Beating the logarithmic lower bound: randomized preemptive disjoint paths and call control algorithms

Ran Adler *   Yossi Azar †

July 4, 1998

Abstract

We consider the *maximum disjoint paths* problem and its generalization, the *call control* problem, in the on-line setting. In the *maximum disjoint paths* problem, we are given a sequence of connection requests for some communication network. Each request consists of a pair of nodes, that wish to communicate over a path in the network. The request has to be immediately connected or rejected, and the goal is to maximize the number of connected pairs, such that no two paths share an edge. In the *call control* problem, each request has an additional bandwidth specification, and the goal is to maximize the total bandwidth of the connected pairs (throughput), while satisfying the bandwidth constraints (assuming each edge has unit capacity). These classical problems are central in routing and admission control in high speed networks and in optical networks.

We present the first known constant-competitive algorithms for both problems on the line. This settles an open problem of Garay et al. and of Leonardi. Moreover, to the best of our knowledge, all previous algorithms for any of these problems, are $\Omega(\log n)$-competitive, where n is the number of vertices in the network (and obviously non-competitive for the continuous line). Our algorithms are randomized and preemptive. Our results should be contrasted with the $\Omega(\log n)$ lower bounds for deterministic preemptive algorithms of Garay et al. and the $\Omega(\log n)$ lower bounds for randomized non-preemptive algorithms of Lipton and Tomkins and Awerbuch et al. Also, non-constant lower bounds proved by Canetti and Irani for randomized preemptive algorithms for related problems emphasize the distinction between problems that have constant competitive algorithms and other that do not.

*Department of Computer Science, Tel Aviv University, Tel-Aviv 69978, Israel. E-mail: adleran@math.tau.ac.il.

†Department of Computer Science, Tel Aviv University, Tel-Aviv 69978, Israel. Research supported in part by the Israel Science Foundation and by the United States-Israel Binational Science Foundation (BSF). E-mail: azar@math.tau.ac.il.

1 Introduction

We consider the *maximum disjoint paths* problem and its generalization, the *call control* problem, in the on-line setting. In the *maximum disjoint paths* problem, we are given a sequence of connection requests for some communication network. Each request consists of a pair of nodes, that wish to communicate over a path in the network. The request has to be immediately connected or rejected, and the goal is to maximize the number of connected pairs, such that no two paths share an edge. In the *call control* problem, each request has additional *bandwidth* and *benefit* specifications (the benefit is usually proportional to the bandwidth). The goal is to maximize the total benefit of the connected pairs while satisfying the bandwidth constraints (assuming each edge has unit capacity).

These classical problems were extensively studied in recent years, since they are applicable to routing and admission control in high speed networks [2, 5, 7] and optical networks [1, 3, 4, 17].

The algorithms we consider are also *preemptive*, that is, they may, at any point of time, decide to stop an on-going call in the network. Of course, if a call is preempted, then it's benefit is not accounted in the total benefit.

We focus on the case where the benefit is proportional to the bandwidth. This corresponds to maximizing the total throughput of the network. Also, we consider only the case where the network is a line, and thus the requested paths are intervals.

The performance of the online algorithm is measured in terms of its approximation ratio, called the competitive ratio. A deterministic or randomized algorithm is defined to be c-competitive, if for any sequence of requests its (expected) benefit is no less than c times the benefit of the optimal offline algorithm.

Our results. We present the first known constant-competitive algorithms for the maximum disjoint paths problem and for the call control problem on the line. This settles an open problem of [11, 14]. Moreover, to the best of our knowledge, all previous algorithms for any of these problems are $\Omega(\log n)$-competitive, where n is the number of vertices in the network (and obviously non-competitive for the continuous line). Constant approximation ratios were achieved only in off-line settings (see e.g [12, 13] and their references). Our algorithms are randomized and preemptive. Our results should be contrasted with the $\Omega(\log n)$ lower bound for deterministic preemptive algorithms in [11], and the $\Omega(\log n)$ lower bound for randomized non-preemptive algorithms [5, 6, 16]. Also, non-constant lower bounds were proved in [10] for randomized preemptive algorithms in various cases. However, these lower bounds do not apply to the standard disjoint paths and call control problems.

In the way of constructing our algorithm for the disjoint paths problem, we first design a 4-competitive *deterministic* algorithm for requests of bandwidth $1/2$. This algorithm is used for establishing the randomized disjoint paths algorithm. Some techniques of [4] are used for transforming the above deterministic algorithm to an algorithm for requests of bandwidth $1/k$ for $k > 1$.

It is important to mention that our randomized algorithm for the disjoint paths problem does not suffer from the known undesired property of many randomized on-line call control algorithms, that high benefit is attained only with very poor probability (see discussion in [15]). In fact, not only that our algorithm succeeds with constant probability, but the number of paths that the algorithm provides is quite concentrated around its mean.

For constructing the general call control algorithm, we first design a constant-competitive deterministic algorithm for requests of arbitrary bandwidth limited by $\delta < 1/2$. A crucial ingredient of this algorithm is to ignore "stuffed intervals" - intervals that contain a large mass of previous intervals. Then, we easily combine this algorithm with the disjoint paths algorithm using randomization and establish the final algorithm.

We note that our algorithms are not memoryless, that is, the decision to accept or reject a new call depends not only on the currently active calls, but also on previously rejected calls. Attempting to remove this dependency, by modifying the algorithm in some natural ways, can be shown to result in non constant-competitive algorithms. It seems very interesting to find out whether there exist memoryless constant-competitive algorithms.

The constants of most of our algorithms are not large (although we make no specific attempt to make them small). We also prove a lower bound of 2 for deterministic or randomized algorithms for all the problems that we consider.

We note that our techniques can be easily applied to optical networks, that is, we can provide constant throughput competitive algorithm for one or more wavelengths in the line network.

Related work. The disjoint paths problem was considered by Garay et al. [11]. They showed an $O(\log n)$-competitive deterministic preemptive algorithm for the line network. They also showed that no deterministic preemptive algorithm can achieve a better competitive ratio. Randomized non-preemptive algorithms for the line network were considered in [6, 16]. They showed an $O(\log n)$-competitive algorithm and a matching lower bound for randomized non-preemptive algorithms. The randomized non-preemptive lower bound holds also for the call control problem, even when requests are limited to a small fraction of the available bandwidth. Note that for general networks one can achieve logarithmic competitive ratios by deterministic algorithms for requests of small bandwidth [5], while no poly-logarithmic competitive ratio can be achieved for requests of full bandwidth even by randomized algorithms [9]. For special networks, e.g., trees, meshes, classes of planner graphs [6, 7, 13] it is possible to design logarithmic competitive algorithms for the call control problem without limiting the requested bandwidth. Nevertheless, we are not aware of any constant-competitive algorithm for disjoint paths problems or call control problem.

Also, some work was done for different measures of benefit. For the disjoint paths problem on the line, [11] considered the case where the benefit of an interval is equal to its length. Here constant-competitive ratio is achieved by deterministic preemptive algorithms. For the call control problem on the line, [8] considered the case where the benefit of a call equals the product of its length and its bandwidth. Here again, a constant-competitive ratio can be achieved by deterministic preemptive algorithms, with the additional constraint that the requested bandwidths are limited to $\delta < 1$. They also showed that deterministic algorithms have very poor competitive ratio on the line, if a call may request the entire bandwidth (that is, $\delta = 1$). For general benefits, [10] showed that even with randomized preemptive algorithms, one cannot achieve a constant competitive ratio even on the line. More specifically, they showed $\Omega(\sqrt{\log \mu / \log \log \mu})$ lower bounds for randomized preemptive algorithms, where μ is the maximum among various variances in the parameters of different calls. Fortunately, the lower bounds are not applicable to our problems.

Structure of the paper. In section 2 we present some definitions. In section 3 we describe a 4-competitive algorithm for requests of bandwidth 1/2. Then, in section 4 we show how to transform it to a randomized algorithm for the disjoint paths problem. In section 5 we transform the algorithm of section 3 to an algorithm for requests of bandwidth $1/k$. In section 6 we design the general algorithm for call control for any requested bandwidth, where we start by showing a deterministic algorithm for requests of bandwidth less than 1/2. Some of the proofs are deferred to the appendix.

2 Preliminaries

We consider a network G which is a line, i.e. consists of chain of links. We denote the sequence of call requests by $\sigma = \sigma_1, \sigma_2..\sigma_k$. Call request i is characterized by a pair: (I_i, r_i), where I_i is the requested path and r_i is the requested bandwidth The requested bandwidth is assumed to satisfy $0 < r_i \leq 1$.

A *valid* set of calls is a set of calls $C \subseteq \sigma$, which satisfies the bandwidth constraints for each of the links, that is:

$$\forall e \in E(G) \quad \sum_{\{\sigma_i \in C \ \mid \ e \in I_i\}} r_i \leq 1$$

We focus on the case where the benefit of a call is proportional to its bandwidth. Thus maximizing the total benefit corresponds to maximizing the total throughput of the network. For any set of calls C, we denote by $B(C)$ the total benefit of the calls (which is equal to the total bandwidths of the calls).

The performance of the online algorithm is measured in terms of its competitive ratio, defined as follows: let $\mathcal{OPT}_\sigma$ be an optimal valid set for the given request sequence, and let $\mathcal{ON}_\sigma$ be the valid set of calls produced by the online algorithm. Then randomized $\mathcal{ON}$ is ρ-competitive if for all sequences σ we have $E(B(\mathcal{ON}_\sigma)) \geq \frac{1}{\rho} B(\mathcal{OPT}_\sigma)$.

We denote the set of the first i requests by S_i , and denote by $\mathcal{A}_i$ the set of accepted calls just before the arrival of request $i+1$. We also denote $S^* = S_k$, $\mathcal{A}^* = A_k$, where k is the sequence length. We omit the index i from S_i and $\mathcal{A}_i$ when it is clear from the context. When all the requested bandwidths are equal, we write $|C|$ instead of $B(C)$ for a set of calls C.

Since our algorithms and bounds do not depend on the number of links, we may replace the network by a continuous line, and replace each discrete path by an open interval. We denote by $left(I)$ and $right(I)$ the left and right endpoint of an interval I respectively. We will often refer to the calls as intervals, ignoring the attached bandwidth. We will also use interval notations for calls, for example, we abuse the notation and use S_i to denote the first i requested intervals.

For simplicity, we assume that S^* has no identical intervals (if there are, we can extend the containment relation by ordering identical intervals in the order of arrival).

3 A deterministic algorithm for bandwidth $1/2$

In this section we show a constant competitive online algorithm for the case where all requested calls σ_i have $r_i = 1/2$, that is, at most two calls are allowed to overlap for each link. Since all the benefits are equal, we set all of them to 1.

Definition 3.1 *Given a set S of intervals, an interval I is* a middle interval of S *if there are two intervals $I_L, I_R \in S$ such that $left(I_L) \leq left(I) \leq left(I_R) \leq right(I_L) \leq right(I) \leq right(I_R)$.*

Informally, the idea of the algorithm is the following: when there is a "collision" between more than two intervals, none of which contains the other, it preempts the middle interval. Also, we need to reject calls that contain previous calls, even if the previous calls have already been rejected or preempted. Note that the algorithm is not memoryless.

More formally, given a new call request, I, the procedure in figure 1 describes how the algorithm decides whether to accept it or reject it.

Procedure: $BW\frac{1}{2}$

begin

(1) **if** there is a $J \in S$ such that $J \subset I$ **then**

(2) reject I

(3) **elseif** there is a $J \in \mathcal{A}$ such that $J \supset I$

(4) preempt all intervals $J' \in \mathcal{A}$ such that $J' \supset I$

(5) accept I

(6) **elseif** I is a middle interval in $\mathcal{A} \cup \{I\}$ **then**

(7) reject I

(8) **else**

(9) preempt middle intervals $J' \in \mathcal{A} \cup \{I\}$

(10) accept I

(11) **end if**

end

Figure 1: Algorithm for bandwidth 1/2 calls

In order to prove that the algorithm is valid (that is, it maintains a valid set $\mathcal{A}$), we first observe the following immediate fact:

Fact 3.2 *Let S be a set of intervals. If there are no $J_1, J_2 \in S$ such that $J_1 \subset J_2$, then among any 3 intersecting intervals there is a middle interval.*

Lemma 3.3 *At any time, there is no $J_2 \in \mathcal{A}$ such that $J_1 \subset J_2$ and $J_1 \in S$.*

Proof: If J_1 arrives first, when J_2 arrives it fails condition (1), and it is rejected. Otherwise, if $J_2 \in \mathcal{A}$ when J_1 arrives then by condition (3) J_2 is preempted in step (4). ∎

Lemma 3.4 *The set $\mathcal{A}$ is a* valid *set of intervals.*

Proof: By induction on the number of input intervals. Initially the claim holds for $\mathcal{A} = S = \phi$. Assume $\mathcal{A}$ is valid, and now a new interval I arrives. If I is rejected, $\mathcal{A}$ is unchanged. Otherwise, if step (4) is executed, at least one interval $J' \supset I$ is preempted, so that I can be allocated in the evicted bandwidth. Otherwise, step (9) is executed. Let $T = \{J' | J' \cap I \neq \phi\}$. By lemma 3.3 each of the intervals in T intersects exactly one endpoint of I. Thus we can partition T to T_L and T_R, such that $|T_L| \leq 2$ and $|T_R| \leq 2$, since $\mathcal{A}$ is valid. Let $I_R \in T_R$ and $I_L \in T_L$. It follows from lemma 3.3 and lemma 3.2 that $I_R \cap I_L = \phi$, otherwise I would be a middle interval. Thus it is sufficient to show that the allocation of I would not violate the bandwidth limitation on the left endpoint of I, and use the symmetrical claim for the right endpoint. If $|T_L| \leq 1$ then I does not cause violation of the bandwidth constraint on it's left side. Otherwise, let J_1 and J_2 denote the intersecting intervals. By lemma 3.3 and lemma 3.2 one of the intervals J_1 and J_2 is a middle interval, it meets the condition of (9), and it is preempted. Thus, I can be allocated in the evicted bandwidth. ■

Corollary 3.5

1. *At most 2 intervals are preempted when step* (4) *is executed.*
2. *At most 2 intervals are preempted when step* (9) *is executed.*

Proof: Claim 1 is obvious. Claim 2 follows from the proof above. ■

Let $\mathcal{OPT}_\sigma^{(k)}$ be an optimal solution for σ when $b_i = 1$ and $r_i = \frac{1}{k}$ for all $\sigma_i \in \sigma$.

Lemma 3.6 $|\mathcal{OPT}_\sigma^{(k)}| \leq k|\mathcal{OPT}_\sigma^{(1)}|$.

Proof: We use the fact that the clique number of an interval graph equals the coloring number. Since $\mathcal{OPT}_\sigma^{(k)}$ is a valid set when all $r_i = \frac{1}{k}$, the maximum clique size in $\mathcal{OPT}_\sigma^{(k)}$ is no more than k. Thus $\mathcal{OPT}_\sigma^{(k)}$ can be colored in k colors. Each of the color classes is an independent set of intervals, and one of them has size at least $|\mathcal{OPT}_\sigma^{(k)}|/k$. Now this set is also valid when all $r_i = 1$, resulting in: $|\mathcal{OPT}_\sigma^{(k)}|/k \leq |\mathcal{OPT}_\sigma^{(1)}|$ as claimed. ■

Lemma 3.7 $|BW\frac{1}{2}\ _\sigma| \geq \frac{1}{2}|\mathcal{OPT}_\sigma^{(1)}|$

Proof: Let $\mathcal{OPT}_\sigma^{(1)}$ be an optimal set of intervals for bandwidth equal to 1, as defined above, and $m = |\mathcal{OPT}_\sigma^{(1)}|$. By definition, the set $\mathcal{OPT}_\sigma^{(1)}$ is a set of pairwise disjoint intervals. Let us denote the intervals by $I_1, I_2 .. I_m$. We can also assume that no $I \in \mathcal{OPT}_\sigma^{(1)}$ contains an interval of S. Define the following intervals, referred to as "cells", as follows: cell j for $1 \leq j \leq \lfloor \frac{m}{2} \rfloor$, denoted by C_j, is the interval $(left(I_{2j-1})..right(I_{2j}))$. (See figure 2). If m is odd then we add another cell, $C_{\lfloor \frac{m}{2} \rfloor + 1} = (left(I_m), +\infty)$. We refer to cells $1..\lfloor \frac{m}{2} \rfloor$ as "regular", and to $C_{\lfloor \frac{m}{2} \rfloor + 1}$, if it exists, as the "infinite" cell. The following claim shows that after a certain point of time, cell C_j always contains an interval, more specifically, there will always be an interval $J \in \mathcal{A}$ s.t. $J \subseteq C_j$. The claim completes the proof of the theorem, since the cells are disjoint.

Claim 3.8 *For regular cells, after the intervals I_{2j-1} and I_{2j} have arrived, there is always an interval $J \in \mathcal{A}$, s.t. $J \subseteq C_j$. If an infinite cell exists, after I_m arrives, there is always an interval $J \in \mathcal{A}$ s.t. $J \subseteq C_{\lfloor \frac{m}{2} \rfloor + 1}$.*

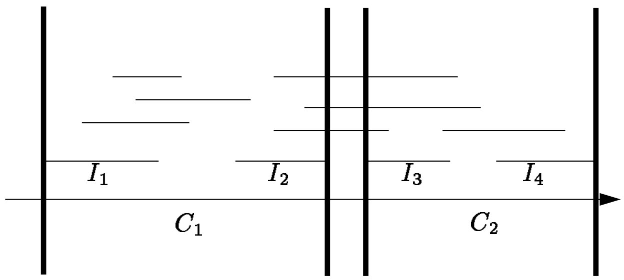

Figure 2: Definition of cells

Proof: First we consider "regular" cells, and prove the claim by induction on the number of intervals that have arrived.

- *Initial step:* Assume I_{2j-1} arrives after I_{2j}. The proof is symmetrical for the other case. If I_{2j-1} is accepted, then the claim holds. Otherwise, since I_{2j-1} contains no other intervals, condition (6) in the algorithm must hold. Let I_R be a right interval, as in definition 3.1. It follows that $left(C_j) = left(I_{2j-1}) \leq left(I_R)$. Since I_R intersects I_{2j-1} and I_{2j-1} lies to the left of I_{2j}, $left(I_R) \leq left(I_{2j})$. Since $I_R \in \mathcal{A}$, by lemma 3.3 it does not contain I_{2j}, and thus $right(I_R) \leq right(I_{2j}) = right(C_j)$. Hence, $I_R \subseteq C_j$ as claimed.

- *Induction:* Assume $J \subseteq C_j$ is preempted when a new interval I arrives. If condition (3) in the algorithm holds, then $I \subset J \subseteq C_j$ satisfies the claim conditions. Otherwise, step (9) in the algorithm is executed, and J is a middle interval. Thus there are intervals $I_L, I_R \in \mathcal{A}$ such that $left(I_L) \leq left(J) \leq left(I_R) \leq right(I_L) \leq right(J) \leq right(I_R)$. Since $J \subseteq C_j$, we get $right(I_L) \leq right(C_j)$ and $left(C_j) \leq left(I_R)$. Assume by contradiction that $left(I_L) < left(C_j)$ and $right(C_j) < right(I_R)$ hold, then we get that $I_L \cup I_R \supseteq C_j \supseteq I_{2j-1} \cup I_{2j}$. Since $I_L \cap I_R \neq \phi$, one of I_L and I_R contains I_{2j-1} or I_{2j}, which is impossible by lemma 3.3. Thus $left(C_j) < left(I_L)$ or $right(I_R) < right(C_j)$, yielding $I_L \subseteq C_j$ or $I_R \subseteq C_j$ respectively, which completes the proof.

Now, for the "infinite" cell the proof is similar, observing the fact that if $J \subseteq C_{\lfloor \frac{m}{2} \rfloor + 1}$ is rejected or preempted, I_R is always contained in the cell. ■

■

Theorem 3.9 $BW\frac{1}{2}$ *is 4-competitive.*

Proof: By lemma 3.7 $|BW\frac{1}{2}{}_{\sigma}| \geq \frac{1}{2}|\mathcal{OPT}_{\sigma}^{(1)}|$. Now by lemma 3.6, $|\mathcal{OPT}_{\sigma}^{(2)}| \leq 2|\mathcal{OPT}_{\sigma}^{(1)}|$. ■

4 A randomized algorithm for bandwidth 1

Next we show how an algorithm for bandwidth 1/2, like $BW\frac{1}{2}$, can be used to construct a randomized constant-competitive algorithm for bandwidth 1. Actually, any deterministic algorithm, with following properties can be used to construct such an algorithm, as is proved by the sequel theorem.

Consider a deterministic preemptive algorithm for call control DET, that maintains a set of intervals D, with the following properties:

- $|DET_\sigma| \geq \frac{1}{c}|OPT_\sigma^{(1)}|$.
- There is a constant d such that any newly accepted interval I intersects at most d other intervals in D (after it has been accepted).

Theorem 4.1 *Any algorithm DET with the above properties can be used to construct a randomized algorithm for bandwidth* 1 *with competitive ratio* $4dc$.

Proof: We construct a randomized algorithm $RAND$. $RAND$ maintains a valid set of intervals (for bandwidth 1) denoted by R. R is initially empty. Let p satisfy $0 \leq p < \frac{1}{2}$. Given a new interval I, we simulate DET on the new interval, and take the actions described in figure 3.

```
begin
(1) preempt from R those intervals that were preempted by DET
(2) if I was rejected then
(3)     reject I
    else
(4)     toss a p-coin
(5)     if coin shows "success" and there is no J ∈ R s.t. J ∩ I ≠ φ then
(6)         accept I
        else
(7)         reject I
        end if
    end if
end
```

Figure 3: A randomized reduction from bandwidth 1 to an algorithm with the above properties

It follows immediately that the algorithm is correct: initially R is valid, and whenever an interval I is accepted, by condition (5), R remains valid.

Let R^* (respectively D^*) denote the final set of intervals accepted by $RAND$ (respectively DET). We proceed to show that $E(|R^*|) \geq \frac{1}{4dc} \times |\mathcal{OPT}_\sigma^{(1)}|$.

- Every interval I is accepted by $RAND$ with probability at most p, because in order for an interval to get accepted, the coin-toss in step (4) has to show "success".
- The set R satisfies $R \subseteq D$, since every interval is accepted by $RAND$ only if it is accepted by DET.

- For all $s \in D^*$, $s \in R^*$ if and only if s is accepted by $RAND$: If s is accepted by $RAND$, then it is never preempted by $RAND$, since $RAND$ preempts intervals only in step (1), only if DET preempts them.

For $s \in S$ define the indicator random variable χ_s to be 1 if $s \in R^*$, and 0 otherwise. By the above observations, for $s \in D^*$ we have

$$E(\chi_s) = Pr[s \in R^*] = Pr[s \textit{ is accepted by } RAND].$$

For $s \in D^*$, $RAND$ accepts s if and only if condition (5) holds. By DET's properties, s intersects at most d intervals in D when it is accepted by DET, and thus it intersects at most d intervals in $R \subseteq D$. The probability that none of those intervals is accepted by $RAND$ is no less than $1 - dp$, so

$$Pr[s \textit{ is accepted by } RAND] \geq p(1 - dp).$$

Thus,

$$E(|R^*|) \geq E(\sum_{s \in D^*} \chi_s) \geq \sum_{s \in D^*} p(1 - dp) \geq p(1 - dp) \cdot |D^*| \geq p(1 - dp) \cdot \frac{|\mathcal{OPT}_\sigma^{(1)}|}{c}$$

To complete the proof, choose $p = \frac{1}{2d}$.

■

Using the theorem we get the following result:

Theorem 4.2 *There is a* 16*-competitive randomized online algorithm for bandwidth* 1.

Proof: Use the algorithm $BW\frac{1}{2}$ for theorem 4.1. The algorithm satisfies the properties above with $c = 2$ (by lemma 3.7) and $d = 2$ (any accepted interval intersects at most 2 intervals in the valid set). ■

5 A constant competitive algorithm for bandwidth $1/k$

In this section we give a constant competitive algorithm for the case where all the intervals request bandwidth of $1/k$, for some fixed $k \geq 2$. We first note that by lemma 3.7 and lemma 3.6, the algorithm $BW\frac{1}{2}$ is at most $2k$-competitive for this problem. Here, however, we present an algorithm whose competitive ratio does not depend on k.

We apply a general method of allocation in bins as in [4], with adaptation to handle preemption. In the general setting of [4], the algorithm has to accommodate a new item in one of several independent "bins". A ρ-competitive allocation algorithm is used for each of the bins, and when an item is rejected from one bin, it proceeds to the next bin. In the above algorithm, however, the bin-algorithms are not allowed to preempt items. We generalize the algorithm for the preemptive case. We will use the term *preemptive allocation algorithm* for algorithms, which decide, for a new item, whether to accept it or reject it, and may preempt old items when a new item is accepted. We prove the following theorems in the appendix:

Theorem 5.1 *Assume there are k "abstract" bins, and an online algorithm has to assign items into one of the bins. Assume A is a ρ-competitive preemptive allocation algorithm for one bin. Then there is a $\rho+1$-competitive preemptive allocation algorithm for allocation into k bins.*

Theorem 5.2 *There is a 5-competitive algorithm for bandwidth allocation of bandwidth $\frac{1}{k}$ intervals for even k, and a 7-competitive algorithm for odd k, $k \geq 3$.*

We also show in the appendix a lower bound theorem for arbitrary bandwidth:

Theorem 5.3 *For any k no deterministic or randomized online algorithm can achieve competitive ratio less than 2 for intervals with bandwidth $= \frac{1}{k}$.*

6 A randomized algorithm for bandwidth ≤ 1

In this section we consider a generalized setting, in which all requested calls σ_i have bandwidth $r_i \leq \delta$, for some fixed δ, $\delta \leq 1$, and the benefit accrued from the interval is it's bandwidth i.e. $b_i = r_i$.

First we show a deterministic constant-competitive algorithm for the case $\delta < \frac{1}{2}$, then we show how this algorithm can be used to construct a randomized constant competitive algorithm for the case $\delta = 1$.

6.1 A constant competitive algorithm for $\delta < \frac{1}{2}$

As mentioned before, we now focus on the case $\delta < \frac{1}{2}$. We first introduce some useful definitions and notations. Let I be an interval, and S be an interval set.

- $S \subset I$ iff $\forall s \in S,\ s \subset I$.
- $S[I] = \{s \in S \mid s \subset I\}$ is the induced subset of S on I. Note that $S = S[I]$ iff $S \subset I$.
- $S[x] = \{s \in S \mid x \in s\}$, the subset of intervals of S which contain x.

In the algorithm for the bandwidth 1 case, we used the fact that there is an optimal solution with no "containing" intervals. We introduce the definition of "stuffed" intervals, which are intervals that contain calls of large total benefit. Such intervals, as shown in the sequel, can be excluded from constant-factor approximations to the optimum, since they can be replaced by non-"stuffed" intervals in the approximation.

Definition 6.1 *Let S be a set of intervals. Let $0 < \lambda \leq 1$.*
An interval $K \in S$ is stuffed *in S if $B(S[K] \setminus \{K\}) \geq \lambda$.*

Definition 6.2 *Let x be violation point of $\mathcal{A}$. Let $\mathcal{L}_R$ be the list of the intervals in $\mathcal{A}[x]$, ordered by ascending order of their right-endpoint. Similarly, let $\mathcal{L}_L$ be the list of the intervals, ordered by descending order of their left-endpoint.*
The right-closest intervals of x in $\mathcal{A}$, $\Psi_R(x)$, is the maximal prefix of $\mathcal{L}_R$ which has total bandwidth $\leq \frac{1}{2}$. The left-closest intervals of x in $\mathcal{A}$, $\Psi_L(x)$, is the maximal prefix of $\mathcal{L}_L$ which has total bandwidth $\leq \frac{1}{2}$.

Procedure: STICKY(interval I)

begin

(1) **if** I is stuffed in S **then** reject I and **return**

(2) add I to $\mathcal{A}$

(3) **while** there are bandwidth violations **do**

(4) pick a **violation point**, x

(5) remove all the intervals K such that $x \in K$ and $K \notin \Psi_L(x) \cup \Psi_R(x)$

end

Figure 4: Algorithm for $\delta < \frac{1}{2}$

Figure 4 shows the deterministic algorithm STICKY which handles an arriving interval I.

Lemma 6.3 *Algorithm STICKY maintains a* valid *set $\mathcal{A}$ of intervals.*

Proof: When the algorithm terminates there are no bandwidth violations in $\mathcal{A}$, by condition (3). The algorithm terminates, since when an iteration of the loop in (5) is executed, at least one interval is removed, and there is a finite number of intervals. ■

In the appendix, we prove the following theorem:

Theorem 6.4 *For $\lambda = \frac{1}{3}$, algorithm STICKY is constant competitive for $\delta < \frac{1}{2}$. Specifically, for $\delta = \frac{1}{4}$, the algorithm is 72-competitive.*

We note that the algorithm does not need to know δ in advance, as long as $\delta < \frac{1}{2}$.

6.2 A randomized algorithm for $\delta = 1$

Now we can construct a randomized competitive algorithm for $\delta = 1$ by classifying the request series, and applying one of the previous algorithms on each class.

More specifically, the requests are classified into the following 2 classes:

- All the requests with $r_i \geq \frac{1}{4}$. Those requests are handled with the 16-competitive randomized algorithm, setting all $r_i = 1$.
- All the requests with $r_i < \frac{1}{4}$. Those requests are handled by STICKY.

The algorithm for $\delta = 1$ randomly chooses one of the classes, each with probability $\frac{1}{2}$, and handles only requests of this class by the appropriate algorithm.

Theorem 6.5 *The above algorithm is 144-competitive.*

Proof: The first algorithm is 16*4-competitive with respect to the requests of the first class, by lemma 3.6. The second algorithm is 72-competitive with respect to the requests of the second class, by theorem 6.4. Since each algorithm is chosen with probability $\frac{1}{2}$, a standard argument shows that the final algorithm is $2 * \max\{64, 72\} = 144$ competitive. ■

References

[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan. Efficient routing and scheduling algorithms for optical networks. In *Proc. 5th ACM-SIAM Symp. on Discrete Algorithms*, pages 412–423, 1994.

[2] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *Journal of the ACM*, 44(3):486–504, 1997. Also in *Proc. 25th ACM STOC*, 1993, pp. 623-631.

[3] Y. Aumann and Y. Rabani. Improved bounds for all optical routing. In *Proc. 6th ACM-SIAM Symp. on Discrete Algorithms*, pages 567–576, 1995.

[4] B. Awerbuch, Y. Azar, A. Fiat, S. Leonardi, and A. Rosen. On-line competitive algorithms for call admission in optical networks. In *Proc. 4th Annual European Symposium on Algorithms*, pages 431–444, 1996.

[5] B. Awerbuch, Y. Azar, and S. Plotkin. Throughput-competitive online routing. In *34th IEEE Symposium on Foundations of Computer Science*, pages 32–40, 1993.

[6] B. Awerbuch, Y. Bartal, A. Fiat, and A. Rosén. Competitive non-preemptive call control. In *Proc. of 5th ACM-SIAM Symposium on Discrete Algorithms*, pages 312–320, 1994.

[7] Baruch Awerbuch, Rainer Gawlick, Tom Leighton, and Yuval Rabani. On-line admission control and circuit routing for high performance computation and communication. In *Proc. 35th IEEE Symp. on Found. of Comp. Science*, pages 412–423, 1994.

[8] A. Bar-Noy, R. Canetti, S. Kutten, Y. Mansour, and B. Schieber. Bandwidth allocation with preemption. In *Proc. 27th ACM Symp. on Theory of Computing*, pages 616–625, 1995.

[9] Y. Bartal, A. Fiat, and S. Leonardi. Lower bounds for on-line graph problems with application to on-line circuit and optical routing. In *Proc. 28th ACM Symp. on Theory of Computing*, pages 531–540, 1996.

[10] R. Canetti and S. Irani. Bouding the power of preemption in randomized scheduling. In *Proc. 27th ACM Symp. on Theory of Computing*, pages 606–615, 1995.

[11] Juan Garay, Inder Gopal, Shay Kutten, Yishay Mansour, and Moti Yung. Efficient on-line call control algorithms. *Journal of Algorithms*, 23:180–194, 1997. Also in *Proc. 2'nd Annual Israel Conference on Theory of Computing and Systems*, 1993.

[12] J. Kleinberg and E. Tardos. Approximations for the disjoint paths problem in high-diameter planar networks. In *Proc. 27th ACM Symp. on Theory of Computing*, pages 26–35, 1995.

[13] J. Kleinberg and E. Tardos. Disjoint paths in densely embedded graphs. In *Proc. 36th IEEE Symp. on Found. of Comp. Science*, pages 52–61, 1995.

[14] S. Leonardi. On-line routing. Dagstuhl 1996, to be published in Springer lecture notes, 1998.

[15] S. Leonardi, A. Marchetti-Spaccamela, A. Presciutti, and A. Rosén. On-line randomized call control revisited. In *Proc. 9th ACM-SIAM Symp. on Discrete Algorithms*, pages 323–332, 1998.

[16] R. J. Lipton and A. Tomkins. Online interval scheduling. In *Proc. of the 5th ACM-SIAM Symposium on Discrete Algorithms*, pages 302–311, 1994.

[17] Prabhakar Raghavan and Eli Upfal. Efficient routing in all-optical networks. In *Proc. 26th ACM Symp. on Theory of Computing*, pages 134–143, 1994.

Appendix

A The bandwidth $\frac{1}{k}$ case

Proof of theorem 5.1:
We describe an algorithm A' for k bins, in terms of a procedure B_i for $1 \leq i \leq k$, which maintains the i'th bin. The procedure B_i proceeds as follows, given a new item t:

Procedure B_i
begin
(1) run A on t for bin i
(2) **if** t was rejected **then**
(3)   **if** $i < k$ **then** call B_{i+1} on t
(4)   **else** reject t
    else
(5)   accept I into bin i
(6)   **if** items $r_1, r_2..r_m$ were preempted **then**
(7)   **if** $i < k$ **then**
(8)   **for each** $1 \leq j \leq m$ sequentially call B_{i+1} on r_j
(9)   **else** preempt $r_1, r_2..r_m$
        end if
    end if
end

Figure 5: A procedure B_i for allocation into bin i

Algorithm A' calls B_1. It is easy to see, by induction from k to 1, that procedure B_i terminates, and thus A' terminates. We claim that A' is a $\rho + 1$-competitive algorithm for k bins.

For $1 \leq i \leq k$, let O_i denote the set of items accepted into bin i by the optimal algorithm, and let T_i denote the set of items accepted into bin i, which were not preempted in step (9).

By construction of the algorithm, procedure B_i is presented with at least all the items in the set $O_i \setminus \cup_{j<i} T_j$ (not necessarily in the original order). Since B_i uses A internally, it is ρ competitive, so it will gain at least:

$$B(T_i) \geq \frac{1}{\rho} \cdot B(O_i \setminus (\cup_{j<i} T_j)) = \frac{1}{\rho} B(O_i) - \frac{1}{\rho} B((\cup_{j<i} T_j) \cap O_i) \ .$$

It follows that

$$\begin{aligned} \sum_{i=1}^{k} B(T_i) &\geq \sum_{i=1}^{k} \frac{1}{\rho} B(O_i) - \sum_{i=1}^{k} \frac{1}{\rho} B((\cup_{j<i} T_j) \cap O_i) \geq \sum_{i=1}^{k} \frac{1}{\rho} B(O_i) - \frac{1}{\rho} B(\cup_{i \leq k} T_i) \\ &\geq \sum_{i=1}^{k} \frac{1}{\rho} B(O_i) - \frac{1}{\rho} \sum_{i=1}^{k} B(T_i) \ , \end{aligned}$$

where the second inequality follows since the sets O_i are pairwise disjoint. Thus $(1+\rho)\sum_{i=1}^{k} B(T_i) \geq \sum_{i=1}^{k} B(O_i)$, and A' is $\rho+1$-competitive.

■

Proof of theorem 5.2:
We divide the total bandwidth into bins, and use the previous theorem to allocate intervals to the bins, in the following way:

- For even k, divide the total bandwidth into $\frac{k}{2}$ "bins", each consisting of bandwidth $\frac{2}{k}$, and use algorithm $BW\frac{1}{2}$ for allocation in each bin. By lemma 3.7 $BW\frac{1}{2}$ is 4-competitive on each bin, so we get a 5-competitive algorithm.
- For odd k, divide the total bandwidth into $\lfloor\frac{k}{2}\rfloor$ "bins", $\lfloor\frac{k}{2}\rfloor-1$ of which consisting of bandwidth $\frac{2}{k}$, the remaining one consisting of bandwidth $\frac{3}{k}$, and use algorithm $BW\frac{1}{2}$ for allocation in each bin. As mentioned before, $BW\frac{1}{2}$ is 6-competitive on each bin, so we get a 7-competitive algorithm.

■

Proof of theorem 5.3:
For an appropriate M, the adversary requests k calls in the interval $(0, M+1)$ (left side calls), and k more calls in the interval $(M, 2M+1)$ (right side calls). Define the "internal part" of the left side (right, respectively) to be $(0, M)$ ($(M+1, 2M+1)$, respectively).

Since all the calls overlap, in at least one of the sides the expected number of calls accepted by the online algorithm is at most $\frac{k}{2}$. Then the adversary accepts *all* the calls in this side (k calls), and continues the construction recursively in the "internal part" of the opposite side.

Repeating the construction recursively n times we conclude that the expected gain of the online algorithm is at most $1+(n-1)\frac{1}{2}$, whereas the adversary gains n, which proves the claim. ■

B STICKY is constant competitive

In this appendix we prove theorem 6.4.

The following claim follows immediately from the fact that the bandwidth of each interval is at most δ:

Claim B.1 *Let $\mathcal{A}$ be an interval set. If $\delta < \frac{1}{2}$, then $B(\Psi_R(x)) \geq \frac{1}{2}-\delta$, $B(\Psi_L(x)) \geq \frac{1}{2}-\delta$,*

Lemma B.2 *Let the intervals J_1, J_2 satisfy $J_1 \subset J_2$. If interval J_1 is removed in step (5), then by the time step (5) is over, $J_2 \notin \mathcal{A}$.*

Proof: Let $x, \mathcal{L}_L, \mathcal{L}_R$ be as in definition 6.2, at the moment J_1 is removed in step (5). Now $x \in J_1$, $x \in J_2$, so if $J_2 \in \mathcal{A}$ by the time J_1 is being removed, J_2 appears in $\mathcal{L}_L$ and $\mathcal{L}_R$. Since $J_1 \subset J_2$, J_2 appears after J_1 in both of the lists $\mathcal{L}_L$ and $\mathcal{L}_R$, and since $J_1 \notin \Psi_L(x) \cup \Psi_R(x)$, $J_2 \notin \Psi_L(x) \cup \Psi_R(x)$, so J_2 is now removed. ■

Definition B.3 *For an interval set N, interval $C \subseteq \Re$ is a* cell *of N, if it contains two disjoint intervals C_L, C_R (C_L is the left one) such that $B(N[C_L]) \geq 1$ and $B(N[C_R]) \geq 1$.*

Next is a "subdivision" lemma, which demonstrates how to split the original interval set S into $\Omega(B(\mathcal{OPT}_\sigma))$ cells, like the ones in the proof of lemma 3.7.

Lemma B.4 *Let N be a valid set of intervals. Then there exist at least $\frac{B(N)}{2(2+\delta)} - \frac{3}{2}$ pairwise disjoint cells of N.*

Proof: Order the intervals of N from left to right by their left-endpoint. Now, scanning the list, pick minimal subsets $\{L_i\}_{i\geq 0}$ from the start of the list, such that $B(L_i) \geq 2$. Then $B(L_i) < 2 + \delta$. Thus, there are at least $\lfloor \frac{B(N)}{2+\delta} \rfloor \geq \frac{B(N)}{2+\delta} - 1$ such subsets.

Let x_i be the left most left endpoint of the intervals in L_i. By construction $\{x_i\}_{j\geq 0}$ is an ascending sequence. Add a last element $x_{last} = \infty$ to the sequence.
Define $N_i = \{n \in L_i \mid x_{i+1} \notin n\}$. Then $B(N_i) \geq 1$, otherwise N violates the bandwidth constraint on x_{i+1}. Thus $B(N[(x_i, x_{i+1})]) \geq 1$.

Now let $C^{(i)} = (x_{2i}, x_{2(i+1)})$ for $i \geq 0$. By the above, $C^{(i)}$ satisfies the definition of a cell of N, with the following subintervals: $C_L^{(i)} = (x_{2i}, x_{2i+1})$, and $C_R^{(i)} = (x_{2i+1}, x_{2i+2})$. By definition, the intervals $C^{(i)}$ are pairwise disjoint, and there are at least $\lfloor \frac{\frac{B(N)}{(2+\delta)} - 1}{2} \rfloor \geq \frac{B(N)}{2(2+\delta)} - \frac{3}{2}$ such intervals, as claimed. ■

Lemma B.5 *Let $J \subset \Re$, and let N be a set of intervals, with no stuffed intervals of the final set S^*. If $B((S_t \cap N)[J]) \geq 2\lambda$ and there is $K \in \mathcal{A}_t$ s.t. $K \supset J \quad (K \neq J)$, then $B(\mathcal{A}_t[J]) \geq \lambda$.*

Proof: Suppose $K \supset J$, $K \neq J$ is in $\mathcal{A}_t$. Denote by t' the time when K arrived. Since K was not rejected in step (1), $B((S_{t'} \cap N)[J]) \leq B(S_{t'}[J]) < \lambda$. Let $N' = (S_t \setminus S_{t'}) \cap N$, the intervals of $S_t \cap N$ which arrived after t'. Then $B(N'[J]) \geq (2\lambda - \lambda) = \lambda$. By lemma B.2, if one of the intervals in $N'[J]$ is removed, K is removed too, since K strictly contains all the intervals of $N'[J]$. Thus by time t none of the intervals in $N'[J]$ was removed, and $B(\mathcal{A}_t[J]) \geq B(N'[J]) \geq \lambda$. ■

Lemma B.6 *Let N be a valid set with no stuffed intervals of S^*, and let C be a cell of N. Assume $\lambda \leq \frac{1}{3}$. Then $B(\mathcal{A}^*[C]) \geq \min\{\frac{1}{2} - \delta, \lambda\}$.*

Proof: Since $B(S_t \cap N)$ is non-decreasing with t, let t_r be the first time $B((S_{t_r} \cap N)[C_R]) \geq 2\lambda$, and t_l be the first time $B((S_{t_l} \cap N)[C_L]) \geq 2\lambda$. Since C_L and C_R are disjoint, $t_l \neq t_r$, and we can assume $t_r < t_l$. The proof for the other case is symmetrical.

Now there are three cases:

- *No interval is removed from $\mathcal{A}[C]$ in time t_l, or later.*
  Let $N' = (S^* \setminus S_{t_l - 1}) \cap N$. Since N contains no stuffed intervals of S^*, no interval of N' can be rejected in step (1). Thus all the intervals in N' are accepted, and never removed, so $B(\mathcal{A}^*[C]) \geq B(N'[C_L]) \geq 1 - 2\lambda \geq \lambda$, which completes the proof.

- *Interval J is removed from $\mathcal{A}[C]$ at time t_l.*
  Denote by x the violation point that caused the removal of J. Then $x \in C_L$, since x is in the last arriving interval, which belongs to $N[C_L]$, by the assumption. Now examine the intervals in $\Psi_L(\mathrm{x})$. Since J was removed, by the definition of $\Psi_L(\mathrm{x})$ we

get $\forall l \in \Psi_L(x)$, $right(C_L) \geq x \geq left(l) \geq left(J)$. Now, if there is an interval $l \in \Psi_L(x)$ such that $right(l) > right(C_R)$, l strictly contains C_R, and by lemma B.5, $B(\mathcal{A}[C]) \geq B(\mathcal{A}[C_R]) \geq \lambda$ holds. Otherwise, all the intervals in $\Psi_L(x)$ are contained in C, so $B(\mathcal{A}[C]) \geq B(\Psi_L(x)) \geq \frac{1}{2} - \delta$ holds.

- *Interval J is removed from $\mathcal{A}[C]$ after time t_l.*
  Denote by $x \in C$ the violation point that caused the removal of J. Then $x \in J \subseteq C$. Now examine the intervals in $\Psi_L(\text{x})$ and $\Psi_R(\text{x})$. Since J was removed, by the definition of $\Psi_L(\text{x})$ and $\Psi_R(\text{x})$, we get that $\forall r \in \Psi_R(x)$, $right(r) \leq right(J) \leq right(C)$ and $\forall l \in \Psi_L(x)$, $left(l) \geq left(J) \geq left(C)$. If there is an interval $l \in \Psi_L(x)$ and an interval $r \in \Psi_R(x)$ such that $left(r) < left(C)$ and $right(l) > right(C)$, then one of the intervals l or r would strictly contain one of the intervals C_L or C_R, and then by lemma B.5, either $B(\mathcal{A}[C_L]) \geq \lambda$ or $B(\mathcal{A}[C_R]) \geq \lambda$. Otherwise, either $\forall r \in \Psi_R(x)$, $left(r) \geq left(C)$ or $\forall l \in \Psi_L(x)$, $right(l) \leq right(C)$. In either case, $\Psi_R(x) \subseteq C$ and $B(\Psi_R(x)) \geq \frac{1}{2} - \delta$ or $\Psi_L(x) \subseteq C$ and $B(\Psi_L(x)) \geq \frac{1}{2} - \delta$, which completes the proof.

■

Lemma B.7 *For $\lambda \leq 1 - \delta$ let $T = \{t \in S \mid t$ is stuffed in $S\}$, and let $S' = S \setminus T$. Let $O = \mathcal{OPT}_\sigma$, and O' be an optimal subset of S'. Then $B(O') \geq \frac{\lambda}{1+\lambda} \cdot B(O)$, and the bound is tight.*

Proof: We construct a valid set $\mathcal{N} \subseteq S'$, such that $B(\mathcal{N})$ satisfies the above inequality. Since for $t \in T$, $B(S[t]) \geq \lambda$, there is a subset $F_t \subseteq S[t]$ such that $\lambda \leq B(F_t) < \lambda + \delta$. Define the mapping $F^* : T \to 2^{S'}$ by the following recursive definition:

$$F^*(t) = \begin{cases} F_t & \text{if } F_t \subseteq S' \\ F^*(t') & \text{otherwise, for some } t' \in F_t \cap T \end{cases}$$

Since T is finite, and there are no identical intervals, F^* is well defined. By definition, $F^*(t) \subset t$ and $\lambda \leq B(F^*(t)) \leq \lambda + \delta$.

Let $\mathcal{U}$ be a (maximum size) set of pairwise-disjoint intervals in $T \cap O$ defined as follows: order all the intervals in $T \cap O$ from left to right by their left endpoint, and keep adding the next interval with the left-most right endpoint, which does not intersect previously added intervals, until the intervals are exhausted.

By construction, every interval of $T \cap O$ intersects the right endpoint of some interval in $\mathcal{U}$. For $u \in \mathcal{U}$, denote by T_u the set of intervals $t \in T \cap O$ for which u is the left-most interval such that t intersects the right endpoint of u. Then $O \cap T = \bigcup_{u \in \mathcal{U}} T_u$, and $B(O \cap T) = \sum_{u \in \mathcal{U}} B(T_u)$, where the union above is disjoint.

Now define $\mathcal{N}' = \bigcup_{u \in \mathcal{U}} F^*(u)$. It is clear that $\mathcal{N}' \subseteq S'$. $\mathcal{N}'$ is valid, since $F^*(u) \subset u$, $u \in \mathcal{U}$ are disjoint, and $B(F^*(u)) \leq \lambda + \delta \leq 1$. Since T_u is a subset of a valid set, and $right(u) \in \cap_{v \in T_v} v$, we have $B(T_u) \leq 1 \leq \frac{1}{\lambda} B(F^*(u))$. Thus

$$B(O \cap T) = \sum_{u \in \mathcal{U}} B(T_u) \leq \sum_{u \in \mathcal{U}} \frac{1}{\lambda} B(F^*(u)) \leq \frac{1}{\lambda} B(\mathcal{N}')$$

where the last inequality follows from the fact $F^*(u) \subset u$ are disjoint. Let $\mathcal{N} = \mathcal{N}'$ if $B(\mathcal{N}') > B(O \cap S')$ or $\mathcal{N} = O \cap S'$ otherwise. We claim that $B(\mathcal{N}) \geq \frac{\lambda}{\lambda+1} B(O)$.

To prove the claim, assume $B(O \cap S') = \alpha B(O)$, and thus $B(O \cap T) = (1-\alpha)B(O)$. Then $B(\mathcal{N}) \geq \max\{\alpha B(O), (1-\alpha)\lambda B(O)\}$, which attains it's minimal value for $\alpha = \frac{\lambda}{\lambda+1}$, for which $B(\mathcal{N}) \geq \frac{\lambda}{\lambda+1} B(O)$, as claimed.

The following set S shows an upper bound of $\frac{\lambda}{1+\lambda}$ for $\frac{B(O')}{B(O)}$: for small ϵ, pick $\frac{\lambda}{\epsilon}$ disjoint intervals of bandwidth ϵ, and one more interval of bandwidth $1-\epsilon$ which contains all the other intervals. Then:

$$\frac{B(O')}{B(O)} = \frac{\lambda}{\lambda+1-\epsilon} \stackrel{\epsilon \to 0}{\to} \frac{\lambda}{\lambda+1}$$

■

Proof of theorem 6.4:

Let S', O' be as in lemma B.7. Then $B(O') \geq \frac{\lambda}{1+\lambda} B(\mathcal{OPT}_\sigma)$. By lemma B.4, there exist at least $\frac{B(O')}{2(2+\delta)} - \frac{3}{2}$ pairwise disjoint cells $C \in \mathcal{C}$ of O'. By lemma B.6, $B(\mathcal{A}[C]) \geq \min\{\frac{1}{2} - \delta, \lambda\}$ for each such cell, so summing over all cells we get:

$$B(\mathcal{A}) \geq \sum_{C \in \mathcal{C}} B(\mathcal{A}[C]) \geq \min\{\tfrac{1}{2} - \delta, \lambda\} \cdot \left(\frac{\frac{\lambda}{1+\lambda} B(\mathcal{OPT}_\sigma)}{2(2+\delta)} - \frac{3}{2}\right)$$

Thus choosing $\lambda = \frac{1}{3}$, the competitive ratio of the algorithm is $\frac{8(2+\delta)}{\min\{\frac{1}{2}-\delta, \frac{1}{3}\}}$, i.e., for $\delta \leq \frac{1}{6}$ the algorithm is $24(2+\delta)$-competitive, and for $\frac{1}{6} \leq \delta < \frac{1}{2}$ the algorithm is $\frac{8(2+\delta)}{\frac{1}{2}-\delta}$-competitive. Specifically, the algorithm is 72-competitive for $\delta = \frac{1}{4}$. ■