

Online Selection of Intervals and t -Intervals*

Unnar Th. Bachmann[†] Magnús M. Halldórsson^{†‡} Hadas Shachnai[§]

Abstract

A t -interval is a union of at most t half-open intervals on the real line. An interval is the special case where $t = 1$. We consider the problems of online selection of intervals and t -intervals, which show up in Video-on-Demand services, high speed networks and molecular biology, among others. We derive lower bounds and (almost) matching upper bounds on the competitive ratios of randomized algorithms for selecting intervals, 2-intervals and t -intervals, for any $t > 2$. While offline t -interval selection has been studied before, the online version is considered here for the first time.

1 Introduction

Interval scheduling is a form of a resource allocation problem, in which the machines are the resource. As argued by Kolen et al. [16], operations management has undergone a “transition in the last decennia from resource oriented logistics (where the availability of resources has dictated the planning and completion of jobs) to demand oriented logistics (where the jobs and their completion are more or less fixed and the appropriate resources must be found).” The problem of allocating resources to such jobs, for which we know a-priori the start and completion times, can be viewed as *interval scheduling*.

Suppose you are running a resource online. Customers call and request to use it from time to time, for up to t time periods, not necessarily of equal length. These requests must either be accepted or declined. If a request is accepted then it *occupies* the resource for these periods of time. A request cannot be accepted if one or more of its periods intersect a period of a previously accepted request. The goal is to accept as many requests as possible.¹

This can be modeled as the following *online t -interval selection problem (t -ISP)*, where t is the maximum number of periods involved in any request. Each request is represented by a t -interval, namely, a union of at most t half-open intervals (or *segments*) on the real line. The t -intervals arrive one by one (not necessarily in order of their left endpoints) and need to be scheduled non-preemptively on a single machine. Two t -intervals, I and J , are disjoint if none of their segments intersect, and intersect if a segment of one intersects a segment of the other.

*A preliminary version of this paper appeared in the Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT), Bergen, June 2010.

[†]School of Computer Science, Reykjavik University, 101 Reykjavik, Iceland. {unnar07,mmh}@ru.is.

[‡]Supported by the Icelandic Research Fund (grant 060034022).

[§]Department of Computer Science, Technion, Haifa 32000, Israel. hadas@cs.technion.ac.il. Corresponding author.

¹Demand oriented logistics may either increase the amount of available resources, so as to satisfy *all* the demands, or attempt to satisfy as much demand as possible, given a fixed amount of resources. We adopt the second approach.

Upon arrival of a t -interval, the scheduler needs to decide whether it is accepted; if not, it is lost forever. The goal is to select a maximum cardinality subset (or “form a schedule”) of non-intersecting t -intervals. The special case where $t = 1$ is known as the *online interval selection problem* (ISP). An example of an instance of online t -ISP is given in Figure 1.

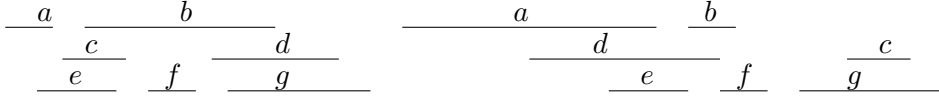


Figure 1: A linear resource is requested by customers a, b, c, d, e, f and g in that order, for two periods each. If b is accepted then each of the following requests must be declined. An optimal selection consists of a, f and g .

The performance of an online algorithm is measured in terms of its competitive ratio. Formally, let OPT be an optimal offline algorithm for the problem. The competitive ratio of A is defined as $\sup_{\sigma} \frac{OPT(\sigma)}{A(\sigma)}$, where σ is an input sequence, and $OPT(\sigma), A(\sigma)$ are the number of t -intervals selected by OPT and A , respectively. For randomized algorithms, we replace $A(\sigma)$ with the expectation $\mathbb{E}[A(\sigma)]$ and define the competitive ratio as $\rho_A = \sup_{\sigma} \frac{OPT(\sigma)}{\mathbb{E}[A(\sigma)]}$. An algorithm with competitive ratio of at most ρ is called ρ -competitive. Let n be the number of intervals in the instance; also, denote by Δ the ratio between the longest and shortest segment lengths.

1.1 Related Work

Selecting intervals and t -intervals We can view the offline t -ISP as the problem of finding a *maximum independent set (IS)* in a t -interval graph. While the problem is polynomially solvable in the special case of interval graphs (see, e.g., [15]), it becomes APX-hard already for $t = 2$ [4]. The paper [4] presents a $2t$ -approximation algorithm for the offline weighted t -ISP. Later works extended the study to the selection of t -intervals with demands, where each t -interval is associated with a set of segments and a demand for machine capacity [5], as well as the study of other optimization problems on t -interval graphs (see, e.g., [7]).

There is a wide literature on the IS problem in various classes of graphs. The online version was considered in [11], where an $\Omega(n)$ -lower bound on the competitive ratios of randomized algorithms was given, even for interval graphs (but not when the interval representation is given). A survey of other works is given in [3].

Online interval selection Lipton and Tomkins [17] considered an online interval selection problem where the intervals have weights proportional to their length, and the intervals arrive by time (i.e., in order of their left endpoints). They showed that a $\theta(\log \Delta)$ -competitive factor is optimal, when Δ is known, and introduced a technique that gives an $O(\log^{1+\epsilon} \Delta)$ -competitive factor when Δ is unknown.

Similar problems have been studied also in the area of call admission. ISP can be viewed as call admission on a line, with the objective of maximizing the number of accepted calls. The paper [2] presents a strongly $\lceil \log N \rceil$ -competitive algorithm for the problem, where N is the number of nodes on the line. This yields an $O(\log \Delta)$ -competitive algorithm for general ISP

instances when Δ is known *a priori*. Our algorithm for 2-ISP (see Section 4.3) can be modified to yield (almost) the same ratio for ISP where Δ is unknown.

Recently, Halldórsson et al. [12] studied the problem of online scheduling with interval conflicts. Specifically, given a ground set of (indexed) items, the input is a collection of conflicts, each containing all the items whose index lies within some interval on the real line. Only one item can survive from each conflict. The goal is to maximize the number (or, total weight) of selected items. The paper presents centralized as well as distributed algorithms for the problem, whose competitive ratios are $O(\log \sigma)$, where σ is the size of the largest conflict, as well as matching lower bounds.

The preemptive version of online interval selection turns out to be significantly easier: Adler and Azar [1] devise a 16-competitive randomized algorithm. In the weighted case, there is still a lower bound of $\Omega(\log \Delta / \log \log \Delta)$ even for randomized algorithms [8], but constant competitiveness becomes possible if the intervals arrive by time, i.e., in order of left endpoints [20, 9]. Another way of easing the task of the algorithm is to assume the instance is *monotone*, i.e., the order of the right endpoints of the intervals coincide with that of the left endpoints [18].

Interval scheduling Numerous results are known about interval scheduling under the objective of minimizing the number of machines, or alternatively, online coloring interval graphs. In particular, a 3-competitive algorithm was given by Kierstead and Trotter [14]. The t -ISP problem bears a resemblance to the JISP problem see, e.g., [19, 10], where each job consists of several intervals and the task is to complete as many jobs as possible. The difference is that in JISP, it suffices to select only one of the possible segments of the job.

1.2 Our Results

We derive the first lower and upper bounds on the competitive ratios of online algorithms for t -ISP and new or improved bounds for ISP. Table 1 summarizes the results for various classes of instances of ISP, 2-ISP and t -ISP. All of the results apply to randomized algorithms against an oblivious adversary. In comparison, proving strong lower bounds for deterministic algorithms (including a lower bound of $\Delta + 1$ for ISP) is straightforward. The upper bounds for general inputs are for the case where Δ is unknown in advance.

	ISP		2-ISP		t -ISP	
	<i>l.b.</i>	<i>u.b.</i>	<i>l.b.</i>	<i>u.b.</i>	<i>l.b.</i>	<i>u.b.</i>
General inputs	$\Omega(n)$	\cdot	\cdot	\cdot	\cdot	$n \dagger$
Unknown Δ	\cdot	$O(\log^{1+\varepsilon} \Delta)$	\cdot	$O(\log^{2+\varepsilon} \Delta)$	\cdot	—
Known Δ	$\Omega(\log \Delta) \ddagger$	$O(\log \Delta) \ddagger$	\cdot	$O(\log^2 \Delta)$	\cdot	—
Two lengths 1 & d	$4 - O(1/\sqrt{d})$	$4 \dagger$	6	16	\cdot	—
Unit length	2	$2 \dagger$	3	$4 \dagger$	$\Omega(t)$	$2t \dagger$
Bounded depth s	$2 - 1/s$	$3/2$ ($s = 2$)	—	—	—	—

Table 1: Results for randomized online interval and t -interval selection. Entries marked with \cdot follow by inference. Entries marked with \dagger are trivially known. Entries marked with \ddagger are due to [2].

2 Technique: Stacking Construction

We use the following technique to derive lower bounds on randomized algorithms for (t) -interval selection. The adversary takes advantage of the fact that it knows the algorithm it is interacting with and that it can foresee the probability with which the algorithm selects any given action, even if it does not know the outcome. The adversary presents intervals on top of each other, or “stacks” them, until some interval is chosen with sufficiently low probability. The adversary uses that to force a desirably poor outcome for the algorithm. This general idea is similar to a lower bounding technique of Awerbuch et al. [2] for call control.

Let R be an ISP-algorithm, and let q, x be parameters, where $0 < x < q$. A (q, x) -stacking construction for R is a collection of q intervals positioned on the real line with their left endpoints x/q apart, that are staggered towards the left. Formally, consider the unit intervals I_1, \dots, I_q , where $I_i = [x(1 - i/q), 1 + x(1 - i/q))$, for $1 \leq i \leq q$. Let p_i be the (unconditional) probability that R selects I_i . The adversary knows the values p_i and forms its construction accordingly. Let m be the smallest value such that $p_m \leq 1/q$; it exists since $\sum_i p_i \leq 1$. The input sequence construction consists of $\mathcal{I} = \langle I_1, I_2, \dots, I_m, J_m \rangle$, where $J_m = [1 + x(1 - m/q), 2 + x(1 - m/q))$. This is illustrated in Figure 2.

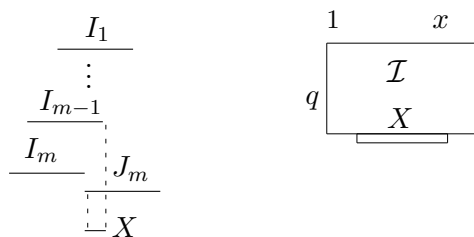


Figure 2: A (q, x) -stacking construction (left). A symbolic picture for a (q, x) -stacking construction with unit intervals \mathcal{I} ; the '1' in the top left corner indicates that the construction consists of unit intervals (right).

Let $\mathbb{E}_R[\mathcal{I}]$ be the expected size of the solution found by R on an input sequence \mathcal{I} of intervals. Further, for any interval $I \in \mathcal{I}$ in the input, let $\mathbb{E}_R[I : \mathcal{I}]$ be the expected contribution of I to the solution size $\mathbb{E}_R[\mathcal{I}]$, and for a subsequence $\mathcal{I}' \subseteq \mathcal{I}$, let $\mathbb{E}_R[\mathcal{I}' : \mathcal{I}] = \sum_{I \in \mathcal{I}'} \mathbb{E}_R[I : \mathcal{I}]$. Observe that if \mathcal{I}' and \mathcal{I}'' partition \mathcal{I} , then $\mathbb{E}_R[\mathcal{I}] = \mathbb{E}_R[\mathcal{I}' : \mathcal{I}] + \mathbb{E}_R[\mathcal{I}'' : \mathcal{I}]$.

Observation 2.1 *A (q, x) -stacking construction \mathcal{I} has the following properties.*

1. All intervals in $\mathcal{I} \setminus \{I_m\}$ overlap the segment $[1, 1 + x)$.
2. All intervals in \mathcal{I} are contained within the interval $[0, 2 + x)$.
3. The intervals in $\mathcal{I} \setminus \{I_m\}$ have a common intersection of length x/q , given by the segment $X = I_{m-1} \cap J_m = [1 + x(1 - m/q), 1 + x(1 - (m - 1)/q))$.
4. $\mathbb{E}_R[I_m : \mathcal{I}] = p_m \leq 1/q$. Thus, $\mathbb{E}_R[\mathcal{I}] = \mathbb{E}_R[\mathcal{I} \setminus \{I_m\} : \mathcal{I}] + \mathbb{E}_R[I_m : \mathcal{I}] \leq 1 + 1/q$.
5. $OPT(\mathcal{I}) = 2$.

From Observation 2.1 (4) and (5) it follows that the performance ratio of R is at least $2/(1+1/q)$. By taking q arbitrarily large, we obtain the following performance bound.

Theorem 1 *Any randomized online algorithm for ISP with unit intervals has competitive ratio at least 2.*

We may use the stacking construction *shifted* by a displacement f , by adding f to the starting point of each interval. We may also use intervals of non-unit length. In basic usage, the parameter x equals the length of the intervals, but it can be reduced if required to fit within a given window.

We can apply the stacking construction with 2-intervals by repeating the construction for both segments. We refer to this as a *2-interval* (q, x) -stacking construction.

3 Online Interval Selection

3.1 Unit Intervals and Depth

We give upper and lower bounds on the competitiveness of ISP with unit intervals. We parameterize the problem in terms of the *depth* of the interval system, which is the maximum number of intervals that overlap a common point and equals the clique number of the corresponding interval graph.

Theorem 2 *The competitive ratio of any randomized algorithm for ISP of unit intervals is at least $2 - 1/s$, where s is the depth of the instance.*

Proof: We modify the $(s, 1)$ -stacking construction slightly. Let p_i be the unconditional probability that the given algorithm R selects interval I_i , for $i = 1, 2, \dots, s$. We distinguish between two cases.

- (i) If $p_1 \leq 1/(2 - 1/s) = s/(2s - 1)$, then we conclude the input with the unit sequence $\langle I_1 \rangle$. The performance ratio is then at least $1/p_1 \geq 2 - 1/s$.
- (ii) Otherwise, when $p_1 > s/(2s - 1)$, we stop the sequence at I_m , where $m \leq s$ is the smallest number such that $p_m \leq 1/(2s - 1)$. This number exists since otherwise $\sum_{i=1}^s p_i > p_1 + (s - 1)/(2s - 1) > 1$, which contradicts the fact that the s intervals overlap. As before, the sequence $\langle I_1, \dots, I_m \rangle$ is followed by the interval J_m , intersecting only the first $m - 1$ intervals. The algorithm obtains expected value at most $1 + p_m \leq 1 + 1/(2s - 1) = 2s/(2s - 1)$, while the optimal solution value is 2, for a ratio of at least $2/(2s/(2s - 1)) = 2 - 1/s$.

The above procedure can be repeated arbitrarily often, ensuring that the lower bound holds also in the asymptotic case. ■

We now describe a randomized algorithm *RoG* (*Random_or_Greedy*) that achieves the ratio in Theorem 2 for $s = 2$. The algorithm handles each arriving interval I with the following rule: If I does not overlap any previously presented interval, select I with probability $2/3$, and otherwise select it greedily.

Theorem 3 *Algorithm *RoG* is $3/2$ -competitive for unit intervals with depth 2.*

Proof: Consider any connected component separately. The depth restriction means that each interval can intersect at most two other intervals: one from the left and one from the right. The instance is therefore a chain of unit intervals. We divide the intervals into three types, based on the number of previous intervals the given interval intersects. A type- i interval, for $i = 0, 1, 2$,

intersects i previously presented intervals. Two type-2 intervals cannot intersect, as otherwise the one that appears earlier will have degree 3, leading to depth at least 3. Therefore, the instance consists of chains of type-0 and type-1 intervals attached together by type-2 intervals. Each chain is started by a type-0 interval, followed by type-1 intervals. Let n_i denote the number of intervals of type i , we then have that

$$n_0 \geq n_2 + 1 . \tag{1}$$

Consider now the unconditional probability that intervals of each type are selected, i.e. the probability independent of other selections. The probability of type-0 intervals being selected is $2/3$. The probability of the selection of type-1 intervals alternates between $1/3$ and $2/3$. The expected number of intervals selected by the algorithm is then, using (1), bounded below by

$$\frac{2}{3}n_0 + \frac{1}{3}n_1 \geq \frac{1}{3}(n_0 + n_1 + n_2 + 1) = \frac{n+1}{3} .$$

On the other hand, the number of intervals in an optimal solution is the independence number of the path on n vertices, or $\lceil \frac{n}{2} \rceil \leq \frac{n+1}{2}$. Hence, the competitive ratio is at most $3/2$. ■

3.2 ISP with intervals of two lengths

Consider now ISP instances where the intervals can be of two different lengths, 1 and d . It is easy to argue a 4-competitive algorithm by the Classify-and-Select approach (see, e.g., in [6]): Flip a coin, choosing either the unit intervals or the length- d intervals, and then greedily add intervals of that length only. We find that it is not possible to significantly improve on that very simplistic approach.

Theorem 4 *Any randomized online algorithm for ISP with intervals of two lengths, 1 and d , has performance ratio at least $4 - O(1/\sqrt{d})$.*

Proof: Consider any randomized online ISP algorithm R . Let $q = \lfloor \sqrt{d} \rfloor$.

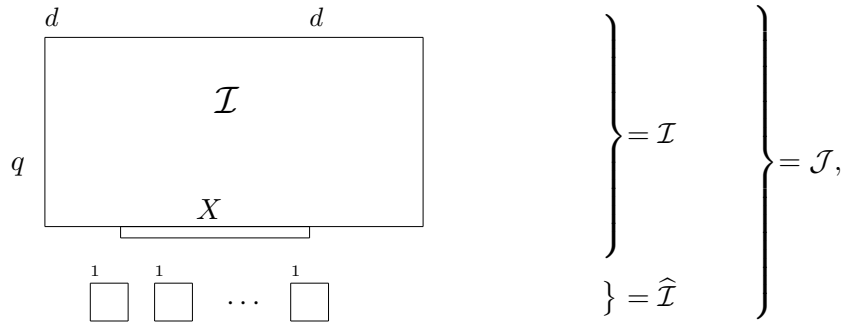


Figure 3: Lower bound for ISP with segments of lengths 1 and $d \gg 1$.

We start with a (q, d) -stacking construction \mathcal{I} for R , using intervals of length d . Recall that, by Observation 2.1 (4), the expected gain of R on interval I_m is $\mathbb{E}_R[I_m : \mathcal{I}] \leq 1/q$. Let p be the

probability that R selects one of the intervals in $\mathcal{I}' = \mathcal{I} \setminus \{I_m\} = \{J_m\} \cup \{I_1, I_2, \dots, I_{m-1}\}$. If $p < 1/2$ then we stop the construction. In that case, the expected solution size found by R is $\mathbb{E}_R[\mathcal{I}] = \mathbb{E}_R[I_m : \mathcal{I}] + \mathbb{E}_R[\mathcal{I}' : \mathcal{I}] \leq p + 1/q$, while the optimal solution is of size 2 (given by I_m and J_m), for a ratio of

$$\frac{2}{p + 1/q} \geq \frac{2}{1/2 + 1/q} = \frac{4}{1 + 2/q} = 4 - O(1/q).$$

Assume then that $p \geq 1/2$. Let X be the common intersection of intervals in \mathcal{I}' , and let f denote the starting point of X . By Observation 2.1 (3), the length of X is $d/q \geq \sqrt{d}$. Let $s = q/3$. We now form a sequence of s disjoint $(q, 1)$ -stacking constructions of unit intervals which, by Observation 2.1 (2), can all be contained within the span of X . Let $\widehat{\mathcal{I}}$ denote the union of these s gadgets, and let $\mathcal{J} = \mathcal{I} \cup \widehat{\mathcal{I}}$. This completes the construction.

Observe that $OPT(\mathcal{J}) \geq 2s$, given by the non-overlapping pairs of the gadgets of $\widehat{\mathcal{I}}$. All intervals in $\widehat{\mathcal{I}}$ overlap X . The expected gain of the algorithm on \mathcal{J} is

$$\mathbb{E}_R[\mathcal{J}] \leq \mathbb{E}_R[\mathcal{I}] + (1-p)\mathbb{E}_R[\widehat{\mathcal{I}}] \leq p + \frac{1}{q} + (1-p)s \left(1 + \frac{1}{q}\right) \leq 1 + \frac{1}{q} + \frac{s}{2} \left(1 + \frac{1}{q}\right) \leq \frac{s}{2} + 2.$$

The last inequality follows from the fact that $p \geq 1/2$. Hence, the performance ratio of R on \mathcal{J} is given by

$$\frac{OPT(\mathcal{J})}{\mathbb{E}_R[\mathcal{J}]} \geq \frac{2s}{s/2 + 2} = 4 - O(1/\sqrt{d}),$$

since $s = \theta(\sqrt{d})$. ■

3.3 ISP with Parameter n

ISP is easily seen to be difficult for a deterministic algorithm on instances without constraints on the length of the intervals. The adversary keeps introducing disjoint intervals until the algorithm selects one of them, I ; the remaining intervals presented will then be contained in I . This leaves the algorithm with a single interval, while the optimal solution contains the rest, for a ratio of $n - 1$. It is less obvious that a linear lower bound holds also for randomized algorithms against oblivious adversary.

Theorem 5 *Any randomized online algorithm for ISP has competitive ratio $\Omega(n)$.*

Proof: We use Yao's principle [21]; namely, we show a lower bound for any deterministic algorithm on a random input sequence. Given an integer $n > 1$, let r_1, r_2, \dots, r_n be a sequence of uniformly random bits. Let $b_i = \sum_{j=1}^{i-1} r_j \cdot 2^{n-j}$. Denote by I_1, \dots, I_n a sequence of intervals, where $I_i = [b_i, b_i + 2^{n-i})$. Observe that the position of I_i depends only on the previous random bits r_1, \dots, r_{i-1} , but not on r_i . The collection $A = \{I_i : r_i = 1\} \cup \{I_n\}$ forms an independent set, informally referred to as the "good" intervals. The set $B = \mathcal{I}_n \setminus A = \{I_i : r_i = 0\}$ forms a clique. Moreover, any interval $I_i \in B$ contains all the intervals I_{i+1}, \dots, I_n . Informally, these are the "bad" intervals.

Consider an algorithm R and the sequence of intervals chosen by R . The event that a chosen interval is good is a Bernoulli trial, and these events are independent. Thus, the number of intervals chosen until a bad one is chosen is a geometric random variable with a mean of 2. Even

accounting for the last interval, which is known to be good, the expected number of accepted intervals $\mathbb{E}[\mathbf{R}(\sigma)]$ is at most 3.

On the other hand, the expected number of good intervals is $(n - 1)/2 + 1$, and so the expected size of the optimal solution is $n/2$. Applying Yao's principle, the competitive ratio of \mathbf{R} on \mathcal{I}_n is at least $n/6$. ■

Observe that the intervals constructed have special properties. For one, they form a *laminar* family of intervals, i.e., they form a *containment* interval graph: whenever two intervals overlap one contains the other. Also, the interval graph is a *split* graph, as the vertex set can be partitioned into an independent set A and a clique B .

We note that, in Theorem 5, the intervals are presented in order of increasing left endpoints. Thus, the bound holds also for the scheduling-by-time model. The adversary in Theorem 5 has also the property of being *transparent* [13] in the sense that as soon as the algorithm has made its decision on an interval, the adversary reveals its own choice.

Corollary 3.1 *There is an $\Omega(n)$ -lower bound on the competitive ratio of any randomized online algorithm for ISP, even on laminar interval systems that induce a split graph. This holds also in the scheduling-by-time model.*

It is instructive to contrast our result with the $\Omega(\log \Delta)$ -lower bound of [2]. Recall that their construction works on a discrete line with N known points. If our construction was placed on this discrete line, it would have $N = \log n$, which does not improve on the $\Omega(\log N)$ -lower bound obtained by [2]. However, their construction potentially involves many more intervals, and thus does not give an $\Omega(n)$ -lower bound. In particular, if the algorithm picked intervals with probability $1/\log n$, their construction yields only an $\Omega(\log n)$ -lower bound.

4 Online 2-Interval Selection

4.1 Unit Segments

In this section we derive a lower bound on the competitive ratio of randomized online algorithms for 2-ISP with unit intervals. Our proof relies on *unit 2-interval stacking construction* for a randomized online 2-ISP algorithm \mathbf{R} , which consists of two steps:

1. *Layout step:* Let h be a parameter, and $h' = 3h/2$. Form a 2-interval $(h', 1)$ -stacking construction $\mathcal{I}_g = \{I_1, \dots, I_{m-1}, I_m, J_m\}$ for \mathbf{R} (see the top half of Figure 4).
2. *Extension step:* Let X_1 be the common intersection of the first segments of the 2-intervals in $\mathcal{I}'_g = \mathcal{I}_g \setminus \{I_m\}$, and X_2 be the common intersection of the second segments. Let f_i denote the starting point of X_i , $i = 1, 2$.
 - (a) Form an (h, x) -stacking construction, \mathcal{I}_{g1} , of 2-intervals shifted by $f_1 - x$, where $x = |X_1| = 1/h'$. The first segments are positioned to overlap X_1 ; the second segments are immaterial as long as they do not intersect any previous intervals. This is shown in the bottom left of Figure 4.
 - (b) Form an identical construction, \mathcal{I}_{g2} , shifted by $f_2 - x$; again, the second segments do not factor in.

We now make several observations about the combined construction $\mathcal{J} = \mathcal{I}_g \cup \mathcal{I}_{g1} \cup \mathcal{I}_{g2}$.

Observation 4.1 *The following holds for the construction \mathcal{J} .*

1. All intervals in \mathcal{I}_{g1} overlap X_1 , and all intervals in \mathcal{I}_{g2} overlap X_2 .
2. $OPT(\mathcal{J}) = 4$, given by the last two 2-intervals in both \mathcal{I}_{g1} and \mathcal{I}_{g2} .
3. Let p be the probability that algorithm R selects some interval in \mathcal{I}'_g . Then, $\mathbb{E}_R[\mathcal{I}_{g1} : \mathcal{I}] \leq (1-p)(1+1/h)$, by Observation 2.1 (3) and item 1 of this observation.
4. The total space occupied by \mathcal{J} , excluding the second segments of 2-intervals in \mathcal{I}_{g1} and \mathcal{I}_{g2} , is of length at most 12, or at most 3 for each of the four disjoint single-segment structures.

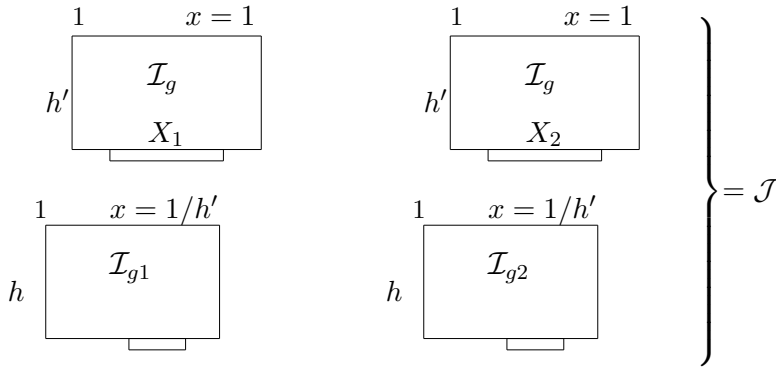


Figure 4: Unit 2-interval stacking construction

Theorem 6 *Any randomized online algorithm for 2-ISP of unit intervals has competitive ratio at least $3 - o(1)$.*

Proof: Consider any randomized online 2-ISP algorithm R . Let h be an even number and $h' = 3h/2$. We start with the *Layout step* to form a 2-interval $(h', 1)$ -stacking construction \mathcal{I}_g . Recall that the expected gain of R on interval I_m is $\mathbb{E}_R[I_m : \mathcal{I}_g] \leq 1/h'$. Let p be the probability that R selects some interval in $\mathcal{I}'_g = \mathcal{I}_g \setminus \{I_m\}$. If $p < 2/3$ then we stop the construction. The expected solution size found by R is then

$$\mathbb{E}_R[\mathcal{I}_g] \leq p + 1/h' , \quad (2)$$

while the optimal solution is of size 2, for a ratio of

$$\frac{2}{p + 1/h'} \geq \frac{2}{2/3 + 2/(3h)} = \frac{3}{1 + 1/h} = 3(1 + o(1)).$$

Assume therefore that $p \geq 2/3$. Let X_1 be the common intersection of the first segments of the 2-intervals in \mathcal{I}'_g , and X_2 be the common intersection of the second segments. Let f_i denote the starting point of X_i , $i = 1, 2$. By Observation 2.1 (3), the length of each X_i is $1/h'$.

We now apply the *Extension step* to complete the unit 2-interval stacking construction. Using Observation 4.1 (3) and Eqn. (2), we have that

$$\begin{aligned}\mathbb{E}_{\mathbf{R}}[\mathcal{J}] &= \mathbb{E}_{\mathbf{R}}[\mathcal{I}_g : \mathcal{J}] + \mathbb{E}_{\mathbf{R}}[\mathcal{I}_{g1} : \mathcal{J}] + \mathbb{E}_{\mathbf{R}}[\mathcal{I}_{g2} : \mathcal{J}] \\ &\leq p + 1/h' + 2(1-p)(1+1/h) \\ &= 2 - p + \left(\frac{8}{3} - 2p\right) \frac{1}{h}.\end{aligned}$$

Since $p \geq \frac{2}{3}$, $\mathbb{E}_{\mathbf{R}}[\mathcal{J}] \leq \frac{4}{3}(1 + \frac{1}{h})$, thus the performance ratio of \mathbf{R} on \mathcal{J} is at least

$$\frac{OPT(\mathcal{J})}{\mathbb{E}_{\mathbf{R}}[\mathcal{J}]} \geq \frac{4}{\frac{4}{3}(1 + \frac{1}{h})} = \frac{3}{1 + \frac{1}{h}} = 3 - o(1), \quad (3)$$

taking the value of h to be sufficiently large and using Observation 4.1 (2). \blacksquare

4.2 Segments of Two Lengths

We treat in this section instances of 2-ISP where the 2-interval segments have lengths either 1 or d , for some $d > 1$. We give a 16-competitive algorithm and an asymptotic lower bound of 6 for the competitive ratio of any online algorithm.

Consider the following algorithm A_v , which either selects (i.e., schedules) a given 2-interval, rejects it, or selects it *virtually*.² A virtually selected interval does not occupy the resource, and will not be a part of the online solution, but it blocks other 2-intervals from being selected. The length of each segment is either *short* (1) or *long* (d). A 2-interval is *short-short* (*long-long*) if both segments are short (long), respectively, and *short-long* if one is short and the other long. In processing a 2-interval I that overlaps with no selected interval, A_v applies the following rules,

1. **I is short-short:** Select I greedily (with probability 1).
2. **A long segment of I intersects a virtually selected 2-interval:** Reject I .
3. **Otherwise:** Select I greedily with probability 1/2 and select it virtually with probability 1/2.

Theorem 7 *Algorithm A_v is 16-competitive for online 2-ISP with segments of length 1 and d .*

Proof: Consider a particular input instance \mathcal{I} and a fixed optimal solution S_{OPT} for \mathcal{I} . For each $J \in S_{OPT}$, let $B(J)$ consist of the intervals in \mathcal{I} with an endpoint in J , and informally refer to it as J 's *bucket*. We now fix a particular interval $J \in S_{OPT}$. We shall show that the expected gain of the algorithm on $B(J)$ is $\mathbb{E}_{A_v}[B(J)] \geq 1/4$; this will imply the theorem, since each interval $I \in \mathcal{I}$ is contained in at most four buckets.

We shall say that an interval is *considered* in a particular execution of the algorithm, if it is not rejected (because it intersects an interval already selected or virtually selected), and thus either selected or virtually selected. Let $S(J) = \{I \in \mathcal{I} : I \cap J \neq \emptyset\}$ denote the set of intervals intersecting J ; besides $B(J)$, it includes the intervals that properly contain a segment of J . For $I' \in S(J)$, let $\sigma_{I'} = \sigma_{I',J}$ denote the event that I' was the first interval in $S(J)$ that was considered. This is well defined, because J itself will be considered, unless it intersects an

²This term has been used before, e.g., in [17].

already (possibly virtually) selected interval that was considered before. Note that $\{\sigma_{I'}\}_{I' \in S(J)}$ partitions the space of all possibilities. Thus,

$$\mathbb{E}_{A_v}[B(J)] = \sum_{I' \in S(J)} \mathbb{E}_{A_v}[B(J)|\sigma_{I'}] \cdot \Pr[\sigma_{I'}].$$

We shall show that the expected gain of the algorithm on $B(J)$ is at least $1/4$, assuming that I' was considered first in $S(J)$, or formally $\mathbb{E}_{A_v}[B(J)|\sigma_{I'}] \geq 1/4$, for any $I' \in S(J)$. This implies the theorem.

Let I' be a particular interval in $S(J)$, and assume $\sigma_{I'}$ holds. We consider some cases depending on the nature of the intersection of I' and J . If I' is in $B(J)$, then I' is selected with probability at least $1/2$, which implies the claim. So, we assume from now that $I' \in S(J) \setminus B(J)$. That means that I' contains a long segment that properly includes a short segment of J . By the definition of the algorithm, I' is virtually selected with probability $1/2$. To establish the claim, it suffices then to show that (assuming that I' was virtually selected, contained in $S(J) \setminus B(J)$ and the first considered interval in $S(J)$), the expected gain on $B(J)$ is at least $1/2$.

We now observe that another interval $I'' \in S(J) \setminus \{I'\}$ gets considered. Namely, if J itself is not considered, it is because it either intersects an already selected interval, or it intersects a virtually selected interval with a long segment. Neither of these cases apply to I' , so there must be yet another interval considered. Let I'' be the first interval in $S(J)$ considered after I' . I'' cannot have a long segment intersecting I' , as otherwise rule 2 of the algorithm would apply. If I'' is in $B(J)$, then I'' is selected with probability at least $1/2$, establishing the claim. So, assume from now that I'' is in $S(J) \setminus B(J)$. Then, both I' and I'' properly contain a different short segment of J . Thus, J is short-short.

With probability $1/2$, I'' is virtually selected. We now claim that some interval in $B(J)$ will be selected with probability 1, which establishes the theorem. Namely, J will be greedily selected, unless it intersects an already selected interval \hat{I} . That interval, \hat{I} , cannot properly include a segment of J , because it would then contain a long segment overlapping either I' or I'' , both of which were virtually selected, and this would conflict with rule 2 of the algorithm. Hence, J is selected unless some other $\hat{I} \in B(J)$ gets selected. This completes the proof of the theorem. \blacksquare

We now give a lower bound of 6 for this problem using (q, d) -stacking constructions of length d 2-intervals, as well as the unit 2-interval stacking construction of Section 4.1.

Theorem 8 *Any randomized online algorithm for 2-ISP with segments of two lengths 1 and d has performance ratio at least $6 - O(d^{-1/4})$.*

Proof: We construct a combination of stacking constructions, built progressively depending on the choices made by the algorithm.

Consider any randomized online 2-ISP algorithm R . Let q be the largest even number, such that $q \leq \sqrt{\lfloor \sqrt{d} \rfloor}$. Furthermore, let $q' = 3q$.

We initially form a (q', d) -stacking construction \mathcal{I} for R with length- d 2-intervals, as illustrated in Figure 5. Recall that $\mathcal{I} = \{I_1, I_2, \dots, I_m, J_m\}$, where m is the smallest number ($m \leq q'$) such that the probability that R chooses I_m satisfies $p_m \leq 1/q'$; thus, the expected gain of the algorithm on that interval is $\mathbb{E}_R[I_m : \mathcal{I}] \leq 1/q'$. Recall that the intervals $\tilde{\mathcal{I}} = \mathcal{I} \setminus \{I_m\} = \{I_1, I_2, \dots, I_{m-1}, J_m\}$ have a common intersection, and let X_1 (X_2) denote their common intersection of the left (right) segments, respectively. Let p be the probability that R selects some 2-interval in $\tilde{\mathcal{I}}$. We distinguish between several cases.

- (i) Suppose that $p < 1/3$. Then, the construction is completed with \mathcal{I} . In this case, the expected solution size found by \mathbf{R} is $\mathbb{E}_{\mathbf{R}}[\mathcal{I}] \leq p + 1/q'$, while the optimal solution is of size 2. The competitive ratio is then

$$\frac{2}{p + 1/q'} \geq \frac{2}{1/3 + 1/(3q)} = \frac{6}{1 + 1/q} \geq 6 - \frac{6}{q} = 6 - O(d^{-1/4}).$$

- (ii) Otherwise, $p \geq 1/3$. In this case, we continue the construction (similar to the proof of Theorem 6) to form $\mathcal{K} = \mathcal{I} \cup \mathcal{I}_1 \cup \mathcal{I}_2$, where \mathcal{I}_1 and \mathcal{I}_2 are separate $(q, d/q')$ -stacking constructions with segments of length d for \mathbf{R} . They are positioned so that the right endpoint of the first left segment of \mathcal{I}_1 (\mathcal{I}_2) is the right endpoint of X_1 (X_2), respectively, while the right segments are located somewhere separate away from \mathcal{I} or the rest of the construction. See Figure 5 for illustration.

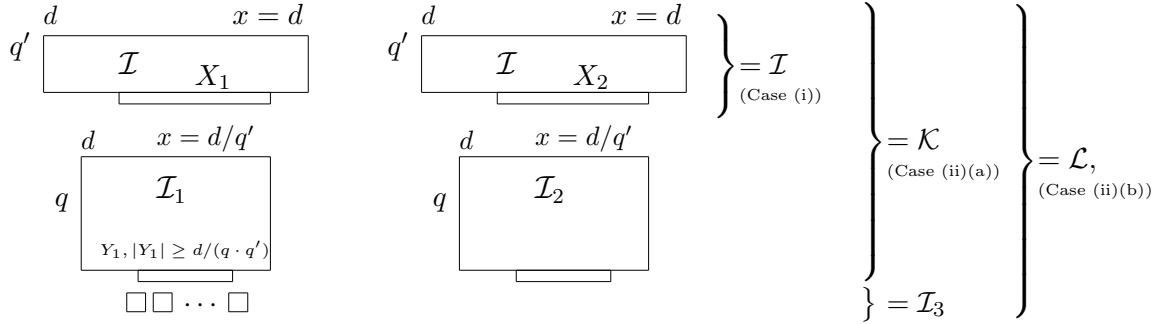


Figure 5: The cases in the proof of Theorem 8. The small boxes represent the gadgets composing \mathcal{I}_3 .

Notice that all intervals in $\mathcal{I}_1 = \{I'_1, \dots, I'_{m'}, J'_{m'}\}$ intersect X_1 , and all intervals in $\mathcal{I}_2 = \{I''_1, \dots, I''_{m''}, J''_{m''}\}$ intersect X_2 , since the extent with which they are shifted is less than d/q' .

Let p' (p'') be the probability that \mathbf{R} selects some 2-interval in $\mathcal{I}_1 \setminus \{I'_{m'}\} = \{I'_1, \dots, I'_{m'-1}, J'_{m'}\}$ ($\mathcal{I}_2 \setminus \{I''_{m''}\}$), conditioned on \mathbf{R} being able to (i.e., not having chosen any 2-interval in $\mathcal{I} \setminus \{I_m\}$), respectively. By symmetry, assume w.l.o.g. that $p' \geq p''$. We further distinguish between two sub-cases.

- (a) If $p + 2p'(1 - p) < 2/3$, then the construction is terminated with \mathcal{K} . Observe that

$$\begin{aligned} \mathbb{E}_{\mathbf{R}}[\mathcal{K}] &= \mathbb{E}_{\mathbf{R}}[\mathcal{I} : \mathcal{K}] + \mathbb{E}_{\mathbf{R}}[\mathcal{I}_1 : \mathcal{K}] + \mathbb{E}_{\mathbf{R}}[\mathcal{I}_2 : \mathcal{K}] \\ &= \mathbb{E}_{\mathbf{R}}[\mathcal{I}] + (1 - p)\mathbb{E}_{\mathbf{R}}[\mathcal{I}_1] + (1 - p)\mathbb{E}_{\mathbf{R}}[\mathcal{I}_2] \\ &\leq p + \frac{1}{q'} + 2(1 - p)(p' + \frac{1}{q}) \\ &= p + 2p'(1 - p) + O(1/q), \end{aligned}$$

while $OPT(\mathcal{K}) = 4$. The performance ratio is then

$$\frac{OPT(\mathcal{K})}{\mathbb{E}_{\mathbf{R}}[\mathcal{K}]} = \frac{4}{2/3 + O(1/q)} = 6 - O(1/q).$$

- (b) Otherwise, $p + 2p'(1 - p) \geq 2/3$. We then continue the construction on top of \mathcal{K} . Let Y_1 be the common intersection of X_1 and the intervals in $\mathcal{I}_1 \setminus \{I'_{m'}\}$. The length of Y_1 is at least $\frac{1}{3} \cdot d/q^2$, since

$$|Y_1| = |I'_{m'-1} \cap J'_{m'}| = |[d + \frac{d}{q'}(1 - \frac{m'}{q}), d + \frac{d}{q'}(1 - \frac{m'-1}{q})| = \frac{d}{3q^2}.$$

Within the segment Y_1 , $\lfloor d/(36q^2) \rfloor$ disjoint unit 2-interval stacking constructions (see Figure 4) are formed, taking $h = q$. Each such gadget takes space at most 12 by Observation 4.1 (4), and thus they fit within Y_1 .

Let \mathcal{I}_3 form the union of the gadgets, and let $\mathcal{L} = \mathcal{K} \cup \mathcal{I}_3$. The probability that the algorithm is in position to select anything from \mathcal{I}_3 is $(1 - p)(1 - p')$. Thus, from (3), its expected gain from \mathcal{I}_3 is given by

$$\mathbb{E}_{\mathbb{R}}[\mathcal{I}_3] \leq (1 - p)(1 - p') \frac{OPT(\mathcal{I}_3)(1 + 1/q)}{3}. \quad (4)$$

Note that the lower bounds on p and $2p'(1 - p)$ imply that

$$(1 - p)(1 - p') = 1 - p/2 - (p + 2p'(1 - p))/2 \leq 1 - 1/6 - 1/3 = 1/2. \quad (5)$$

Clearly, $E_{\mathbb{R}}[\mathcal{L}] = O(1) + \mathbb{E}_{\mathbb{R}}[\mathcal{I}_3]$, and $OPT(\mathcal{L}) \geq OPT(\mathcal{I}_3)$; thus, using (4) and (5), we have that

$$\begin{aligned} \frac{OPT(\mathcal{L})}{\mathbb{E}_{\mathbb{R}}[\mathcal{L}]} &\geq \frac{OPT(\mathcal{I}_3)}{\mathbb{E}_{\mathbb{R}}[\mathcal{I}_3] + O(1)} \\ &\geq \frac{OPT(\mathcal{I}_3)}{\frac{(1 - p)(1 - p')OPT(\mathcal{I}_3)(1 + 1/q)}{3} + O(1)} \\ &\geq \frac{3(1 + O(1/q))}{(1 - p)(1 - p')} \\ &\geq 6 - O(1/q). \end{aligned}$$

■

4.3 Segments of Arbitrary Lengths

We now treat instances with intervals of arbitrary lengths. We do so by partitioning the intervals into groups, where each group consist of intervals with segments of roughly equal length.

We first apply algorithm A_v to 2-ISP instances where the length of the short segment is in $[1, 2)$ and the long segment in $[d, 2d)$. A_v makes scheduling decisions as before, using the new definitions of ‘short’ and ‘long’ segments.

Theorem 9 *Algorithm A_v is 24-competitive for 2-ISP instances with segments of two types: short with lengths in $[1, 2)$, and long with lengths in $[d, 2d)$, where $d \geq 1$.*

Proof: Each interval I now intersects at most six intervals in S_{OPT} that it does not dominate. For instance, a long segment can now contain one long segment from S_{OPT} and properly overlap two other segments. $B(J)$ can contain intervals properly containing J that are of the same length class, but at most one per segment. So, an interval can be contained in at most 6 buckets. The rest of the proof of Theorem 7 is unchanged. ■

Consider next more general instances of 2-ISP, in which the ratio between the longest and shortest segment is Δ , for some $\Delta > 1$. W.l.o.g. we may assume that the short segment is of length 1. We partition the set of first segments to $K = \lceil \log \Delta \rceil$ groups, such that the segments in group i have lengths in $[2^{i-1}, 2^i)$, $1 \leq i \leq K$. Partition the second segments similarly into K groups. A 2-interval whose first segment is of length in $[2^{i-1}, 2^i)$, and whose second segment is of length $[2^{j-1}, 2^j)$, $1 \leq i, j \leq K$, is in group (i, j) .

Given a general instance of 2-ISP, suppose that Δ is known *a priori*. Consider algorithm A_{vg} which applies A_v on groups of 2-intervals. The instance is partitioned into $K^2 = \lceil \log \Delta \rceil^2$ groups, depending on the lengths of the first and second segments of each 2-interval. A_{vg} selects uniformly at random a group (i, j) , $1 \leq i, j \leq K$ and considers scheduling only 2-intervals in this group. All other 2-intervals are declined. The next result follows from Theorem 9.

Theorem 10 A_{vg} is $O(\log^2 \Delta)$ -competitive for 2-ISP with intervals of various lengths, where Δ is known in advance.

For the case where Δ is *a priori unknown*, we apply a technique of Lipton and Tomkins [17] to form algorithm \tilde{A}_{vg} , which produces groups as follows. Each group is identified with the first interval assigned to the group. A presented 2-interval, I , belongs to a group marked by a first interval I' if, the ratio between the lengths of the first segments, as well as the ratio between the lengths of the second segments, is between 1 and 2. If I can belong to more than one group, it is assigned to one arbitrarily. If it does not belong to any group, a new group is created for I .

Each group can then be indexed by $i \in \{1, \dots, \lceil \log \Delta \rceil^2\}$. The algorithm chooses randomly at most one group from the countably infinite set of groups, and selects only 2-intervals from that group, using algorithm A_v . For $i \geq 1$ and $\epsilon > 0$, define

$$c_i = \frac{1}{\zeta(1 + \epsilon/2) \cdot i^{1+\epsilon/2}}, \quad \text{and} \quad p_i = \frac{c_i}{\prod_{j=1}^{i-1} (1 - p_j)}, \quad (6)$$

where $\zeta(x) = \sum_{r=1}^{\infty} r^{-x}$ is the Riemann zeta function.³ Recall that $\zeta(x) < \infty$, for $x > 1$.

If a given 2-interval belongs to a new group i , and none of the groups $1, 2, \dots, i-1$ has been selected, then group i is chosen with probability p_i and rejected with probability $1 - p_i$. If a given 2-interval belongs to an already selected group i , it is selected using algorithm A_v ; if the given 2-interval belongs to an already rejected group then it is rejected. Note that by the definition of p_i , as given in (6), it follows that c_i is the unconditional probability that \tilde{A}_{vg} chooses the i -th group.

In analyzing \tilde{A}_{vg} we first show that the values p_i form valid probabilities, and that the c_i values give a probability distribution.

Lemma 11 $\sum_{i=1}^{\infty} c_i = 1$. Also, $p_i \leq 1$, for all $i \geq 1$.

³Lipton and Tomkins made similar use of the zeta function for online interval scheduling [17].

Proof: Observe that $\sum_{i=1}^{\infty} c_i = \frac{1}{\zeta(1+\epsilon/2)} \sum_{i=1}^{\infty} \frac{1}{i^{1+\epsilon/2}} = 1$, proving the first half of the lemma. It follows that $c_i \leq 1 - \sum_{j=1}^{i-1} c_j$. To prove the second half of the lemma, it suffices to show that

$$p_i = \frac{c_i}{1 - \sum_{j=1}^{i-1} c_j}, \quad (7)$$

for each $i \geq 1$. We prove (7) by induction on i . The base case holds since $p_1 = c_1$. Suppose now that (7) holds for $i = k - 1$. Then, using (6) we have that

$$p_k = \frac{c_k}{c_{k-1}} \cdot \frac{p_{k-1}}{1 - p_{k-1}}.$$

Plugging in the value of p_{k-1} from (7) we get the equation in (7) for $i = k$. \blacksquare

The value ϵ is a parameter to the algorithm. The smaller it gets, the larger the coefficient hidden in the big-oh notation.

Theorem 12 \tilde{A}_{vg} is $O(\log^{2+\epsilon} \Delta)$ -competitive for 2-ISP with intervals of various lengths, where Δ is unknown in advance.

Proof: Let S_i denote the set of 2-intervals in group i , $1 \leq i \leq \log^2 \Delta$. The probability that \tilde{A}_{vg} chooses any given group S_i is at least $c_{\log^2 \Delta}$. After choosing the group, \tilde{A}_{vg} uses A_v to select the 2-intervals in the group. For a given group, S_i , we have:

$$\mathbb{E}[\tilde{A}_{vg}(S_i)] \geq c_{\log^2 \Delta} \cdot \mathbb{E}[A_v(S_i)] \geq \frac{1}{\zeta(1 + \epsilon/2)(\log \Delta)^{2+\epsilon}} \cdot \frac{1}{24} \cdot \mathbb{E}[OPT(S_i)],$$

applying Theorem 9. Thus, by linearity of expectation, \tilde{A}_{vg} is $O(\log^{2+\epsilon} \Delta)$ -competitive. Observe that $\zeta(1 + \epsilon/2) \leq \frac{2}{\epsilon}$, and thus the competitiveness can be bounded by $O(\epsilon^{-1} \log^{2+\epsilon} \Delta)$, as a function of both Δ and ϵ . \blacksquare

5 Online t -Interval Selection

In this section we show that any online algorithm for t -ISP has competitive ratio $\Omega(t)$. Our approach, which uses reduction to a known problem, is standard in the offline setting, but rather unusual in the online case. We reduce from the online version of the independent set (IS) problem in graphs: Given the vertices of a graph one by one, along with edges to previous vertices, determine for each vertex whether to add it to a set of independent vertices.

Theorem 13 Any randomized online algorithm for t -ISP with unit segments has competitive ratio $\Omega(t)$.

Proof: Let n be a positive integer. We show that any graph on n vertices, presented vertex by vertex, can be converted on-the-fly to an n -interval representation with unit segments. Then, an $f(t)$ -competitive online algorithm for t -ISP applied to the n -interval representation yields an $f(n)$ -competitive algorithm for the independent set problem. As shown in [11] (which follows also from Theorem 5 of the current paper), there is no cn -competitive algorithm for the online IS problem, for any fixed $0 < c < 1$. The theorem then follows.

Let $G = (V, E)$ be a graph on n vertices with vertex sequence $\langle v_1, v_2, \dots, v_n \rangle$. Given a vertex v_i and the induced subgraph $G[\langle v_1, v_2, \dots, v_i \rangle]$, form the n -interval I_i by

$$I_i = \bigcup_{j=1}^n X_{ij}, \quad \text{where} \quad X_{ij} = \begin{cases} [nj + i, nj + i + 1) & \text{if } j < i \text{ and } (i, j) \in E \\ [ni + j, ni + j + 1) & \text{otherwise.} \end{cases}$$

It is not hard to verify that $I_i \cap I_j \neq \emptyset$ iff $(i, j) \in E$. Hence, solutions to the t -ISP instance are in one-one correspondence with independent sets in G . ■

A greedy selection of t -intervals yields a $2t$ -competitive algorithm for unit t -ISP, implying that the bound above is tight.

6 Conclusions

We have given tight bounds on the competitive ratios of randomized algorithms for online interval selection with different assumptions on interval lengths. An obvious open question is to close the gaps for 2-intervals. The case of unit 2-intervals is of particular interest. Finally, our algorithms and the stacking technique apply for intervals with the same (unit) weights. It is natural to consider ISP and its variants with arbitrary interval weights.

Acknowledgment. We thank the anonymous referees for many helpful comments on the paper.

References

- [1] R. Adler and Y. Azar. Beating the logarithmic lower bound: Randomized preemptive disjoint paths and call control algorithms. *J. Scheduling*, 6(2):113–129, 2003.
- [2] B. Awerbuch, Y. Bartal, A. Fiat, and A. Rosen. Competitive non-preemptive call control. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 312–320, 1994.
- [3] U. T. Bachmann. *Online t -Interval Scheduling*. M.Sc. thesis, School of Computer Science, Reykjavik University, Dec. 2009.
- [4] R. Bar-Yehuda, M. M. Halldórsson, J. Naor, H. Shachnai, and I. Shapira. Scheduling split intervals. *SIAM J. Comput.*, 36(1):1–15, 2006.
- [5] R. Bar-Yehuda and D. Rawitz. Using fractional primal-dual to schedule split intervals with demands. *Discrete Optimization*, 3(4):275 – 287, 2006.
- [6] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.
- [7] A. Butman, D. Hermelin, M. Lewenstein, and D. Rawitz. Optimization problems in multiple-interval graphs. *ACM Transactions on Algorithms*, 6(2), 2010.
- [8] R. Canetti and S. Irani. Bounding the power of preemption in randomized scheduling. *SIAM J. Comput.*, 27(4):993–1015, 1998.

- [9] L. Epstein and A. Levin. Improved randomized results for the interval selection problem. *Theor. Comput. Sci.*, 411(34-36):3129–3135, 2010.
- [10] T. Erlebach and F. C. R. Spieksma. Interval selection: Applications, algorithms, and lower bounds. *J. Algorithms*, 46(1):27–53, 2003.
- [11] M. M. Halldórsson, K. Iwama, S. Miyazaki, and S. Taketomi. Online independent sets. *Theoretical Computer Science*, 289(2):953 – 962, 2002.
- [12] M. M. Halldórsson, B. Patt-Shamir, and D. Rawitz. Online scheduling with interval conflicts. In *Proc. of 28th International Symposium on Theoretical Aspects of Computer Science, STACS 2011, LIPIcs # 9*, pages 472–483, 2011.
- [13] M. M. Halldórsson and M. Szegedy. Lower bounds for on-line graph coloring. *Theoretical Comput. Sci.*, 130:163–174, Aug. 1994.
- [14] H. A. Kierstead and W. T. Trotter. An extremal problem in recursive combinatorics. In *Congr. Numer. 33*, pages 143–153, 1981.
- [15] J. Kleinberg and E. Tardos. *Algorithm Design*. Addison Wesley, 2005.
- [16] A. W. Kolen, J. K. Lenstra, C. H. Papadimitriou, and F. C. Spieksma. Interval scheduling: A survey. *Naval Research Logistics*, 54:530–543, 2007.
- [17] R. J. Lipton and A. Tomkins. Online interval scheduling. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 302–311, 1994.
- [18] H. Miyazawa and T. Erlebach. An improved randomized on-line algorithm for a weighted interval selection problem. *J. of Scheduling*, 7(4):293–311, 2004.
- [19] F. Spieksma. On the approximability of an interval scheduling problem. *J. Sched.*, 2:215–227, 1999.
- [20] G. J. Woeginger. On-line scheduling of jobs with fixed start and end times. *Theor. Comput. Sci.*, 130(1):5–16, 1994.
- [21] A. C.-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 222–227, Washington, DC, USA, 1977. IEEE Computer Society.