,1]$, the number of colors used for each S_i, where $i \geq 1$ is at most $10OPT_{VP-k-Bounded}(S_i)$. Also since for every $j \neq i$, $b_i(I) \geq b_j(I)$, the coloring is proper.
The coloring of S_0 is similar to the coloring of the sub-algorithm A_{S_1} presented for bounded cardinality. Since all the bandwidths in the vector are at most $\frac{1}{k}$, the coloring of S_0 is proper. The number of colors used for odd k is at most $2 \cdot \frac{k}{k-1} OPT_{VP-k-Bounded}(S_0)$. For even values of k, the number of colors is at most $2OPT_{VP-k-Bounded}(S_0)$. Combining both parts, we get a competitive ratio of $10d+2$ for even values of k and $10d+2\frac{k}{k-1}$ for odd values of k.

Case 2. In this case we simply apply the algorithm of Kierstead and Trotter ignoring the bandwidth vector. Since $OPT_{VP-k-Bounded}$ can color at most k intersecting intervals with the same color, we lose an additional factor of k. Therefore, the competitive ratio of this case is $3k$.

□

4 Lower Bound

In this section we present a lower bound of $\frac{24}{7} \approx 3.428571$ on the competitive ratio of any algorithm for interval coloring with bandwidth. This is an improvement of the previously known lower bound of 3.26 for standard interval coloring with bandwidth presented in [9]. Note that the lower bound of [9] does not apply for most variants studied in this paper. For cardinality constrained coloring, a simplification of that construction can be applied only to very large values of k. It is not valid for the open-end model either. However the lower bound of Kierstead and Trotter [16] can be used in both these models. By using intervals of bandwidth 1, as done in the construction of [16], two intersecting intervals can not receive the same color in any of our models. Therefore the best lower bound previously known for these cases is 3.

We prove the following theorem, using a single type of construction for all models. The construction uses parameters, and the lower bound for each model separately can be achieved by fixing the parameters appropriately. The theorem holds for all models studied in this paper. The variant with vector constraints is a generalization of standard coloring with bandwidth, where the vector has dimension 1. Additional dimensions can be added trivially by adding zero components. Similarly, the variant with both vector and cardinality constraints is a generalization of cardinality constrained coloring.

Theorem 5. *Any deterministic online algorithm for interval coloring with bandwidth in the standard model, open-end model, and cardinality constrained model, has competitive ratio of at least* $\frac{24}{7} \approx 3.428571$.

Proof. The general structure of the input sequence is as follows. In the first part of the construction, all intervals have bandwidth α and in the second (optional) part all intervals have bandwidth $\beta > \alpha$. The values of α and β are picked depending on the exact problem. The choice is such that it is possible to assign the same color to two intersecting intervals of bandwidth α, or even to one interval of bandwidth α and one of bandwidth β, which are intersecting. However, we need to make sure that it is impossible to assign the same color to two intersecting intervals of bandwidth β, or to any three intersecting intervals of bandwidth at least α.

Such choices can be e.g. $\alpha = 0.4$ and $\beta = 0.6$ for the standard problem, or to the cardinality constrained problem, for any $k \geq 2$ (since the case $k = 1$ is equivalent to standard interval coloring, it is impossible to improve the lower bound of 3 in this case). For the open-end problem, we can take $\alpha = 0.6$ and $\beta = 1$.

Given an integer value s, the first part of the sequence is built so that the largest clique size (ignoring the bandwidth) is $2s$. Since interval graphs are perfect, it is clearly possible to color the graph using $2s$ colors, so that no two intervals colored with the same color may intersect. We next show that it is possible to color the input using s colors, as follows. First, we distribute $2s$ colors so that no two intersecting intervals receive the same color. Then we can partition colors into pairs, and unite every two colors into one. This can be done since at every point there will be at most two intersecting intervals of bandwidth α.

The second part of the sequence is built in a way that the largest clique size (again, ignoring the bandwidth) of intervals introduced in this part is $2s$. The complete sequence can be colored using $2s$ colors, similarly to the explanation above, by coloring each part of the sequence using $2s$ colors. The same palette of $2s$ colors can be used for both parts.

Consider a sub-set of the input, usually this is a sub-set of input intervals contained in some mega-interval (a continuous part of the real line). A color which was used for at least one interval in the sub-set is called a "used color". Next, we define the notion of "full colors" and "partial colors" in a coloring of this sub-set as follows. If there exists an interval colored with color c, from the second part of the input in the sub-set, *i.e.*, there exists an interval with bandwidth β colored with color c, then c is a full color. Moreover, if there exists a point p and a color c such that two distinct intervals X and Y from the first part of the input, such that $p \in X$ and $p \in Y$, both received the color c, then c is a full color. Otherwise, if this does not hold, but c is a used color, then it is a partial color. This happens if there exists at least one interval in the sub-set that is colored with c, and all intervals in the sub-set that are colored with c are independent, that is, no two of them overlap.

The construction of the two parts of the sequence are adaptations of the lower bound in [16]. The first part of the sequence uses intervals of bandwidth α and therefore two intersecting intervals may receive the same color. This is a main difference with the proof in [16], since we need to deal with such a situation, whereas in [16] all intervals have bandwidth 1. Another difference, that we already used in the construction in [9] is the assumption that some information on the optimal cost (which is either s or $2s$ in our case) is known in advance.

The construction of each part works in phases, after a phase we shrink some parts of the line into single points. Consider a point p, that is a result of shrinking an interval $[a, b]$. Every interval presented in the past which is contained in $[a, b]$ is also shrunk into p and therefore such a point inherits a list of used (partial and full) colors that some interval received. A partial color can be used again exactly once in some interval containing p. A full color cannot be assigned to any interval that contains the point p. This is done for simplification. In practice it means that for a given point p that is the result of shrinking, every future interval either contains this point or not, *i.e.*, it either contains all intervals that were shrunk into this point, or has no overlap with any of them.

We would like to show that either the number of colors used in the first part is at least $\frac{24s-2}{7}$, or the number of colors used after the second step is at least $\frac{48s-4}{7}$. This would imply the lower bound. Therefore, the sequence construction can clearly stop once $7s$ colors have been used. Therefore we may assume that we are initially given a palette of $7s$ colors, $1, \ldots, 7s$, that may be used by the algorithm. The ith color ever used by the algorithm is called color number i. As soon as color $7s$ is used, the proof is complete. This is just one stopping condition. We may stop the sequence earlier as well, as will be discussed later.

The first part of the sequence has intervals of bandwidth α and starts with introducing $S(0)$ non-intersecting intervals, this is phase 1. A bound on the value $S(0)$ is fixed later.

Since the algorithm is using at most $7s$ colors, this means that there exists a set of $\frac{S(0)}{7s}$ intervals that share the exact same color c. We shrink all intervals into single points. Later phases result in additional points. Since there are no intersecting intervals, color c is partial in all points colored with it.

We now define phase $i \geq 2$. The phases are constructed in a way that in the beginning of phase i there is a set of at least $S(i-1)$ points that contain two given sub-sets of the $7s$ colors. The first sub-set is of $P(i-1)$ partial colors and the second is of $F(i-1)$ full colors. These points are called points of interest. Note that after phase 1 we have $P(1) = 1$ and $F(1) = 0$.

There exist some other points containing other sub-sets of full and partial colors. All these points are called void points. At this time, we partition the points of interest into consecutive sets of four. At most three points of interest that do not participate in this partitioning become void points.

We next define additional intervals, increasing the size of the largest cardinality clique (with respect to the number of intervals, *i.e.*, ignoring bandwidth) by exactly one. Given a set of four points listed from left to right a_1, a_2, a_3, a_4, let b be the leftmost void point on the right hand side of a_1, between a_1 and a_2. If no such point exists, then let $b = \frac{a_1+a_2}{2}$, *i.e.*, the point which is halfway between a_1 and a_2. Similarly, let d be the rightmost void point between a_3 and a_4, and if no such point exists then $d = \frac{a_3+a_4}{2}$. Let f be a point between a_2 and a_3 that is not a void point. We introduce the intervals $I_1 = [a_1, \frac{a_1+b}{2}]$ and $I_2 = [\frac{d+a_4}{2}, a_4]$.

If they both receive the same color (used or unused at points a_1 and a_4), we introduce the intervals $I_3 = [\frac{a_1+b}{2}, f]$ and $I_4 = [f, \frac{d+a_4}{2}]$. The interval I_3 intersects with a_2, and with I_1. The second interval I_4 intersects I_3, a_3 and I_2. We consider the colors used for the four new intervals. If at most two distinct colors were used, then there exists a point in the range $[a_1, a_4]$ where two intersecting intervals received the same color, and therefore there is at least one new full color in this interval. If a color that is partial in the point a_1, a_2, a_3, a_4 was used, then this color becomes full in $[a_1, a_4]$. If three unused colors were used, then these colors become additional partial colors in $[a_1, a_4]$.

If I_1, I_2 receive distinct colors (used or unused), we introduce the interval $I_5 = [\frac{a_1+b}{2}, \frac{d+a_4}{2}]$, instead of presenting intervals I_3 and I_4 as was done in the previous case. Interval I_5 intersects with I_1, I_2, a_2, a_3. We consider the colors used for the three new intervals. If I_5 gets the same color as I_1 or I_2, then this color becomes full in $[a_1, a_4]$. If a color which is partial in the point a_1, a_2, a_3, a_4 was used, then this color becomes full in $[a_1, a_4]$. If three unused colors were used, then these colors become additional partial colors in $[a_1, a_4]$.

We shrink every such interval $[a_1, a_4]$ into a single point. Each of the new shrunk points received either three new partial colors, or one full (not necessarily new) color.

Note that we do not use more than $7s$ colors, and each new shrunk point receives a number of full and partial colors, which is at most three colors in total. Four intervals are introduced only if the first two received the same color. If the point has no new full colors, then it has exactly three new partial colors. Otherwise, it has at least one new full color, and possibly one or two new partial or full colors. Before the phase, all points of intervals had the exact same sub-sets of partial and full colors. This gives seven options for the type of new colors (or colors which changed status from partial to full). Let "f" denote full and "p" denote partial, then the options are $(p,p,p), (f,p,p), (f,f,p), (f,f,f), (f,p), (f,f), (f)$. There are less than $(7s)^3$ options for each type, and thus in total, there are less then $7 \cdot (7s)^3$ choices for the updated sub-sets given the previous ones. We can choose at least $S(i) = \frac{S(i-1)}{4 \cdot 7 \cdot (7s)^3}$ points having the same sets of full and partial colors. Note that in this calculation, we can neglect the (at most three) points of interest that may have become void. This is the case where the number of points of interest is not divisible by 4. The values $S(i-1)$ are chosen such that all of them are divisible by 4. Therefore, if any points of interest become void, it means that the number of points of interest is actually larger than $S(i-1)$, and does not get reduced below $S(i-1)$ by omitting these at most three points.

The points containing these exact sets of colors become the points of interest of the next phase, and the others become void points of the next phase. Points that are void points of previous phases and are not contained in shrunk intervals remain void points. Note that the only points where the new intervals intersect are points with no previous intervals, and therefore the clique size increases by exactly 1.

After the first $2s$ phases, the sequence may continue with the second part. If $P(2s) + F(2s) \geq \frac{24s-2}{7}$ the sequence stops since the lower bound is obtained. Otherwise, the second part goes on for $2s$ phases, however the intervals have bandwidth β, therefore no new partial colors are introduced, and every phase results in three new full colors. To verify this, it can be checked that in both construction cases, the new intervals must receive three distinct colors, that are either unused or partial. The number of full colors after all phases is at least $F(2s) + 6s$. Let A be the number of phases among the first $2s$ which increased the number of partial colors by 3. Therefore $\frac{24s-2}{7} > P(2s) + F(2s) \geq 3A + 1$ (since in the first phase exactly one color is introduced, which must be partial). In all other phases except the first one and up to phase $2s$, the number of full colors increased. Therefore $F(2s) \geq 2s - 1 - A$. We get $2s - 1 - F(2s) \leq A < \frac{24s-9}{21}$ and therefore $F(2s) > \frac{6s-4}{7}$ and $F(2s) + 6s > \frac{48s-4}{7}$, which proves the lower bound in this case.

Note that in each phase, the number of intervals which can be used for the next phase decreases by a factor of at most $28 \cdot (7s)^3$. To complete the construction, we need $S(4s) \geq 1$. If the initial number of intervals introduced is $S(0) = (28 \cdot (7s)^3)^{4s}$, this holds and we are done. □

5 Conclusion

We designed competitive algorithms for several variants of interval coloring. Our lower bound on the competitive ratio holds for all these models. For the vector constrained model, our upper bound is linear in d whereas our lower bound is constant. It would be interesting to improve the lower bound so that it depends on d. Note that the same problem is known to be open for the online vector packing problem as well (see [7]).

References

1. U. Adamy and T. Erlebach. Online coloring of intervals with bandwidth. In *Proc. of te First International Workshop on Approximation and Online Algorithms (WAOA2003)*, pages 1–12, 2003.
2. Y. Azar, A. Fiat, M. Levy, and N.S. Narayanaswamy. An improved algorithm for online coloring of intervals with bandwidth. *Theoretical Computer Science*, 363(1):18–27, 2006.
3. L. Babel, B. Chen, H. Kellerer, and V. Kotov. Algorithms for on-line bin-packing problems with cardinality constraints. *Discrete Applied Mathematics*, 143(1-3):238–251, 2004.
4. D. Blitz, A. van Vliet, and G. J. Woeginger. Lower bounds on the asymptotic worst-case ratio of online bin packing algorithms. Unpublished manuscript, 1996.
5. A. Caprara, H. Kellerer, and U. Pferschy. Approximation schemes for ordered vector packing problems. *Naval Research Logistics*, 92:58–69, 2003.

6. M. Chrobak and M. Ślusarek. On some packing problems relating to dynamical storage allocation. *RAIRO Journal on Information Theory and Applications*, 22:487–499, 1988.
7. J. Csirik and G. J. Woeginger. On-line packing and covering problems. In *A. Fiat and G. J. Woeginger, editors,* Online Algorithms: The State of the Art, pages 147–177, 1998.
8. L. Epstein. Online bin packing with cardinality constraints. *SIAM Journal on Discrete Mathematics*, 20(4):1015–1030, 2006.
9. L. Epstein and M. Levy. Online interval coloring and variants. In *Proc. of The 32nd International Colloquium on Automata, Languages and Programming (ICALP2005)*, pages 602–613, 2005.
10. G. Galambos, H. Kellerer, and G. J. Woeginger. A lower bound for online vector packing algorithms. *Acta Cybernetica*, 10:23–34, 1994.
11. M. R. Garey, R. L. Graham, D. S. Johnson, and A. C. C. Yao. Resource constrained scheduling as generalized bin packing. *Journal of Combinatorial Theory (Series A)*, 21:257–298, 1976.
12. T. R. Jensen and B. Toft. *Graph coloring problems.* Wiley, 1995.
13. D. S. Johnson. *Near-optimal bin packing algorithms.* PhD thesis, MIT, Cambridge, MA, 1973.
14. H. A. Kierstead. The linearity of first-fit coloring of interval graphs. *SIAM Journal on Discrete Mathematics*, 1(4):526–530, 1988.
15. H. A. Kierstead and J. Qin. Coloring interval graphs with First-Fit. *SIAM Journal on Discrete Mathematics*, 8:47–57, 1995.
16. H. A. Kierstead and W. T. Trotter. An extremal problem in recursive combinatorics. *Congressus Numerantium*, 33:143–153, 1981.
17. L. T. Kou and G. Markowsky. Multidimensional bin packing algorithms. *IBM Journal on Research and Development*, 21:443–448, 1977.
18. K. L. Krause, V. Y. Shen, and H. D. Schwetman. Analysis of several task-scheduling algorithms for a model of multiprogramming computer systems. *Journal of the ACM*, 22(4):522–550, 1975.
19. K. L. Krause, V. Y. Shen, and H. D. Schwetman. Errata: "Analysis of several task-scheduling algorithms for a model of multiprogramming computer systems". *Journal of the ACM*, 24(3):527–527, 1977.
20. S. Pemmaraju, R. Raman, and K. Varadarajan. Buffer minimization using max-coloring. In *Proc. of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2004)*, pages 562–571, 2004.
21. W. T. Trotter. Current research problems: First Fit colorings of interval graphs. http://www.math.gatech.edu/~trotter/trotter.rprob.html.
22. J. D. Ullman. The performance of a memory allocation algorithm. Technical Report 100, Princeton University, Princeton, NJ, 1971.
23. J. Yang and J. Y.-T. Leung . The ordered open-end bin-packing problem. *Oper. Res.*, 51(5):759–770, 2003.