An Improved Algorithm for Online Unit Clustering

Hamid Zarrabi-Zadeh Timothy M. Chan

School of Computer Science, University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1
{hzarrabi,tmchan}@uwaterloo.ca

Abstract. We revisit the *online unit clustering* problem in one dimension which we recently introduced at WAOA'06: given a sequence of n points on the line, the objective is to partition the points into a minimum number of subsets, each enclosable by a unit interval. We present a new randomized online algorithm that achieves expected competitive ratio 11/6 against oblivious adversaries, improving the previous ratio of 15/8. This immediately leads to improved upper bounds for the problem in two and higher dimensions as well.

1 Introduction

At WAOA'06 [1], we began investigating an online problem we call *unit clustering*, which is extremely simple to state but turns out to be nontrivial surprisingly:

> Given a sequence of n points on the real line, assign points to clusters so that each cluster is enclosable by a unit interval, with the objective of minimizing the number of clusters used.

In the offline setting, variations of this problem frequently appear as textbook exercises and can be solved in $O(n \log n)$ time by a simple greedy algorithm (e.g., see [3]). The problem is equivalent to finding the minimum number of points that stab a given collection of unit intervals (i.e., clique partitioning in unit interval graphs, or coloring unit co-interval graphs), and to finding the maximum number of disjoint intervals in a given collection (i.e., maximum independent set in unit interval graphs). It is the one-dimensional analog of an often-studied and important geometric clustering problem—covering a set of points in d dimensions using a minimum number of unit disks (for example, under the Euclidean or L_∞ metric) [5,6,8,11,12]. This geometric problem has applications in facility location, map labeling, image processing, and other areas.

Online versions of clustering and facility location problems are natural to consider because of practical considerations and have been extensively studied in the literature [2,4,10]. Here, input points are given one by one as a sequence over time, and each point should be assigned to a cluster upon its arrival. The main constraint is that clustering decisions are irrevocable: once formed, clusters cannot be removed or broken up.

For our one-dimensional problem, it is easy to come up with an algorithm with competitive ratio 2; for example, we can use a naïve grid strategy: build a uniform unit grid and simply place each arriving point in the cluster corresponding to the point's grid cell (for the analysis, just observe that every unit interval intersects at most 2 cells). Alternatively, we can use the most obvious greedy strategy: for each given point, open a new cluster only if the point does not "fit" in any existing cluster; this strategy too has competitive ratio 2.

In the previous paper [1], we have shown that it is possible to obtain an online algorithm with expected competitive ratio strictly less than 2 using randomization; specifically, the ratio obtained is at most $15/8 = 1.875$. This result is a pleasant surprise, considering that ratio 2 is known to be tight (among both deterministic and randomized algorithms) for the related *online unit covering* problem [2,1] where the position of each enclosing unit interval is specified upon its creation, and this position cannot be changed later. Ratio 2 is also known to be tight among deterministic algorithms for the problem of online coloring of (arbitrary rather than unit) co-interval graphs [7,9].

In this paper, we improve our previous result further and obtain a randomized online algorithm for one-dimensional unit clustering with expected competitive ratio at most $11/6 \approx 1.8333$. Automatically, this implies improved online algorithms for geometric unit clustering under the L_∞ metric, with ratio $11/3$ in 2D, for example.

The new algorithm is based on the approach from the previous paper but incorporates several additional ideas. A key difference in the design of the algorithm is to make more uses of randomization (the previous algorithm requires only 2 random bits). The previous algorithm is based on a clever grid approach where windows are formed from pairs of adjacent grid cells, and clusters crossing two adjacent windows are "discouraged"; in the new algorithm, crossings of adjacent windows are discouraged to a "lesser" extent, as controlled by randomization. This calls for other subtle changes in the algorithm, as well as a lengthier case analysis that needs further technical innovations.

2 The New Randomized Algorithm

In this section, we present the new randomized algorithm for the online unit clustering problem in one dimension. The competitive ratio of the algorithm is not necessarily less than 2, but will become less than 2 when combined with the naïve grid strategy as described in Section 5. Our new algorithm is based in part on our previous randomized algorithm [1], although we will keep the presentation self-contained. A key difference is to add an extra level of randomization.

Consider a uniform unit grid on the line, where each grid cell is a half-closed interval of the form $[i, i+1)$. To achieve competitive ratio better than 2, we have to allow clusters to cross grid cells occasionally (for example, just consider the input sequence $\langle \frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \ldots \rangle$, where the naïve grid strategy would require twice as many clusters as the optimum). As in the previous algorithm, we accomplish this by forming *windows* over the line each consisting of two grid cells and permit clusters crossing two cells within a window. There are two ways to form windows over the grid; we choose which according to an initial random bit. In the previous algorithm, clusters crossing two adjacent windows are not strictly forbidden but are discouraged in some sense.

In the new algorithm, the idea, roughly speaking, is to permit more clusters crossing windows. More specifically, call the grid point lying between two adjacent windows a *border*; generate a random bit for every border, where a 1 bit indicates an *open* border and a 0 bit indicates a *closed* border. Clusters crossing closed borders are still discouraged, but not clusters crossing open borders. (As it turns out, setting the probability of border opening/closing to 1/2 is indeed the best choice.)

The actual details of the algorithm are important and are carefully crafted. In the pseudocode below, $b(w, w')$ refers to the border indicator between windows w and w'. We say that a point *lies* in a cluster if inserting it to the cluster would not increase the length of the cluster, where the *length* of a cluster refers to the length of its smallest enclosing interval. We say that a point *fits* in a cluster if inserting it to the cluster would not cause the length to exceed 1.

RandBorder Algorithm: Partition the line into *windows* each of the form $[2i, 2i+2)$. With probability 1/2, shift all windows one unit to the right. For each two neighboring windows w and w' set $b(w, w')$ to a randomly drawn number from $\{0, 1\}$. For each new point p, find the window w containing p, and do the following:

1: **if** p fits in a cluster intersecting w **then**
2:     put p in the "closest" such cluster
3: **else if** p fits in a cluster u inside a neighboring window w' **then**
4:     **if** $b(w, w') = 1$ **then** put p in u
5:     **else if** w (completely) contains at least 1 cluster and
        w' (completely) contains at least 2 clusters
6:         **then** put p in u
7: **if** p is not put in any cluster **then** open a new cluster for p

---

Thus, a cluster is allowed to cross the boundary of two grid cells within a window freely, but it can cross the boundary of two adjacent windows only in two exceptional cases: when the corresponding border indicator is set to 1, or when the carefully specified condition in Line 5 arises (this condition is slightly different from the one in the previous algorithm). We will see the rationale for this condition during the analysis.

To see what the "closeness" exactly means in Line 2, we define the following two preference rules:

- Rule I. If p lies in a cluster u, then u is the closest cluster to p.
- Rule II. If p lies in a cell c, then any cluster intersecting c is closer to p than any cluster contained in a neighboring cell.

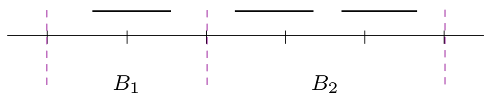

Fig. 1. Two blocks of sizes 2 and 3.

The first preference rule prevents clusters from overlapping each other, and the second rule prevents clusters from unnecessarily crossing the boundary of two neighboring cells. The above preference rules and exceptional cases will be vital to the analysis.

Note that the random bits used for the border indicators can be easily generated on the fly as new borders are created.

3 Preliminaries for the Analysis

To prepare for the analysis, we first state a few definitions (borrowed from [1]).

Let σ be the input sequence. We denote by $\mathtt{opt}(\sigma)$ the optimal offline solution obtained by the following greedy algorithm: sort all points in σ from left to right; cover the leftmost point p and all points within unit distance to it by a unit interval started at p; and repeat the procedure for the remaining uncovered points. Obviously, the unit intervals obtained by this algorithm are disjoint.

We refer to a cluster as a *crossing cluster* if it intersects two adjacent grid cells, or as a *whole cluster* if it is contained completely in a grid cell.

For any real interval x (e.g., a grid cell or a group of consecutive cells), the *cost* of x denoted by $\mu(x)$ is defined to be the number of whole clusters contained in x plus half the number of clusters crossing the boundaries of x, in the solution produced by the RandBorder algorithm. We note that μ is additive, i.e., for two adjacent intervals x and y, $\mu(x \cup y) = \mu(x) + \mu(y)$.

A set of k consecutive grid cells containing $k-1$ intervals from $\mathtt{opt}(\sigma)$ is called a *block* of size k (see Fig. 1). We define $\rho(k)$ to be the expected competitive ratio of the RandBorder algorithm within a block of size k. In other words, $\rho(k)$ upper-bounds the expected value of $\mu(B)/(k-1)$ over all blocks B of size k.

In the following, a list of objects (e.g., grid cells or clusters) denoted by $\langle x_i, \ldots, x_j \rangle$ is always implicitly assumed to be ordered from left to right on the line. Moreover, $p_1 \ll p_2$ denotes the fact that point p_1 arrives before point p_2 in the input sequence.

We now establish some observations concerning the behavior of the RandBorder algorithm. Observations 1(ii) and (iii) are basically from [1] and have similar proofs (which are reproduced here for completeness' sake since the algorithm has changed); the other observations and subsequent lemmas are new and will be used multiple times in the analysis in the next section.

Observation 1

(i) *Any interval in* $\mathtt{opt}(\sigma)$ *that does not cross a closed border can (completely) contain at most one whole cluster.*

(ii) *Any grid cell c can contain at most one whole cluster. Thus, we always have $\mu(c) \leq 1 + \frac{1}{2} + \frac{1}{2} = 2$.*
(iii) *If a grid cell c intersects a crossing cluster u_1 and a whole cluster u_2, then u_2 must be opened after u_1 has been opened, and after u_1 has become a crossing cluster.*

Proof. (i) Let u_1 and u_2 be two whole clusters contained in the said interval and suppose that u_1 is opened before u_2. Then all points of u_2 would be assigned to u_1, because Lines 2 and 4 precede Line 7. (ii) holds by the same argument, because Line 2 precedes Line 7.

For (iii), let p_1 be the first point of u_1 in c and p'_1 be the first point of u_1 in a cell adjacent to c. Let p_2 be the first point of u_2. Among these three points, p_1 cannot be the last to arrive: otherwise, p_1 would be assigned to the whole cluster u_2 instead of u_1, because of Rule II. Furthermore, p'_1 cannot be the last to arrive: otherwise, p_1 would be assigned to u_2 instead. So, p_2 must be the last to arrive. □

Observation 2 *Let u_1 be a whole cluster contained in a grid cell c, and let u_2 and u_3 be two clusters crossing the boundaries of c. Then*

(i) *u_1 and u_2 cannot be entirely contained in the same interval from* $\mathtt{opt}(\sigma)$*.*
(ii) *there are no two intervals I_1 and I_2 in* $\mathtt{opt}(\sigma)$ *such that $u_1 \cup u_2 \cup u_3 \subseteq I_1 \cup I_2$.*

Proof. (i) Suppose by way of contradiction that u_1 and u_2 are entirely contained in an interval I from $\mathtt{opt}(\sigma)$. Then by Observation 1(iii), u_1 is opened after u_2 has become a crossing cluster, but then the points of u_1 would be assigned to u_2 instead: a contradiction.

(ii) Suppose that $u_1 \cup u_2 \cup u_3 \subseteq I_1 \cup I_2$, where I_1 and I_2 are the two intervals from $\mathtt{opt}(\sigma)$ intersecting c. We now proceed as in part (i). By Observation 1(iii), u_1 is opened after u_2 and u_3 have become crossing clusters, but then the points of u_1 would be assigned to u_2 or u_3 instead: a contradiction. □

Lemma 1. *Let $B = \langle c_1, \ldots, c_k \rangle$ be a block of size $k \geq 2$, and S be the set of all odd-indexed (or even-indexed) cells in B. Then there exists a cell $c \in S$ such that $\mu(c) < 2$.*

Proof. Let $\langle I_1, \ldots, I_{k-1} \rangle$ be the $k-1$ intervals from $\mathtt{opt}(\sigma)$ in B, where each interval I_i intersects two cells c_i and c_{i+1} ($1 \leq i \leq k-1$). Let O represent the set of all odd integers between 1 and k. We first prove the lemma for the odd-indexed cells.

Suppose by way of contradiction that for each $i \in O$, $\mu(c_i) = 2$. It means that for each $i \in O$, c_i intersects three clusters $\langle u_i^\ell, u_i, u_i^r \rangle$, where u_i is a whole cluster, and u_i^ℓ and u_i^r are two crossing clusters. We prove inductively that for each $i \in O$, $u_i \cap I_i \neq \emptyset$ and $u_i^r \cap I_{i+1} \neq \emptyset$.

Base Case: $u_1 \cap I_1 \neq \emptyset$ and $u_1^r \cap I_2 \neq \emptyset$.
The first part is trivial, because c_1 intersects just I_1, and hence, $u_1 \subseteq I_1$. The second part is implied by Observation 2(i), because u_1 and u_1^r cannot be entirely contained in I_1.

Inductive Step: $u_i \cap I_i \neq \emptyset \wedge u_i^r \cap I_{i+1} \neq \emptyset \Rightarrow u_{i+2} \cap I_{i+2} \neq \emptyset \wedge u_{i+2}^r \cap I_{i+3} \neq \emptyset$. Suppose by contradiction that $u_{i+2} \cap I_{i+2} = \emptyset$. Therefore, u_{i+2} must be entirely contained in I_{i+1}. On the other hand, $u_i^r \cap I_{i+1} \neq \emptyset$ implies that u_{i+2}^ℓ is entirely contained in I_{i+1}. But this is a contradiction, because u_{i+2} and u_{i+2}^ℓ are contained in the same interval, which is impossible by Observation 2(i). Now, suppose that $u_{i+2}^r \cap I_{i+3} = \emptyset$. Since $u_i^r \cap I_{i+1} \neq \emptyset$, and clusters do not overlap, u_{i+2}^ℓ, u_{i+2}, and u_{i+2}^r should be contained in $I_{i+1} \cup I_{i+2}$, which is impossible by Observation 2(ii).

Repeating the inductive step zero or more times, we end up at either $i = k$ or $i = k - 1$. If $i = k$, then $u_k \cap I_k \neq \emptyset$ which is a contradiction, because there is no I_k. If $i = k - 1$, then $u_{k-1}^r \cap I_k \neq \emptyset$ which is again a contradiction, because we have no I_k.

Both cases lead to contradiction. It means that there exists some $i \in O$ such that $\mu(c_i) < 2$. The proof for the even-indexed cells is similar. The only difference is that we need to prove the base case for $i = 2$, which is easy to get by Observations 2(i) and 2(ii). □

Lemma 2. *Let B be a block of size $k \geq 2$.*

(i) $\mu(B) \leq 2k - 1$.
(ii) *If all borders strictly inside B are open, then* $\mu(B) \leq 2(k - 1)$.

Proof. (i) is a direct corollary of Lemma 1, because there are at least two cells in B (one odd-indexed and one even-indexed) that have cost at most 3/2, and the other cells have cost at most 2.

(ii) is immediate from the fact that each block of size $k \geq 2$ contains exactly $k - 1$ intervals from $\texttt{opt}(\sigma)$, and that each of these $k - 1$ intervals has cost at most 2 by Observation 1(i). □

4 The Analysis

We are now ready to analyze the expected competitive ratio of our algorithm within a block of size $k \geq 2$.

Theorem 1. $\rho(2) = 27/16$.

Proof. Consider a block B of size 2, consisting of two cells $\langle c_1, c_2 \rangle$ (see Fig. 2). Let I be the single unit interval in B in $\texttt{opt}(\sigma)$. There are two possibilities.

Case 1: B falls completely in one window w. Let $\langle b_1, b_2 \rangle$ be the two border indicators at the boundaries of w. Let p_0 be the first point to arrive in I. W.l.o.g., assume p_0 is in c_2 (the other case is symmetric). We consider four subcases.

- Subcase 1.1: $\langle b_1, b_2 \rangle = \langle 0, 0 \rangle$. Here, both boundaries of B are closed. Thus, after a cluster u has been opened for p_0 (by Line 7), all subsequent points in I are put in the same cluster u. Note that the condition in Line 5 prevents points from the neighboring windows to join u and make crossing clusters. So, u is the only cluster in B, and hence, $\mu(B) = 1$.

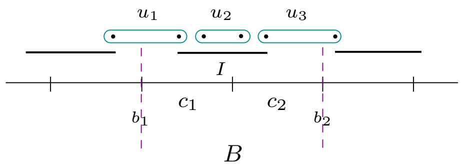

Fig. 2. Illustration of Subcase 1.3.

– Subcase 1.2: $\langle b_1, b_2\rangle = \langle 1, 0\rangle$. When p_0 arrives, a new cluster u is opened, since p_0 is in c_2, the right border is closed, and w contains < 1 cluster at the time so that the condition in Line 5 fails. Again, all subsequent points in I are put in the same cluster, and points from the neighboring windows cannot join u and make crossing clusters. Hence, $\mu(B) = 1$.

– Subcase 1.3: $\langle b_1, b_2\rangle = \langle 0, 1\rangle$. We show that $\mu(B) < 2$. Suppose by contradiction that $\mu(B) = 2$. By Observation 1(i), I cannot contain two clusters entirely. Therefore, the only way to get $\mu(B) = 2$ is that I intersects three clusters $\langle u_1, u_2, u_3\rangle$ (from left to right, as always), where u_1 and u_3 are crossing clusters, and u_2 is entirely contained in I (see Fig. 2). By a similar argument as in the proof of Observation 1(iii), u_2 is opened after u_1 and u_3 have become crossing clusters. Let p_1 be the first point of u_1 in w, and p_2 be the first point of u_1 in the neighboring window. We have two scenarios:
  - Subsubcase 1.3.1: $p_1 \ll p_2$. In this case, cluster u_1 is opened for p_1. But p_2 cannot be put in u_1, because upon arrival of p_2, w contains < 2 clusters, and thus, the condition in line 5 does not hold.
  - Subsubcase 1.3.2: $p_2 \ll p_1$. Here, cluster u_1 is opened for p_2. But p_1 cannot be put in u_1, because upon arrival of p_1, w contains < 1 cluster, and hence, the condition in line 5 does not hold.

  Both scenarios leads to contradiction. Therefore, $\mu(B) \leq 3/2$.

– Subcase 1.4: $\langle b_1, b_2\rangle = \langle 1, 1\rangle$. Here, Lemma 2(ii) implies that $\mu(B) \leq 2$.

Since each of the four subcases occurs with probability 1/4, we conclude that the expected value of $\mu(B)$ in Case 1 is at most $\frac{1}{4}(1 + 1 + \frac{3}{2} + 2) = \frac{11}{8}$.

Case 2: B is split between two neighboring windows. Let b be the single border indicator inside B. Let $\mu_0(B)$ and $\mu_1(B)$ represent the value of $\mu(B)$ for the case that b is set to 0 and 1, respectively. It is clear by Lemma 2(ii) that $\mu_1(B) \leq 2$. We rule out two possibilities:

– Subcase 2.1: $\mu_0(B) = 3$. Since I cannot contain both a whole cluster and a crossing cluster by Observation 2(i), the only possible scenario is that c_1 intersects two clusters $\langle u_1, u_2\rangle$, and c_2 intersects two clusters $\langle u_3, u_4\rangle$, where u_1 and u_4 are crossing clusters, and u_2 and u_3 are whole clusters. Let p_1 be the first point in u_2 and p_2 be the first point in u_3. Suppose w.l.o.g. that $p_1 \ll p_2$. By Observation 1(iii), p_1 arrives after u_1 has been opened, and p_2 arrives after u_4 has been opened. But when p_2 arrives, the window containing it contains one cluster, u_4, and the neighboring window contains two clusters u_1 and u_2. Therefore, p_2 would be assigned to u_2 by Line 5 instead: a contradiction.

- Subcase 2.2: $\mu_0(B) = 5/2$ and $\mu_1(B) = 2$. Suppose that $\mu_1(B) = 2$. Then I intersects three clusters $\langle u_1, u_2, u_3 \rangle$, where u_1 and u_3 are crossing clusters, and u_2 is completely contained in I. Let t be the time at which u_1 becomes a crossing cluster, and let $\sigma(t)$ be the subset of input points coming up to time t. By a similar argument as in the proof of Observation 1(iii), any point in $I \cap c_1$ not contained in u_1 arrives after time t. Therefore, upon receiving the input sequence $\sigma(t)$, u_1 becomes a crossing cluster no matter whether the border between c_1 and c_2 is open or closed. Using the same argument we conclude that u_3 becomes a crossing cluster regardless of the value of b. Now consider the case where $b = 0$. Since both u_1 and u_3 remain crossing clusters, $\mu_0(B)$ must be an integer (1, 2, or 3) and cannot equal 5/2.

Ruling out these two subcases, we have $\mu_0(B) + \mu_1(B) \leq 4$ in all remaining subcases, and therefore, the expected value of $\mu(B)$ in this case is at most 2.

Since each of Cases 1 and 2 occurs with probability 1/2, we conclude that $\rho(2) \leq \frac{1}{2}(\frac{11}{8}) + \frac{1}{2}(2) = \frac{27}{16}$. (This bound is tight: to see this just consider the block $B = [2, 4)$, and the sequence of 8 points $\langle 1.5, 2.5, 0.5, 3.5, 4.5, 2.7, 3.2, 5.5 \rangle$ for which $E[\mu(B)] = \frac{27}{16}$.) $\square$

Theorem 2. $\rho(3) \leq 17/8$.

Proof. Consider a block B of size 3, consisting of cells $\langle c_1, c_2, c_3 \rangle$, and let b be the single border indicator strictly inside B. We assume w.l.o.g. that c_1 and c_2 fall in the same window (the other scenario is symmetric). We consider two cases.

- Case 1: $b = 0$. We rule out the following possibilities.
  - Subcase 1.1: $\mu(c_2) = 2$. Impossible by Lemma 1.
  - Subcase 1.2: $\mu(c_1) = \mu(c_3) = 2$. Impossible by Lemma 1.
  - Subcase 1.3: $\mu(c_1) = 2$ and $\mu(c_2) = \mu(c_3) = 3/2$. Here, B intersects six clusters $\langle u_1, \ldots, u_6 \rangle$, where u_1, u_3, u_6 are crossing clusters and u_2, u_4, u_5 are whole clusters. Let $\langle I_1, I_2 \rangle$ be the two unit intervals in B in $\texttt{opt}(\sigma)$. By Observation 2(i), u_3 cannot be entirely contained in I_1. This implies that $u_4 \cup u_5 \subset I_2$. Now suppose w.l.o.g. that u_4 is opened after u_5. By Observation 1(iii), u_4 is the last to be opened after u_3, u_5, u_6. Consider any point p in u_4. Upon arrival of p, the window containing p contains at least one cluster, u_3, and the neighboring window contains two clusters u_5 and u_6. Therefore, by the condition in Line 5, the algorithm would assign p to u_5 instead of u_4, which is a contradiction.
  - Subcase 1.4: $\mu(c_1) = \mu(c_2) = 3/2$ and $\mu(c_3) = 2$. Similarly impossible.

  In all remaining subcases, $\mu(B)$ is at most $2 + \frac{3}{2} + 1 = \frac{9}{2}$ or $\frac{3}{2} + \frac{3}{2} + \frac{3}{2} = \frac{9}{2}$.
- Case 2: $b = 1$. Here, Lemma 2(ii) implies that $\mu(B) \leq 4$.

Each of Cases 1 and 2 occurs with probability 1/2, therefore $\rho(3) \leq \frac{1}{2}(4 + \frac{9}{2})/2 = 17/8$. $\square$

Theorem 3. $\rho(4) \le 53/24$.

Proof. Consider a block B of size 4. We consider two easy cases.

- CASE 1: B falls completely in two windows. Let b be the single border indicator strictly inside B. Now, if $b = 1$, $\mu(B) \le 6$ by Lemma 2(ii), otherwise, $\mu(B) \le 7$ by Lemma 2(i). Therefore, the expected cost in this case is at most $\frac{1}{2}(6 + 7) = \frac{13}{2}$.
- CASE 2: B is split between three consecutive windows. Let $\langle b_1, b_2 \rangle$ be the two border indicators inside B. For the subcase where $\langle b_1, b_2 \rangle = \langle 1, 1 \rangle$ the cost is at most 6 by Lemma 2(ii), and for the remaining 3 subcases, the cost of B is at most 7 by Lemma 2(i). Thus, the expected cost in this case is at most $\frac{1}{4}(6) + \frac{3}{4}(7) = \frac{27}{4}$.

Since each of Cases 1 and 2 occurs with probability exactly 1/2, we conclude that $\rho(4) \le \frac{1}{2}(\frac{13}{2} + \frac{27}{4})/3 = \frac{53}{24}$. $\square$

Theorem 4. $\rho(k) \le (2k - 1)/(k - 1)$ *for all* $k \ge 5$.

Proof. This is a direct implication of Lemma 2(i). $\square$

5 The Combined Algorithm

The RandBorder algorithm as shown in the previous section has competitive ratio greater than 2 on blocks of size three and more. To overcome this deficiency, we need to combine RandBorder with another algorithm that works well for larger block sizes. A good candidate for this is the naïve grid algorithm:

Grid Algorithm: For each new point p, if the grid cell containing p contains a cluster, then put p in that cluster, else open a new cluster for p.

It is easy to verify that the Grid algorithm uses exactly k clusters on a block of size k. Therefore, the competitive ratio of this algorithm within a block of size k is $k/(k - 1)$. We can now randomly combine the RandBorder algorithm with the Grid algorithm to obtain an expected competitive ratio strictly less than 2.

Combined Algorithm: With probability 8/15 run RandBorder, and with probability 7/15 run Grid.

Theorem 5. *The competitive ratio of the Combined algorithm is at most* 11/6 *against oblivious adversaries.*

Proof. The competitive ratios of RandBorder and Grid within blocks of size 2 are 27/16 and 2, respectively. Therefore, the expected competitive ratio of the Combined algorithm is $\frac{8}{15}(\frac{27}{16}) + \frac{7}{15}(2) = \frac{11}{6}$ within a block of size 2. For larger block sizes, the expected competitive ratio of Combined is always at most 11/6, as shown in Table 1. By summing over all blocks and exploiting the additivity of our cost function $\mu(\cdot)$, we see that the expected total cost of the solution produced by Combined is at most 11/6 times the size of $\mathtt{opt}(\sigma)$ for every input sequence σ. $\square$

Table 1. The competitive ratio of the algorithms within a block.

Block Size	Grid	RandBorder	Combined
2	2	27/16	11/6
3	3/2	$\leq \frac{17}{8}$	$\leq 11/6$
4	4/3	$\leq \frac{53}{24}$	$\leq 9/5$
$k \geq 5$	$\frac{k}{k-1}$	$\leq \frac{2k-1}{k-1}$	$\leq \frac{23k-8}{15(k-1)}$

Remarks. Currently only a 4/3 randomized lower bound and a 3/2 deterministic lower bound are known for the one-dimensional problem [1]. Also, as a corollary to Theorem 5, we immediately get an upper bound of $(\frac{11}{12}) \cdot 2^d$ for the d-dimensional unit clustering problem under the L_∞ metric [1].

References

1. T. M. Chan and H. Zarrabi-Zadeh. A randomized algorithm for online unit clustering. In *Proc. 4th Workshop on Approx. and Online Algorithms*, volume 4368 of *Lecture Notes Comput. Sci.*, pages 121–131, 2006.
2. M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental clustering and dynamic information retrieval. *SIAM J. Comput.*, 33(6):1417–1440, 2004.
3. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, 2nd edition, 2001.
4. D. Fotakis. Incremental algorithms for facility location and k-median. In *Proceedings of the 12th Annual European Symposium on Algorithms*, volume 3221 of *Lecture Notes Comput. Sci.*, pages 347–358, 2004.
5. R. J. Fowler, M. S. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Inform. Process. Lett.*, 12(3):133–137, 1981.
6. T. Gonzalez. Covering a set of points in multidimensional space. *Inform. Process. Lett.*, 40:181–188, 1991.
7. A. Gyárfás and J. Lehel. On-line and First-Fit colorings of graphs. *J. Graph Theory*, 12:217–227, 1988.
8. D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32:130–136, 1985.

9. H. A. Kierstead and J. Qin. Coloring interval graphs with First-Fit. *SIAM J. Discrete Math.*, 8:47–57, 1995.
10. A. Meyerson. Online facility location. In *Proc. 42nd IEEE Sympos. Found. Comput. Sci.*, pages 426–433, 2001.
11. F. Nielsen. Fast stabbing of boxes in high dimensions. *Theoret. Comput. Sci.*, 246:53–72, 2000.
12. S. L. Tanimoto and R. J. Fowler. Covering image subsets with patches. In *Proc. 5th International Conf. on Pattern Recognition*, pages 835–839, 1980.