Online Interval Coloring and Variants*

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 interval coloring problems and present new upper and lower bounds for several variants. We are interested in four problems, online coloring of intervals with and without bandwidth and a new problem called lazy online coloring again with and without bandwidth. We consider both general interval graphs and unit interval graphs. Specifically, we establish the difference between the two main problems which are interval coloring with and without bandwidth. We present the first non-trivial lower bound of **3.2609** for the problem with bandwidth. This improves the lower bound of 3 that follows from the tight results for interval coloring without bandwidth presented in [9].

1 Introduction

We study online interval coloring problems. In the basic problem intervals are presented one by one and the online algorithm must assign each interval a color before the next interval arrives, where two intersecting intervals can not be colored by the same color. We are also interested in the case where every interval has an associated bandwidth in (0,1], this problem was first 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.*

As mentioned in [1], the interval coloring problem with bandwidth arises in many applications. Most of the applications come from networks field. 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 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 that if the requests have constant duration c, and we have to serve all requests as fast as possible. With respect to our online interval coloring problem with bandwidth, 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)

* A preliminary version of this paper appeared in Proc. of the 32nd International Colloquium on Automata, Languages and Programming (ICALP2005).
** Research supported by Israel Science Foundation (grant no. 250/01).

immediately. All the machines have the same capabilities and the objective is to minimize the number of machines used.

The unweighted (basic) problem is equivalent to coloring an interval graph, where each node corresponds to an interval 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 [6], which represents a point where the most intervals intersect. It can be elaborated for the bandwidth case, if we refer to the maximum clique size as the maximum weighted clique. Each node has the weight of the related interval ,*i.e.*, its bandwidth, and the clique size is the sum of weights of the clique.

We study online coloring problems in terms of competitive analysis. Thus we compare an online algorithm to an optimal offline algorithm OPT that knows the complete sequence of intervals in advance. In this paper, we make use of two types of competitive ratios. The *absolute competitive ratio* and the *asymptotic competitive ratio.*

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, $A(\sigma) \leq \mathcal{R} \cdot OPT(\sigma)$. 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

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

Most results in this paper hold for both definitions of competitive ratio. We mention the cases where a result holds for only one definition.

Coloring interval graphs has been intensively studied, Kierstead and Trotter [9] constructed an online algorithm which uses at most $3\omega - 2$ colors where ω is the maximum clique size of the interval graph. They also presented a matching lower bound of $3\omega - 2$ on the number of colors in a coloring of an arbitrary online algorithm. As mentioned above, the chromatic number of interval graphs equals to the size of a maximum clique. This means that the optimal offline algorithm can color every interval graph with ω colors.

Much research has been done analyzing the performance of the simple First Fit algorithm. An upper bound of 40 on the competitive ratio was proven in [7], and later on an upper bound of 25.72 was presented by Kierstead and Qin [8]. In a recent study [11], a competitive ratio of 10 was proved. Chrobak and Slusarek [3] showed a lower bound close to 4.5 on the competitive ratio of First Fit.

Coloring intervals with bandwidth was first posed in 2003 in [1], they presented an online algorithm with a competitive ratio of 195. In [10] Narayanaswamy presented a new algorithm with a competitive ratio of 10.

Motivated by The Maximum Resource Bin Packing Problem [2], we introduce a new problem called Lazy Interval Coloring. As opposed to the regular interval coloring problem, we seek to use as many colors as possible. If all bandwidths are 1, a new color may be used for an interval, only if it intersects intervals with all the previously used colors. We also consider the bandwidth case, where intervals have bandwidth in (0,1]. For both problems, a newly presented interval must be colored by a used color if a proper coloring can be achieved. An application for this problem can be any of the applications mentioned before, where using additional colors (channels/time slots/machines) can improve the quality of service. The scheduler (who assigns colors) has the purpose of using as many colors as possible. However, to avoid the usage of too many colors, the instructions of a scheduler (given by the boss, who pays for the equipment) are not to use a new color unless it is absolutely necessary.

Our Results: We show that introducing bandwidth to interval coloring makes the problem harder. We present the first non-trivial lower bound of **3.2609**, which improves the lower bound of 3, proved in [9], for interval coloring without bandwidth. Recall that for the problem without bandwidth, the bound 3 is tight. We also show that bandwidth makes lazy interval coloring a much harder problem.

In this work we consider resource augmentation for the interval coloring problem with and without bandwidth. We show that there exists an online algorithm that when uses twice as much resource for each color, can perform as well as the optimal offline algorithm. We also present a matching lower bound.

For the bandwidth case we study two interesting cases. In the first case, the online algorithm may use twice as much capacity for each color as the offline algorithm, for which we present an online algorithm that uses at most 3 times the number of colors that the optimal offline algorithm uses. In the second case, we present an online algorithm that uses at most as many colors as OPT in the worst case, where each color has a capacity of 4.

Another interesting variant is to restrict the class of intervals, so all intervals are of the same length. This type of graph is called “Unit Interval Graph”. For the interval coloring problem we show that First Fit uses at most $2\omega - 1$ colors and that the analysis is tight. We also show a lower bound of $\frac{3}{2}$ on the competitive ratio of any online algorithm.

For interval coloring with bandwidth for unit interval graphs, we present several algorithms, the best algorithm has $\frac{7}{2}$ competitive ratio. We also present a lower bound of 2. For the asymptotic competitive ratio, the bounds become 3.17778 and 1.831.

For lazy interval coloring, we prove that for general instances any online algorithm performs arbitrarily bad. However if all intervals have the same length we present upper and lower bounds of 2. When introducing bandwidth to the lazy interval coloring problem, we show that any online algorithm is arbitrarily bad compared to the maximum weighted clique (even for unit interval graphs). We summarize our results and previous results in the following table.

	Interval Graph		Unit Interval Graph	
	LB	UB	LB	UB
Interval Coloring	3 [9]	3 [9]	$\frac{3}{2}$	2
Interval Coloring with Bandwidth	**3.2609**	10 [10]	1.831 2	3.17778 3.5
Lazy Interval Coloring	∞		2	2
Lazy Interval Coloring with Bandwidth	∞		∞	

Table 1. Results obtained in this paper and previous work. For each case, a single entry means that the results hold for both asymptotic and absolute competitive ratios. If two entries exist, the first one is the asymptotic competitive ratio and the second one stands for the absolute competitive ratio.

2 Preliminaries

A weighted interval graph G, is a graph where each node corresponds to an interval. The weight of the node is the bandwidth of the interval 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 $\omega(G)$ denote the size of the maximum cardinality clique in G (ω for short), *i.e.*, ignoring the weights. Let $\omega^*(G)$ (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 a generalized presentation of the algorithm of Kierstead and Trotter [9]. We use specific cases of the generalized algorithm for the variant of resource augmentation in the next sections. We present the algorithm using notations similar to these of [10].

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 $x \neq y$.

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 subsets of $V(G)$ and $E(G)$ respectively.
$V_m(v_i) = \{v_j \in V(G) : j < i, m(v_j) \leq m\}$;
$E_m(v_i) = \{(u, v) \in E(G) : u, v \in V_m(v_i)\}$;
$G_m(v_i) \cup \{v_i\} = G(V_m(v_i) \cup v_i, E_m(v_i) \cup \{(u, v_i) \in E(G) : u \in V_m(v_i)\})$
$\omega_i^*(H) =$ The size of the maximum weighted clique among cliques containing the interval v_i in graph H
2: Let $G_m(v_i) = G(V_m(v_i), E_m(v_i))$
3: $m(v_i) =$ the smallest m such that $\omega_i^*(G_m(v_i) \cup \{v_i\}) \leq m \cdot l$.
4: $A_{m(v_i)} \Leftarrow A_{m(v_i)} \cup \{v_i\}$
5: Color v_i considering only the intervals of $A_{m(v_i)}$ using First Fit on colors of $C_{m(v_i)}$.

A *critical point*, q, in interval $v_i \in A_{m(v_i)}$, is a point where $\omega^*(G_{m(v_i)-1}(v_i) \cup \{v_i\}) > (m(v_i)-1) \cdot l$. 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.

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 have 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 coloring of $KT_{l,b}$ is at most $\lceil \frac{\omega^*}{l} \rceil (\max_m FF(A_m))$, where $FF(A_m)$ denotes the coloring of the First Fit algorithm on the set C_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 [9]. In their algorithm every layer can be colored by First Fit with at most 3 colors. The number of layers equals to the size of the maximum cardinality clique. Therefore the coloring is at most $3OPT$.

2.1 Variants of Interval Coloring

Resource Augmentation in Interval Coloring: In the resource augmentation approach, the online algorithm is given more resources than the offline algorithm. Interval coloring is a natural problem to consider with resource augmentation. In this case, in the online coloring, the total bandwidth of intersecting intervals with the same color can exceed 1. The allowed maximum bandwidth of the intersecting intervals in the online coloring will be denoted by B. For the analysis we use the concept of competitive ratio.

Equal length intervals in Interval Coloring: If the intervals must be of equal length, the associated graph is called a unit interval graph. Recognition of unit interval graphs has been studied in [13, 4, 12]. It was also studied in the context of interval selection in [5]. For simplicity, we use intervals of length 1 in some of the proofs of this paper.

3 Interval Coloring

3.1 Resource Augmentation

In [9] Kierstead and Trotter give a lower and upper bound of 3. The main goal when resource augmentation is allowed, is to find a value of B for which there exists an algorithm with a competitive ratio 1. We argue that for $B = 2$, the online algorithm presented in [9] uses $\max\{1, OPT - 1\}$ colors. We also present a matching lower bound of $\max\{1, OPT - 1\}$ on the number of colors used by any online algorithm.

Theorem 1. *An adaptation of the algorithm of Kierstead and Trotter [9] can be used for interval coloring with* $B = 2$*, and uses at most* $\max\{1, OPT - 1\}$ *colors.*

Proof. **Algorithm:** $KT_{1,1}$ without bandwidth is exactly the the algorithm of Kierstead and Trotter.

According to Lemma 3 every set A_m has a maximum bandwidth of at most 2. Since $B = 2$, First Fit can color each A_m by a unique color. Moreover, by the definition of sets A_1 and A_2, the total bandwidth of intervals in $A_1 \cup A_2$ does not exceed 2. We use the same color for $m = 1, 2$, and one color for each other value of m. By Lemma 4, we get a coloring which uses $\lceil \omega^* \rceil - 1 \leq OPT - 1$ colors, if $OPT \geq 1$ and otherwise it uses a single color.

We prove that the above bound cannot be improved. Clearly if $OPT = 1$ the algorithm also uses at least one color, therefore we need to show a lower bound of $OPT - 1$. The lower bound construction holds only for the absolute competitive ratio.

Theorem 2. *There exists an infinite sequence of values of* α*, such that there exists an input for which* $OPT = \alpha$ *and any online algorithm for interval coloring with* $B = 2$*, uses at least* $OPT - 1$ *colors.*

Proof. We show that the claim holds for all even values of α. In general we show that when the largest clique size increases by 2, the online algorithm must use additional two colors.

Denote by S_i, the structure with maximum clique size of i that is colored by the online algorithm by $i-1$ colors denoted by $1, \ldots, i-1$ and for every such a color there is a point in which two intervals intersect. We claim that we can create such a structure for every even i. In particular we claim that using two S_i structures we can create one structure of S_{i+2}, on which any online algorithm uses $i + 1$ colors.

We prove the claim by induction on i, we construct the sequence of intervals into phases.

Initial Phase:

In the initial phase we provide disjoint sets of two identical intervals each. The number of intervals is determined by the clique size we wish to get. In this phase $OPT = 2$ and the online algorithm may use one color, denoted by 1. Note that for every set of two intervals there is a point in which the clique size is exactly 2 and colored by the same color 1. Since we provide sets of identical intervals, in this case, the intervals are colored by the same color and the complete intervals intersect. Observe that these structures have the properties required from S_2

Phase i:
We show that given two structures of intervals of S_i, we can create a new structure S_{i+2} in which the online algorithm colors this structure by $i+1$ colors and for every color there is a point in the structure where two intervals intersect.

Denote $S_{i,1}$ the first structure and $S_{i,2}$ the second structure, such that all the intervals in $S_{i,1}$ are to the left of all intervals in $S_{i,2}$ and there is no intersection between the intervals of these two different structures.

Denote by l_1, the left endpoint of the leftmost interval in $S_{i,1}$ and by r_1 the right endpoint of the rightmost interval in $S_{i,1}$. Denote l_2 and r_2 respectively for structure $S_{i,2}$. Note that $l_2 > r_1$. Present four intervals as follows. two identical intervals of $[l_1, \frac{l_2+r_1}{2}]$ and two identical intervals of $[\frac{l_2+r_1}{2}, r_2]$.

We claim that the new structure can be used as S_{i+2} (see figure).

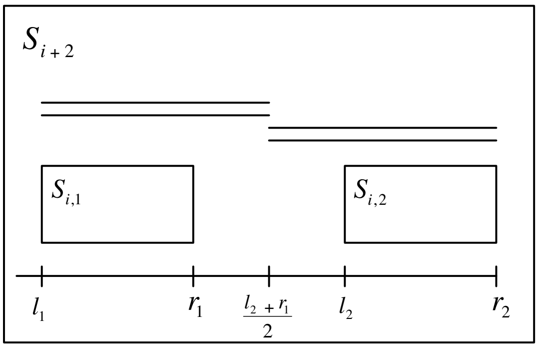

Fig. 1. The structure of S_{i+2} constructed in phase i.

By the hypothesis assumption, the colors $1, \ldots, i-1$ each have a point with two intersecting intervals therefore none of the four intervals could be colored by any of the colors $1, \ldots, i-1$. Since the clique size is increased by two, there are only two additional colors allowed in order for the online algorithm to have at most $OPT - 1$ colors, we denoted these two colors by i and $i+1$. Since all the four intervals intersect in the point $\frac{l_2+r_1}{2}$, we have a point where two intervals are colored by i and also a point where two intervals are colored by $i+1$. Since these are the properties required from S_{i+2}, the claim is proven.

3.2 Unit Interval Graphs

In the following 2 theorems, we show that First Fit uses at most $2\omega - 1$ colors for unit interval graphs and that the analysis is tight.

Theorem 3. *First Fit uses at most $2\omega - 1$ colors for coloring unit interval graphs.*

Proof. In this version, since all the intervals are of equal length, if the maximum clique equals to ω, then there is no interval that intersects with more than $2\omega - 2$ different intervals. Assume by contradiction that there exists an interval $I = [x, x+1]$ that does intersect with more than $2\omega - 2$ different intervals. Since all intervals are of unit length, every interval that intersects I, must contain either the point x or $x+1$ (or both). Therefore either the point x or the point $x+1$ is contained in more than $\omega - 1$ intervals not including interval I. This contradicts the fact that the maximum clique size does not exceeds ω. Therefore First Fit uses at most $2\omega - 1$ colors.

Theorem 4. *There exist unit interval graphs on which First Fit uses exactly $2\omega - 1$ colors.*

Proof. We show a family of instances, where k is the size of largest clique on which First Fit uses exactly $2k - 1$ colors to color it. We define the sequence of intervals in phases.

Phase i ($1 \le i \le 2k-1$) contains $2k - i$ intervals that receive the same color by First Fit. The largest clique size after phase i is $\lfloor \frac{i}{2} \rfloor + 1$. The intervals of phase i are $[i - 1 + 2^{-i+1} + 2j, i - 1 + 2^{-i+1} + 2j + 1]$, for $0 \le j < 2k - i$. All intervals have length 1, and all consecutive intervals of one phase have fixed distances of 1 between them.

We prove the following claims.

1. The largest clique after phase $2\ell + 1$ is of size $\ell + 1$, and in ranges of the form $(2j, 2j + 2^{-2\ell})$, the total requested bandwidth is at most ℓ.
2. All intervals of phase i receive the color i.

Proof of 1: We prove the claim by induction. After phase 1 ($\ell = 0$), the largest clique is clearly of size 1 (the intervals are non-intersecting). The ranges $(2j, 2j+1)$ are empty since the intervals of phase 1 start at odd points. Assume now that the claim holds for $\ell = s - 1$ and prove for $\ell = s$. Phases $2s$ and $2s+1$ introduce two sets of intervals. To show the first part of the claim we need to show that the overlap between the intervals does not overlap with areas where the largest clique is s. Non overlapping parts of new intervals may increase the size of the largest clique by 1. By definition, the overlap interval between intervals of these two phases are intervals of the form $[2j + 2^{-2s}, 2j + 2^{1-2s}]$. Using the inductive hypothesis, intervals of the form $(2j, 2j + 2^{2-2s})$ have bandwidth request of size $s - 1$ only, and therefore the largest clique in these intervals after phase $2s + 1$ does not exceed $s + 1$. To prove the second part of the claim, the interval $(2j, 2j + 2^{-2s})$ is not a part of the overlap between phases $2s$ and $2s+1$, therefore its bandwidth request increases by at most 1, and becomes at most s.

Proof of 2: We prove that an interval of phase i intersects with intervals of all smaller colors. If i is odd, then its left endpoint intersects with all intervals of even colors, and its right endpoint with all intervals of odd colors. If i is even, its left endpoint intersects with intervals of all odd colors, and its right endpoint with even colors. This can be easily verified by the intervals definitions.

Theorem 5. *Any online algorithm for unit interval graphs has a competitive ratio of at least $\frac{3}{2}$*

Proof. We divide the construction of the intervals of the lower bound into three phases.

Initial Phase:

In the initial phase we provide $\frac{\omega}{2}$ identical requests for $[0, 1]$, ω can be any even number. The online algorithm has to color these intervals with exactly $\frac{\omega}{2}$ colors, denote those colors by c_1,...,$c_{\frac{\omega}{2}}$ and the set of those colors by C_{onilne}

Phase 2:
In the next phase we present at most ω intersecting intervals. These intervals are presented one by one in a way that all intervals colored by some color, c, where $c \in C_{onilne}$ are slightly shifted to the right with respect to any interval that is colored by a color $\bar{c}$, where $\bar{c} \notin C_{onilne}$. We present intervals until exactly $\frac{\omega}{2}$ of them are colored by colors that are not in C_{onilne}

Let $I_1 = [a, a+1]$ be the rightmost interval colored by $\bar{c}$ and let $I_2 = [d, d+1]$ be the leftmost interval colored by c among intervals introduced so far. If there is no interval colored $\bar{c}$ we say that I_1 is empty and If there is no interval colored c we say that I_2 is empty. For $0 < \varepsilon << \frac{1}{4\omega}$ a new interval, I, is presented as follows.

1. If both I_1 and I_2 are empty (this holds only when we introduce the first interval) then $I = [1.5, 2.5]$.
2. If only I_1 is empty, $I = [d - \varepsilon, d + 1 - \varepsilon]$
3. If only I_2 is empty, $I = [a + \varepsilon, a + 1 + \varepsilon]$
4. If I_1 and I_2 are not empty then, $I = [\frac{d+a}{2}, \frac{d+a}{2} + 1]$, *i.e.* the interval is halfway between I_1 and I_2 with unit length, intersecting all previous intervals presented in this step.

Note that none of the intervals in this phase intersect intervals of the initial phase. Moreover, the left endpoints of all the intervals in the phase are located within a distance of 1 from the right endpoints of the intervals of the initial phase. Also note that the algorithm stops after introducing at most ω intervals, at that time, if it is reached, there are exactly $\frac{\omega}{2}$ intervals with a color that is not in C_{online}, since $|C_{online}| = \frac{\omega}{2}$.

Phase 3:
Suppose that $[x, x+1]$ is the rightmost interval with color $\bar{c}$ which is not in C_{online} (from the construction we have $1 < x < 2$) after all intervals from phase 2 were presented. we present $\frac{\omega}{2}$ identical intervals $[x-1, x]$. These intervals intersect all the intervals with color not in C_{online} from the previous phase. They also intersect all the intervals from the initial phase.

To complete the analysis, note that the intervals presented in the last phase all intersect with intervals of exactly ω different colors. There are $\frac{\omega}{2}$ colors in C_{online} and $\frac{\omega}{2}$ colors not in C_{online} from the second phase. This gives a coloring of $\frac{3\omega}{2}$ colors while OPT can color with only ω colors.

4 Online Coloring of Intervals with Bandwidth

Interval coloring with bandwidth was recently studied by [1] and [10]. Adamy and Erlebach [1] gave a 195-competitive ratio algorithm and Narayanaswamy [10] gave a 10-competitive ratio algorithm. However, is this case really harder? In the theorem below we answer that question affirmatively.

We give a lower bound which is strictly higher than the upper bound for the problem of interval coloring without bandwidth presented in [9]. To prove the lower bound we adapt the lower bound on classical interval coloring given in [9]. In that paper a lower bound of of $3\omega - 2$ colors is shown, for inputs where $OPT = \omega$. Since interval graphs are perfect, these are exactly inputs where the largest clique has size ω. Note that two intervals whose width is strictly larger than $\frac{1}{2}$ cannot have the same color. Therefore the same lower bound can be applied not only for intervals of width 1, but for intervals of arbitrary widths in $(\frac{1}{2}, 1]$. In this case, let q be the largest number of intersecting intervals of width in $(\frac{1}{2}, 1]$, then we immediately get a lower bound of $3q-2$. Finally, we make another adaptation to the lower bound, namely, we make use of the following lemma.

Lemma 5. *The lower bound $3q-2$ on the number of colors holds even if q is given in advance.*

Proof. The proof is an adaptation of the lower bound in [9]. The construction works in phases, after each phase we shrink some parts of the line into single points. Given 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 colors which no interval that contains it can receive. This is done for convenience purposes. 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 it has no overlap with any of them.

The sequence construction stops once $3q-2$ colors have been used. Therefore we may assume that we are initially given a palette of $3q-3$ colors, $1, \ldots, 3q-3$, that can all be used by the algorithm. The ith color ever used is called color number i. As soon as color $3q-2$ is used, the proof is complete. We construct an input where the largest clique is of size q and therefore $OPT = q$.

The sequence starts with introducing $(4(3q)!)^q$ intervals, this is phase 0. Since the algorithm is using at most $3q-3$ colors, this means that there exists a set of $\frac{(4(3q)!)^q}{3q-3}$ intervals that share the exact same color. We shrink all intervals into single points. Later phases result in additional points. We now define phase i. The phases are constructed in a way that in the beginning of phase i there are at least $4(4(3q)!)^{q-i}$ points that contain a given set of $3i-2$ colors (points of interest). Without loss of generality, assume that these are colors $1 \ldots, 3i-2$ where the size of the largest clique is i. There exist some other points containing other sets of i colors, or sets of at most $i-1$ colors. All these points are called void points. At this time, we partition the points of interest into at least $(4(3q)!)^{q-i}$ consecutive sets of four. Points of interest that do not participate become void points.

We next define additional intervals, increasing the size of largest clique by exactly one. Given a set of four points 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 c be the rightmost void point between a_3 and a_4, and if no such point exists then $c = \frac{a_3+a_4}{2}$. Let d 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 $I_2 = [\frac{c+a_4}{2}, a_4]$. Clearly non of them may receive one of the currently used $3i-2$ colors. If they both receive the same new color, we introduce the intervals $I_3 = [\frac{a_1+b}{2}, d]$ and $I_4 = [d, \frac{c+a_4}{2}]$. The interval I_3 intersects with a_2, and with I_1. Therefore it receives an additional color. The second interval I_4 intersects I_3, a_3 and I_2. Therefore a third new color is given to it. If I_1, I_2 receive distinct new colors, we introduce the interval $I_5 = [\frac{a_1+b}{2}, \frac{c+a_4}{2}]$. Since I_5 intersects with I_1, I_2, a_2, a_3, it must get a third new color. We shrink every such interval $[a_1, a_4]$ into a single point containing $3i+1$ colors. Since there are less than $3k$ colors, and each point uses exactly $3i+1 < 3k$ of them, there are less than $(3k)!$ such choices, and we can pick $4(4(3q)!)^{q-i-1}$ of them having the same set of colors. The points containing this exact set 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 1 exactly.

At this time we can perform phase $i+1$. After phase $k-1$, there are at least $3k-2$ colors in use and the claim is proved. Note that prior to that phase, a minimum number of four points of interest is required.

With this, we are ready to prove our main theorem of this section.

Theorem 6. *Any deterministic online algorithm for interval coloring with bandwidth has competitive ratio of at least* 3.2609

Proof. Let α be a constant rational number fixed later and let t be an integer such that αt is integer (there are infinitely many such values of t). Let $\varepsilon = \frac{1}{2\alpha t+1}$. The first phase of requests is a large number $T = N!(2\alpha t+1)$ of identical requests (for a large enough integer N), all of bandwidth ε. The requests are for a long enough interval. All future requests will be given within this interval, and therefore they all intersect the initial requests. The sequence either stops here (and has $OPT = N!$) or continues with a second phase which contains requests which are all of bandwidth $1 - \frac{k}{2\alpha t+1} = 1 - k\varepsilon$, for some $t \leq k \leq \alpha t$.

Furthermore, these requests all have bandwidth larger than $\frac{1}{2}$, therefore if they are packed independently from the first phase, they are treated as requests of width 1. Adding the first phase means that colors that were used for at least $k+1$ intervals, cannot be used again in the second phase, and other colors can be used again. Intervals of the second phase are introduced as in Lemma 5 so that the optimal number to color them is ω, and the number of colors used in this phase is 3ω. If the bandwidth of these intervals is $1-k\varepsilon$, then we use $\omega = \frac{T}{k}$, so that an optimal coloring uses ω colors, and each color is used for k intervals of the first phase. We denote by X_i $(1 \leq i \leq \alpha t)$ the number of colors used in the first phase for exactly i intervals. The algorithm has no reason to color less than t intervals with one color, since a color used for t intervals can always be used again, therefore $X_i = 0$ for $i < t$. If a color is used for more than αt intervals, it will not be used again, so we can assume that in such a case, the color is used for $\frac{1}{\varepsilon}$ intervals. We denote by Y the number of colors that are used for this maximum number of intervals. We have $T = Y(2\alpha t+1) + \sum_{i=t}^{\alpha t} i \cdot X_i$. If there is no second phase, $OPT = T\varepsilon$ and $ALG = Y + \sum_{i=1}^{2\alpha t} X_i$. Otherwise, we compute the number of colors used by the algorithm for a specific choice of $t \leq k \leq \alpha t$. We get $ALG = Y + \sum_{i=k+1}^{\alpha t} X_i + 3\frac{T}{k} = Y + \sum_{i=k+1}^{\alpha t} X_i + 3 \cdot OPT$.

Let C be the competitive ratio of ALG. We have $Y + \sum_{i=1}^{\alpha t} X_i \leq C \cdot T\varepsilon$ and for every $t \leq k \leq \alpha t$, $Y + \sum_{i=k+1}^{j} X_i \leq (C-3)\frac{T}{k}$. We multiply the first inequality by t, the last inequality (*i.e.*, the second inequality for $k = \alpha t$) by $\alpha t+1$, and all other inequalities by 1 (*i.e.*, the second inequality for all other values of k). We sum them and get $T = \sum_{i=t}^{\alpha t} iX_i + (2\alpha t+1)Y \leq C \cdot T\frac{t}{2\alpha t+1} + (C-3)\sum_{i=t}^{\alpha t} \frac{T}{2\alpha t+1} + (C-3)T$. Letting t tend to infinity, we have $\sum_{i=t}^{\alpha t} \frac{T}{2\alpha t+1} \to \ln \alpha$. We get $C \geq \frac{3\ln\alpha+4}{\frac{1}{2\alpha}+\ln\alpha+1}$. Solving in Maple, we see that for an appropriate choice of α this gives a lower bound of 3.2609.

4.1 Resource Augmentation

In this section we consider two interesting possible values of B. For $B = 4$ we provide an online algorithm that can color with $OPT - 3$ colors. For $B = 2$ we provide an online algorithm that uses

$3OPT - 2$ colors. The lower bound on the competitive ratio for $B = 2$ is at least 1 as was shown for the case of interval coloring without bandwidth.

Proposition 1. *An adaptation of the algorithm of Kierstead and Trotter [9] can be used for interval coloring with $B = 4$, and uses at most $\max\{1, OPT - 3\}$ colors.*

Proof. We use **Algorithm:** $KT_{1,1}$ with bandwidth. According to Lemma 2 every A_m has a maximum bandwidth of at most 4. By Lemma 4 the coloring of $KT_{1,1}$ with bandwidth, is $\omega^* \cdot 1 = \omega^*$. Since $B = 4$, First Fit can color each A_m by a unique color. Moreover, by the definition of sets $A_1, \ldots, A_4$ bandwidth of intervals in $A_1 \cup A_2 \cup A_3 \cup A_4$ does not exceed 4. We use the same color for $m = 1, 2, 3, 4$, and one color for each other value of m. By Lemma 4, we get a coloring which uses $\lceil \omega^* \rceil - 3 \leq OPT - 3$ colors. If $m \leq 4$ we get a coloring using a single color.

Proposition 2. *There exists an online algorithm for interval coloring with bandwidth with $B = 2$, that uses at most 3OPT-1 colors.*

Proof. We split the intervals into two sets, and color them independently.
Algorithm:

Case 1 For intervals I where $b(I) \leq \frac{1}{2}$, we run $KT_{1,\frac{1}{2}}$ with bandwidth;
Case 2 For intervals I where $b(I) > \frac{1}{2}$ we run $KT_{1,1}$ without bandwidth.

Case 1 uses at most $\lceil 2\omega^* \rceil \leq 2\lceil \omega^* \rceil$ colors. By Lemma 2 every A_m has a maximum total bandwidth of 2. Since B=2, we can color each A_m by one color only. By Lemma 4, the coloring which $KT_{1,\frac{1}{2}}$ with bandwidth outputs, is of size at most $\lceil \frac{\omega^*}{\frac{1}{2}} \cdot 1 \rceil = \lceil 2\omega^* \rceil$. If we use the same color for $m = 1, 2$ we get a coloring of $2\omega^* - 1 \geq 2OPT - 1$ colors.

Case 2 uses at most OPT colors, similarly to the case of coloring without bandwidth. The total size of the coloring is therefore at most $3OPT - 1$.

4.2 Unit Interval Graphs

For this version we present three algorithms.

1. **First Fit.**

2. **2-First Fit.** Perform an online partition of the intervals into two subsequences according to the bandwidth of the intervals. One subsequence for intervals with bandwidth b such that $b \leq \frac{1}{2}$ and the other for intervals with bandwidth b such that $b > \frac{1}{2}$. Apply First Fit on each subsequence separately with disjoint sets of colors.

3. **Odd-Even bin packing.** Scale the real line into integers and assume all intervals are of unit length .

Perform an online partition of the intervals into two subsequences called *evens* and *odds*. Each interval intersects an integer point. If an interval is exactly between two integers, assign it to the left integer point, and otherwise there is a unique integer point. If the integer point is an even number assign the interval into the evens subsequence, otherwise assign it to the odds subsequence. Apply the best online bin packing algorithm known separately for the odds subsequence and for the evens subsequence using two disjoint sets of colors. Each class of intervals that was assigned to an integer point is handled as an instance of a bin packing problem.

In the following we show that algorithms First Fit, 2-First Fit and Odd-Even bin packing have an absolute competitive ratios of at most the values 8, 6, 3.5 respectively.

Intensive research has been done analyzing the performance of the simple First Fit algorithm for the problem of interval coloring. Adamy and Erlebach [1] argue that First Fit is arbitrarily bad when introducing bandwidth. In the following theorem we show that on unit interval graph the competitive ratio of first fit is constant.

Theorem 7. *(i) Algorithm First Fit has a competitive ratio of* 8 *for unit interval graphs with bandwidth. (ii) Algorithm First Fit has a competitive ratio of* 4 *for unit interval graphs with bandwidth if each interval has a bandwidth of at most* $\frac{1}{2}$.

Proof. Consider the color assignment for a new interval. We show that the color used for this interval never exceeds the number $8\lceil\omega^*\rceil$, and if all intervals have bandwidth of at most $\frac{1}{2}$, then the color number never exceeds $4\lceil\omega^*\rceil$. Denote a new interval by $J = [x, x+1]$. If the interval receives a color that was already used, the claim certainly holds. Assume therefore that J receives a new color that was not used before. Let p be the color assigned to J, all colors j, $1 \le j < p$ were used. If $p \le 2$ then since the input is non-empty, $\lceil\omega^*\rceil \ge 1$, and the claim holds. Otherwise assume that $p > 2$. Since the input is a unit interval graph, all intervals intersecting with the input interval, intersect with it also in one of the points x or $x+1$. We first discuss the case where all bandwidths are bounded by $\frac{1}{2}$ (and therefore $w(J) \le \frac{1}{2}$).

Consider the intervals which were already assigned a color. For a point y and a color $c < p$, let $A(c, y)$ be the set of intervals which contain the point y and received color c. Let $W(c, y)$ be the total bandwidth of intervals in $A(c, y)$. For each color $j < p-1$, let y_j be the point in the interval $[x, x+1]$ which maximizes $W(j, y)$. Note that every interval in $A(j, y_j)$ intersects also either x or $x+1$, therefore it contributes to $W(j, x)$ or to $A(j, x+1)$ (or to both). For colors $p-1$ and p, let $y_{p-1} = y_p$ a point that maximizes $W(p-1, y) + W(p, y)$. Therefore $2\omega^* \ge \sum_{j=1}^{p} (W(j, x) + W(j, x+1)) \ge \sum_{j=1}^{p} W(j, y_j)$. However, since interval J receives a new color, and since $W(J) \le \frac{1}{2}$, we have that every color $j < p-1$ has a point z_j where $W(j, z_j) + W(J) > 1$. Therefore $W(j, y_j) \ge W(j, z_j) > 1 - W(J) \ge \frac{1}{2}$. Also, $W(p, y_p) + W(p-1, y_{p-1}) > 1$. We get that $\sum_{j=1}^{p} W(j, y_j) > \frac{p}{2}$ and $p < 4\omega^*$.

Consider now the general case. Note that the previous proof still holds if $w(J) \le \frac{1}{2}$, therefore we assume $w(j) > \frac{1}{2}$. We define the points y_j as before, only in the current case these are points maximizing $W(j, y)$ where $y \in [x-1, x+2]$. Let ℓ be the first color such that $W(j, y_j) < \frac{1}{2}$. If ℓ does not exist, then we focus on the four points $x-1, x, x+1, x+2$. Each interval in $A(j, y_j)$ contributes to one of the four values $W(j, x-1+a)$ for $a = 0, 1, 2, 3$. Therefore

$$4\omega^* \ge \sum_{j=1}^{p} (W(j, x-1) + W(j, x) + W(j, x+1) + W(j, x+2))$$
$$\ge \sum_{j=1}^{p} W(j, y_j) \ge \frac{p}{2},$$

which gives $p < 8\omega^*$. If ℓ exists, consider an interval T colored with $j > \ell$ which intersects the point x or the point $x+1$. This interval was not colored ℓ since there exists a point z where

$W(\ell, z) + w(J) > 1$. Since $W(\ell, z)$ can only increase over time, we get that $W(T) > 1 - W(\ell, z) \geq 1 - W(\ell, y) \geq \frac{1}{2}$. Since every color $j < p$ has at least one interval intersecting with x or $x+1$, we get that ℓ is the only color for which $W(j, y_j) < \frac{1}{2}$. Note that if we define the point $y_\ell = y_p$ to be the point in $[x, x+1]$ where $W(\ell, y_\ell) + W(J) > 1$ (which exists since J was not colored with ℓ), we get again $4\omega^* \geq \sum_{j=1}^{p} (W(j, x-1) + W(j, x) + W(j, x+1) + W(j, x+2)) \geq \sum_{j=1}^{p} W(j, y_j) \geq \frac{p}{2}$, which gives $p < 8\omega^*$.

Proposition 3. *Algorithm 2-First Fit uses at most* 6ω *colors.*

Proof. 2-First Fit uses different sets of colors for intervals of bandwidth in $(0, \frac{1}{2}]$ and in $(\frac{1}{2}, 1]$. By Theorem 7 part *(ii)* First Fit on intervals with bandwidth of at most $\frac{1}{2}$ has a competitive ratio of 4. By Theorem 3 First Fit for intervals with bandwidth that exceeds $\frac{1}{2}$, the competitive ratio is 2. Combining these competitive ratios we get a competitive ratio of at most 6.

Proposition 4. *(i) Algorithm Odd Even bin packing has an absolute competitive ratio of 3.5 for coloring unit interval graphs with bandwidth, using First Fit as the online bin packing algorithm. (ii) The asymptotic competitive ratio of the algorithm Odd Even bin packing is at most 3.17778 using the algorithm Harmonic++ of Seiden [14].*

Proof. We claim that the odds subsequence can be split into different classes, such that an interval intersects all intervals within its class, but no other intervals. Each class is represented by an odd number and it contains all the intervals that were assigned to that odd integer by the algorithm. Same argument holds for the evens subsequence. Note that, since all intervals of the same class intersect, each class can be viewed as an instance to the online bin packing problem. For the first part we use the fact that First Fit for bin packing has competitive ratio of at most 1.75 with respect to the absolute measure [15]. Since we use it with two sets of colors, we get a competitive ratio of at most 3.5. For the second part we use the Harmonic++ algorithm of Seiden [14] and therefore get 3.17778.

Theorem 8. *Any online algorithm for unit interval graph with bandwidth has an absolute competitive ratio of at least 2. Any online algorithm for unit interval graph with bandwidth has an asymptotic competitive ratio of at least 1.831.*

Proof. First we prove the absolute competitive ratio lower bound of 2. We introduce two identical intervals $I_1 = I_2 = [1, 2]$ and have bandwidth $\frac{1}{3}$. If they are assigned distinct colors then already $ALG = 2$ and $OPT = 1$, and we are done. Otherwise all future intervals have bandwidth $\frac{2}{3}$ and intersect with the previous intervals. This means that no future interval has the same color as the first two. Two further intervals are first given, $I_3 = [\frac{1}{5}, \frac{6}{5}]$ and $I_4 = [\frac{9}{5}, \frac{14}{5}]$. If they receive distinct colors, we introduce the interval $I_5 = [1, 2]$ which gets a fourth color. It is possible to color using two colors only, coloring I_1, I_3, I_4 with one color and I_2, I_5 with a second color. If they receive the same color, we introduce two intervals $I_6 = [\frac{3}{5}, \frac{8}{5}]$, $I_7 = [\frac{7}{5}, \frac{12}{5}]$, which must receive two new colors. The total number of colors used is again 4, while it is possible to color using only two colors, one color for I_1, I_3, I_7, and a second color for I_2, I_4, I_6.

Next, we prove the lower bound 1.831. Let α be a constant rational number fixed later and let t be an integer such that αt is integer (there are infinitely many such values of t). Let $\varepsilon = \frac{1}{2\alpha t+1}$.

The first phase of requests is a large number $T = N!(2\alpha t+1)$ of identical requests for intervals $[0,1]$ (for a large enough integer N), all of bandwidth ε. The sequence either stops here (and has $OPT = N!$) or continues with a second phase which contains requests which are all of bandwidth $1 - \frac{k}{2\alpha t+1} = 1 - k\varepsilon$, for some $t \leq k \leq \alpha t$. All the new requests intersect with requests of the first phase. Furthermore, these requests all have bandwidth larger than $\frac{1}{2}$, therefore if they are packed independently from the first phase, they are treated as requests of width 1. Adding the first phase means that colors that were used for at least $k+1$ intervals, cannot be used again in the second phase, and other colors can be used again. Intervals of the second phase are introduced as in Theorem 5, so that the optimal number to color them is ω, and the number of colors used in this phase is $\frac{3\omega}{2}$. If the bandwidth of these intervals is $1-k\varepsilon$, then we use $\omega = \frac{T}{k}$, so that an optimal coloring uses ω colors, and each color is used for k intervals of the first phase. We denote by X_i ($1 \leq i \leq \alpha t$) the number of colors used in the first phase for exactly i intervals. The algorithm has no reason to color less than t intervals with one color, since a color used for t intervals can always be used again, therefore $X_i = 0$ for $i < t$. If a color is used for more than αt intervals, it will not be used again, so we can assume that in such a case, the color is used for $\frac{1}{\varepsilon}$ intervals. We denote by Y the number of colors that are used for this maximum number of intervals. We have $T = Y(2\alpha t+1) + \sum\limits_{i=t}^{\alpha t} i \cdot X_i$. If there is no second phase, $OPT = T\varepsilon$ and $ALG = Y + \sum\limits_{i=1}^{2\alpha t} X_i$. Otherwise, we compute the number of colors used by the algorithm for a specific choice of $t \leq k \leq \alpha t$. We get $ALG = Y + \sum\limits_{i=k+1}^{\alpha t} X_i + \frac{3}{2} \cdot \frac{T}{k} = Y + \sum\limits_{i=k+1}^{\alpha t} X_i + \frac{3}{2} \cdot OPT$.

Let C be the competitive ratio of ALG. We have $Y + \sum\limits_{i=1}^{\alpha t} X_i \leq C \cdot T\varepsilon$ and for every $t \leq k \leq \alpha t$, $Y + \sum\limits_{i=k+1}^{j} X_i \leq (C - \frac{3}{2})\frac{T}{k}$. We multiply the first inequality by t, the last inequality (*i.e.*, the second inequality for $k = \alpha t$) by $\alpha t + 1$, and all other inequalities by 1 (*i.e.*, the second inequality for all other values of k). We sum them and get $T = \sum_{i=t}^{\alpha t} iX_i + (2\alpha t+1)Y \leq C \cdot T \frac{t}{2\alpha t+1} + (C - \frac{3}{2}) \sum_{i=t}^{\alpha t} \frac{T}{2\alpha t+1} + (C-\frac{3}{2})T$. Letting t tend to infinity, we have $\sum_{i=t}^{\alpha t} \to \ln \alpha$. We get $C \geq \frac{\frac{3}{2}\ln\alpha + \frac{5}{2}}{\frac{1}{2\alpha} + \ln\alpha + 1}$. Solving in Maple, we see that for an appropriate choice of α this gives a lower bound of 1.83157.

5 Lazy Online Interval Coloring

Motivated by The Maximum Resource Bin Packing Problem, we introduce a new problem called Lazy Online Interval Coloring. In this problem the objective is to use as many colors as possible. A newly presented interval can be colored by a new color only if it intersects intervals with all the previously used colors.

Theorem 9. *Any online algorithm for the problem Lazy Online Interval Coloring is arbitrarily bad.*

Proof. The construction is organized in phases. In the initial phase we provide two intersecting intervals that are colored by two colors by any online algorithm. We denote these colors by 1 and 2. For the next phases, it suffices to show that for every phase i, we increase the number of colors used by OPT from $i+1$ to $i+2$ while the number of colors used by the online algorithm remains two. Intervals from every phase are independent of (*i.e.*, non-intersecting) intervals from different phases.

The first phase (Phase 0) consists of two identical intervals which are $[0, 1]$. Any algorithm colors them using two colors.

Phase i:

In the first step we present at most $2i + 1$ non-intersecting intervals. The online algorithm must color these intervals with colors 1 and 2. These intervals are presented such that all intervals colored by 1 by the online algorithm are consecutive and to the left of all intervals that are colored 2. The intervals are presented as follows.

To present a new interval, let $I_1 = [a, b]$ be the rightmost interval colored by 1 and let $I_2 = [d, e]$ be the leftmost interval colored by 2. If there is no interval colored 1 we say that I_1 is empty and if there is no interval colored 2 we say that I_2 is empty. Let $0 \leq \varepsilon << \frac{1}{16i}$, a new interval, I, is presented as follows.

1. If both I_1 and I_2 are empty (this situation occurs only before the first interval is presented) then $I = [2i, 2i + 1]$. This interval is on the right hand side of all intervals from previous phases. All intervals of the current phase would not intersect with intervals from previous phases.
2. If only I_1 is empty, $I = [d - 2\varepsilon, d - \varepsilon]$
3. If only I_2 is empty, $I = [b + \varepsilon, b + 2\varepsilon]$
4. If I_1 and I_2 are not empty then, $I = [\frac{3b+d}{4}, \frac{b+3d}{4}]$, *i.e.* between I_1 and I_2 with half the size of the distance between I_1 and I_2.

After at most $2i + 1$ intervals are given, there are at least $i + 1$ consecutive intervals with the same color. In the next step, an interval that intersects $i + 1$ consecutive intervals of the same color is presented. The online algorithm has to color this interval by the other color and therefore does not increase the number of colors. An offline algorithm colors these $i + 1$ consecutive intervals using the $i + 1$ distinct colors it used in the past. The next interval must be colored by a new color since it intersects intervals that are colored with distinct $i + 1$ colors. This results in $i + 2$ colors used by the offline algorithm.

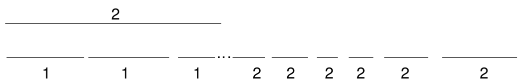

Fig. 2. The coloring of the online algorithm for phase i in a possible execution of the lower bound for lazy intervals coloring without bandwidth. Note that in this example the last interval presented in this phase intersects only intervals colored with color 1.

5.1 Unit Interval Graphs

Proposition 5. *Any online algorithm for the Lazy Online Interval Coloring with equal length intervals uses at least* $\frac{OPT+1}{2}$ *colors.*

Proof. In this version, since all the intervals are of equal length, if the maximum size clique equals to ω, then there is no interval that intersects with more than $2\omega - 2$ different intervals (see proof of Theorem 3). Therefore OPT can only use as much as $2\omega - 1$ colors. Since any online coloring uses at least ω we get a coloring of at least $\frac{OPT+1}{2}$ colors.

Theorem 10. *There is an upper bound of $\frac{OPT+1}{2}$ on the coloring of any online algorithm for the Lazy Online Interval Coloring problem on unit interval graphs.*

Proof. In the following sequence of intervals, we show that in fact an offline algorithm can use $2\omega - 1$ colors for a maximum clique of ω. We also show that any online algorithm uses exactly ω colors.

We construct the input sequence of intervals in phases. In the initial phase we provide ω intersecting intervals that are colored using ω colors by both the online algorithm and the offline algorithm. We denote these colors that the online algorithm uses by $1, 2, ..\omega$. For the next phases, it suffices to show that for every phase i, for $i = 1, ...\omega - 1$, we increase the number of colors used by OPT from $\omega + i - 1$ to $\omega + i$ while the number of colors used by the online algorithm remains ω.

Phase i:
Let $x = 10i$ and $\varepsilon << \frac{1}{4\omega}$.

Step 1 we present the following sequence of at most ω intervals until the online algorithm uses the color ω. $[a_{i_1}, a_{i_1} + 1], [a_{i_2}, a_{i_2} + 1], .., [a_{i_k}, a_{i_k} + 1]$, for $k \leq \omega$.
The first interval is $[a_{i_1}, a_{i_1} + 1] = [x - \frac{9}{4}, x - \frac{5}{4}]$, and for every ,$j$, $[a_{i_j}, a_{i_j} + 1] = [a_{i_{j-1}} + \varepsilon, a_{i_{j-1}} + 1 + \varepsilon]$, *i.e.*, every interval is shifted by ε to the right. Once online algorithm colors any interval in the sequence by ω we stop.

Step 2 Present the following sequence of at most ω intervals until the online algorithm uses the color ω.
$[b_{i_1}, b_{i_1} + 1], [b_{i_2}, b_{i_2} + 1], .., [b_{i_m}, b_{i_m} + 1]$, for $m \leq \omega$. The first interval in this step is $[b_{i_1}, b_{i_1} + 1] = [x + \frac{5}{4}, x + \frac{9}{4}]$, and for every ,$j$, $[b_{i_j}, b_{i_j} + 1] = [b_{i_{j-1}} - \varepsilon, b_{i_{j-1}} + 1 - \varepsilon]$, *i.e.*, every interval is shifted by ε to the left. when the online algorithm colors any interval in the sequence by ω we stop.

Step 3 The intervals that the online algorithm colored by ω are $[a_{i_k}, a_{i_k} + 1]$ and $[b_{i_m}, b_{i_m} + 1]$ for some $m, k \leq \omega$. In the next step, we present $\omega - 1$ intervals $[a_{i_k} + 1, a_{i_k} + 2]$ intersecting only $[a_{i_k}, a_{i_k} + 1]$ from the first step and $\omega - 1$ intervals $[b_{i_m} - 1, b_{i_m}]$ intersecting only $[b_{i_m}, b_{i_m} + 1]$ from the second step. In this step the online algorithm must color all $2(\omega - 1)$ intervals by colors $1, ..., \omega - 1$ since they all intersect an interval colored ω. Note that the two sets of identical intervals do not intersect each other.

Step 4 In the final step we present the interval $[x - \frac{1}{2}, x + \frac{1}{2}]$. This interval intersect all intervals from previous step and none of the intervals from steps 1 and 2.

The online algorithm has to color the interval presented in the last step by ω, maintaining a coloring of exactly ω colors. On the other hand an offline algorithm can increase the number of colors by one. The intervals of step 1 are colored with colors $k, ldots, 1$ (from left to right). The intervals of step 2 are colored with colors $\omega, \ldots, \omega - m + 1$ (from left to right). The intervals of step 3 are colored as follows. The ones that intersect the interval colored by color 1 (from step 2) receive colors $2, \ldots, \omega$. The other intervals of step 3 are colored with colors $\omega + i - 1, \ldots, \omega + 1, 1, \ldots, \omega + 1 - i$ if $i > 1$ and with colors $1, \ldots, \omega - 1$ otherwise. Therefore the intervals of step 3 is colored with all the colors in $1, ..., \omega + i - 1$. The interval from the last step is therefore colored by $\omega + i$.

5.2 Lazy Online Coloring of Intervals with Bandwidth

In this case we show that no algorithm is competitive even on unit interval graphs.

Theorem 11. *Any online algorithm for lazy online coloring of intervals with bandwidth and equal length intervals is arbitrarily bad.*

Proof. For an integer $D > 0$, we show that the competitive ratio is at least D. We break up the construction of the sequence of intervals into $2D-1$ phases. In the initial phase (phase 0) we provide two identical intervals of bandwidth 1 which are requests for $[0, 1]$. These are colored by two colors by any online algorithm. We denote these colors by 1 and 2. For the next phases, it suffices to show that for every phase i, we increase the number of colors used by OPT from $i+1$ to $i+2$ while the number of colors used by the online algorithm remains two.
Phase i: In the first step we present intersecting intervals of bandwidth $\delta = \frac{1}{2D}$. The amount of intervals never exceeds $2D$ so the total requested bandwidth is at most 2. The online algorithm can only color these intervals with colors 1 and 2. These intervals are presented such that all intervals colored by 1 by the online algorithm are slightly shifted to the left with respect to all intervals that are colored 2. Moreover all the intervals presented here intersect. We present exactly $2i+1$ intervals in the following way.

Let $I_1 = [a, a+1]$ be the rightmost interval colored by 1 and let $I_2 = [d, d+1]$ be the leftmost interval colored by 2. If there is no interval colored 1 we say that I_1 is empty and If there is no interval colored 2 we say that I_2 is empty. For $0 < \varepsilon << \frac{1}{12D}$ a new interval, I, is presented as follows.

1. If both I_1 and I_2 are empty (presentation of the first interval) then $I = [5i-1, 5i]$.
2. If only I_1 is empty, $I = [d-\varepsilon, d+1-\varepsilon]$
3. If only I_2 is empty, $I = [a-1+\varepsilon, a+\varepsilon]$
4. If I_1 and I_2 are not empty then, $I = [\frac{d+a}{2}, \frac{d+a}{2}+1]$, this is an interval of length 1, located halfway between I_1 and I_2 and intersecting all previously presented intervals of this step.

After $2i+1$ intervals there are at least $i+1$ intervals with the same color. In the next step, an interval of bandwidth 1 that intersects these $i+1$ intervals of the same color and does not intersect any other interval is presented. The online algorithm has to color this interval by the second color and therefore does not increase the number of colors. An offline algorithm can color the $i+1$ intervals with $i+1$ colors it used before. The next interval can be colored by a new color $i+2$ since it intersects intervals that are colored with $i+1$ distinct colors.

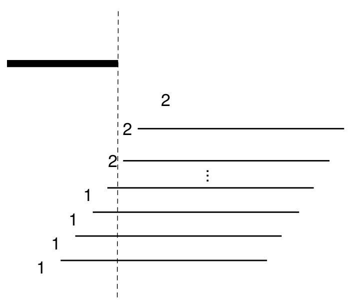

Fig. 3. Possible running of the lower bound for lazy with bandwidth and equal length intervals. The thick interval has bandwidth 1 and the thin intervals have bandwidth ε. Note that in this example the thick interval intersects only intervals with the same color 1.

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. J. Boyar, L. Epstein, L. M. Favrholdt, J. S. Kohrt, K. S. Larsen, M. M. Pedersen, and S. Wøhlk. The maximum resource bin packing problem. manuscript.
3. 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.
4. D. G. Corneil, H. Kim, S. Natarajan, S. Olariu, and A. P. Sprague. Simple linear time recognition of unit interval graphs. *Information Processing Letters*, 55(2):99–104, 1995.
5. T. Erlebach and F. C. R. Spieksma. Interval selection: applications, algorithms, and lower bounds. *J. Algorithms*, 46(1):27–53, 2003.
6. T. R. Jensen and B. Toft. *Graph coloring problems.* Wiley, 1995.
7. H. A. Kierstead. The linearity of first-fit coloring of interval graphs. *SIAM Journal on Discrete Mathematics*, 1(4):526–530, 1988.
8. H. A. Kierstead and J. Qin. Coloring interval graphs with First-Fit. *SIAM Journal on Discrete Mathematics*, 8:47–57, 1995.
9. H. A. Kierstead and W. T. Trotter. An extremal problem in recursive combinatorics. *Congressus Numerantium*, 33:143–153, 1981.
10. N. S. Narayanaswamy. Dynamic storage allocation and online colouring interval graphs. In *Proc of the 10th Annual International Conference on Computing and Combinatorics (COCOON2004)*, pages 329–338, 2004.
11. 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.
12. I. Rabinovitch. The scott-suppes theorem on semiorders. *J. Math. Psych.*, 15:209–212, 1977.
13. D. Scott and P. Suppes. Foundational aspects of theories of measurement. *J. Symbolic Logic*, 23:113–128, 1958.
14. S. S. Seiden. On the online bin packing problem. *Journal of the ACM*, 49(5):640–671, 2002.
15. D. Simchi-Levi. New worst-case results for the bin-packing problem. *Naval Res. Logist.*, 41(4):579–585, 1994.

Acknowledgment. We would like to thank an anonymous referee for pointing out an error in an earlier version of this paper.