Online Unit Clustering in Higher Dimensions

Adrian Dumitrescu* Csaba D. Tóth†

Tuesday 8th August, 2017

Abstract

We revisit the online Unit Clustering problem in higher dimensions: Given a set of n points in $\mathbb{R}^d$, that arrive one by one, partition the points into clusters (subsets) of diameter at most one, so as to minimize the number of clusters used. In this paper, we work in $\mathbb{R}^d$ using the L_∞ norm. We show that the competitive ratio of any algorithm (deterministic or randomized) for this problem must depend on the dimension d. This resolves an open problem raised by Epstein and van Stee (WAOA 2008). We also give a randomized online algorithm with competitive ratio $O(d^2)$ for Unit Clustering of integer points (i.e., points in $\mathbb{Z}^d$, $d \in \mathbb{N}$, under L_∞ norm). We complement these results with some additional lower bounds for related problems in higher dimensions.

Keywords: online algorithm, unit covering, unit clustering, competitive ratio, greedy algorithm.

1 Introduction

Covering and clustering are ubiquitous problems in the theory of algorithms, computational geometry, optimization, and others. Such problems can be asked in any metric space, however this generality often restricts the quality of the results, particularly for online algorithms. Here we study lower bounds for several such problems in a high dimensional Euclidean space and mostly in the L_∞ norm. We first consider their *offline* versions.

Problem 1. k-Center. *Given a set of n points in $\mathbb{R}^d$ and an integer k, cover the set by k congruent balls centered at the points so that the diameter of the balls is minimized.*

The following two problems are dual to Problem 1.

Problem 2. Unit Covering. *Given a set of n points in $\mathbb{R}^d$, cover the set by balls of unit diameter so that the number of balls is minimized.*

Problem 3. Unit Clustering. *Given a set of n points in $\mathbb{R}^d$, partition the set into clusters of diameter at most one so that number of clusters is minimized.*

Problems 1 and 2 are easily solved in polynomial time for points on the line, i.e., for $d = 1$; however, both problems become NP-hard already in the Euclidean plane [17, 22]. Factor 2 approximations are known for k-Center in any metric space (and so for any dimension) [16, 18];

*Department of Computer Science, University of Wisconsin–Milwaukee, WI, USA. Email: dumitres@uwm.edu.

†Department of Mathematics, California State University Northridge, Los Angeles, CA; and Department of Computer Science, Tufts University, Medford, MA, USA. Email: cdtoth@acm.org.

see also [23, Ch. 5], [24, Ch. 2], while polynomial-time approximation schemes are known for Unit Covering for any fixed dimension [20].

Problems 2 and 3 look similar; indeed, one can go from clusters to balls in a straightforward way; and conversely one can assign multiply covered points to unique balls. As such, the two problems are identical in the offline setting.

We next consider their *online* versions. In this paper we focus on problems 2 and 3 in particular. It is worth emphasizing two common properties: (i) a point assigned to a cluster must remain in that cluster; and (ii) two distinct clusters cannot merge into one cluster, i.e., the clusters maintain their identities.

The performance of an online algorithm ALG is measured by comparing it to an optimal offline algorithm OPT using the standard notion of competitive ratio [5, Ch. 1]. The competitive ratio of ALG is defined as $\sup_\sigma \frac{\mathsf{ALG}(\sigma)}{\mathsf{OPT}(\sigma)}$, where σ is an input sequence of request points, $\mathsf{OPT}(\sigma)$ is the cost of an optimal offline algorithm for σ and $\mathsf{ALG}(\sigma)$ denotes the cost of the solution produced by ALG for this input. For randomized algorithms, $\mathsf{ALG}(\sigma)$ is replaced by the expectation $E[\mathsf{ALG}(\sigma)]$, and the competitive ratio of ALG is $\sup_\sigma \frac{E[\mathsf{ALG}(\sigma)]}{\mathsf{OPT}(\sigma)}$. Whenever there is no danger of confusion, we use ALG to refer to an algorithm or the cost of its solution, as needed.

Charikar et al. [8] have studied the online version of Unit Covering. The points arrive one by one and each point needs to be assigned to a new or to an existing unit ball upon arrival; the L_2 metric is used in $\mathbb{R}^d$, $d \in \mathbb{N}$. The location of each new ball is fixed as soon as it is opened. The authors provided a deterministic algorithm of competitive ratio $O(2^d d \log d)$ and gave a lower bound of $\Omega(\log d / \log\log\log d)$ on the competitive ratio of any deterministic online algorithm for this problem. Tight bounds of 2 and 4 are known for $d = 1$ and $d = 2$, respectively.

Chan and Zarrabi-Zadeh [7] introduced the online Unit Clustering problem at WAOA 2006. While the input and the objective of this problem are identical to those for Unit Covering, this latter problem is more flexible in that the algorithm is not required to produce unit balls at any time, but rather the smallest enclosing ball of each cluster should have diameter *at most* 1; moreover, the ball may change (grow or shift) in time. The L_∞ metric is used in $\mathbb{R}^d$, $d \in \mathbb{N}$. The authors showed that several standard approaches for Unit Clustering, namely the deterministic algorithms `Centered`, `Grid`, and `Greedy`, all have competitive ratio at most 2 for points on the line ($d = 1$). Moreover, the first two algorithms above are applicable for Unit Covering, with a competitive ratio at most 2 for $d = 1$, as well.

In fact, Chan and Zarrabi-Zadeh [7] showed that no online algorithm (deterministic or randomized) for Unit Covering can have a competitive ratio better than 2 in one dimension ($d = 1$). They also showed that it is possible to get better results for Unit Clustering than for Unit Covering. Specifically, they developed the first algorithm with competitive ratio below 2 for $d = 1$, namely a randomized algorithm with competitive ratio 15/8. Moreover, they developed a general method to achieve competitive ratio below 2^d in $\mathbb{R}^d$ under L_∞ metric for any $d \geq 2$, by "lifting" the one-dimensional algorithm to higher dimensions. In particular, the existence of an algorithm for Unit Clustering with competitive ratio ρ_1 for $d = 1$ yields an algorithm with competitive ratio $\rho_d = 2^{d-1}\rho_1$ for every $d \geq 2$ for this problem. The current best competitive ratio for Unit Clustering in $\mathbb{R}^d$ is obtained in exactly this way: the current best ratio 5/3, for $d = 1$, is due to Ehmsen and Larsen [13], and this gives a ratio of $2^{d-1}\frac{5}{3}$ for every $d \geq 2$.

As such, the 1-dimensional case becomes quite important since no other significantly faster method is currently available for dealing with the problem in higher dimensions. It is however worth noting that `Algorithm Grid` is easily extendable to $\mathbb{R}^d$. In higher dimensions, its competitive ratio is only marginally worse than that obtained by "lifting" the one-dimensional algorithm to higher dimensions.

`Algorithm Grid.` Build a uniform grid in $\mathbb{R}^d$ where cells are unit cubes of the form $\prod [i_j, i_j + 1)$, where $i_j \in \mathbb{Z}$ for $j = 1, \ldots, d$. For each new point p, if the grid cell containing p is nonempty, put p in the corresponding cluster; otherwise open a new cluster for the grid cell and put p in it.

Since in $\mathbb{R}^d$ each cluster of OPT can be split to at most 2^d grid-cell clusters created by the algorithm, its competitive ratio is at most 2^d, and this analysis is tight.

The 15/8 ratio [7] has been subsequently reduced to 11/6 by the same authors [25]; that algorithm is still randomized. Epstein and van Stee [15] gave the first deterministic algorithm with ratio below 2, namely one with ratio 7/4, and further improving the earlier 11/6 ratio. In the latest development, Ehmsen and Larsen [13] provided a deterministic algorithm with competitive ratio 5/3, which holds the current record in both categories.

From the other direction, the lower bound for deterministic algorithms has evolved from 3/2 in [7] to 8/5 in [15], and then to 13/8 in [21]. Whence the size of the current gap for the competitive ratio of deterministic algorithms for the one-dimensional case of Unit Clustering is quite small, namely $\frac{5}{3} - \frac{13}{8} = \frac{1}{24}$, but remains nonzero. The lower bound for randomized algorithms has evolved from 4/3 in [7] to 3/2 in [15].

For points in the plane (i.e., $d = 2$), the lower bound for deterministic algorithms has evolved from 3/2 in [7] to 2 in [15], and then to 13/6 in [13]. The lower bound for randomized algorithms has evolved from 4/3 in [7] to 11/6 in [15].

As such, the best lower bounds on the competitive ratio for $d \geq 2$ prior to our work are 13/6 for deterministic algorithms [13] and 11/6 for randomized algorithms [15].

Notation and terminology. Throughout this paper the L_∞-norm is used. Then the Unit Clustering problem is to partition a set of points in $\mathbb{R}^d$ into clusters (subsets), each contained in a unit cube, i.e., a cube of the form $\mathbf{x} + [0,1]^d$ for some $\mathbf{x} \in [0,1]^d$, so as to minimize the number of clusters used. $\mathbb{E}[X]$ denotes the expected value of a random variable X.

Contributions. We obtain the following results:

(i) The competitive ratio of every online algorithm (deterministic or randomized) for Unit Clustering in $\mathbb{R}^d$ under L_∞ norm is $\Omega(d)$ for every $d \geq 2$ (Theorem 1 in Section 2). We thereby give a positive answer to a question of Epstein and van Stee; specifically, they asked whether the competitive ratio grows with the dimension [15, Sec. 4]. The question was reposed in [13, Sec. 7].

(ii) The competitive ratio of every online algorithm (deterministic or randomized) for Unit Covering in $\mathbb{R}^d$ under L_∞ norm is at least $d + 1$ for every $d \geq 1$ (Theorem 2 in Section 3). This generalizes a result of Chan and Zarrabi-Zadeh [7] from $d = 1$ to higher dimensions. We also give a randomized algorithm with competitive ratio $O(d^2)$ for Unit Covering in $\mathbb{Z}^d$, $d \in \mathbb{N}$, under L_∞ norm (Theorem 3 in Section 3). The algorithm applies to Unit Clustering in $\mathbb{Z}^d$, $d \in \mathbb{N}$, with the same competitive ratio.

(iii) The competitive ratio of `Algorithm Greedy` for Unit Clustering in $\mathbb{R}^d$ under L_∞ norm is unbounded for every $d \geq 2$ (Theorem 4 in Section 4). The competitive ratio of `Algorithm Greedy` for Unit Clustering in $\mathbb{Z}^d$ under L_∞ norm is at least 2^{d-1} and at most $2^{d-1} + \frac{1}{2}$ for every $d \geq 2$ (Theorem 5 in Section 4).

Related work. Several other variants of Unit Clustering have been studied in [14]. A survey of algorithms for Unit Clustering in the context of online algorithms appears in [9]. Clustering with variable sized clusters has been studied in [10, 11]. Grid-based online algorithms for clustering problems have been developed by the same authors [12].

Unit Covering is a variant of Set Cover. Alon et al. [1] gave a deterministic online algorithm of competitive ratio $O(\log m \log n)$ for Set Cover, where n is the number of possible points (the size of the ground set) and m is the number of sets in the family. If every element appears in at most Δ sets, the competitive ratio of the algorithm can be improved to $O(\log \Delta \log n)$. Buchbinder and Naor [6] improved these competitive ratio to $O(\log m \log(n/\mathsf{OPT}))$ and $O(\log \Delta \log (n/\mathsf{OPT}))$, respectively, under the same assumptions. For several combinatorial optimization problems (e.g., covering and packing), the classic technique that rounds a fractional linear programming solution to an integer solution has been adapted to the online setting [2, 3, 4, 6, 19].

In these results, the underlying set system for the covering and packing problem must be finite: The online algorithms and their analyses rely on the size of the ground set. For Unit Clustering and Unit Clustering over infinite sets, such as $\mathbb{R}^d$ or $\mathbb{Z}^d$, these techniques could only be used after a suitable discretization and a covering of the domain with finite sets, and it is unclear whether they can beat the trivial competitive ratio of 2^d in a substantive way.

2 Lower bounds for online Unit Clustering

In this section, we prove the following theorem.

Theorem 1. *The competitive ratio of every (i) deterministic algorithm (with an adaptive deterministic adversary), and (ii) randomized algorithm (with a randomized oblivious adversary), for* Unit Clustering *in $\mathbb{R}^d$ under L_∞ norm is $\Omega(d)$ for every $d \geq 1$.*

Proof. Let ϱ be the competitive ratio of an online algorithm. We may assume $\varrho \leq d$, otherwise there is nothing to prove. We may also assume that $d \geq 4$ since this is the smallest value for which the argument gives a nontrivial lower bound. Let K be a sufficiently large even integer (that depends on d).

Deterministic Algorithm. We first prove a lower bound for a deterministic algorithm, assuming an adaptive deterministic adversary. We present a total of $\lfloor d/2 \rfloor K^d$ points to the algorithm, and show that it creates $\Omega(d \cdot \mathsf{OPT})$ clusters, where OPT is the offline minimum number of clusters for the final set of points. Specifically, we present the points to the algorithm in $\lfloor d/2 \rfloor$ rounds. Round $i = 1, \ldots, \lfloor d/2 \rfloor$ consists of the following three events:

(i) The adversary presents (inserts) a set S_i of K^d points; S_i is determined by a vector $\sigma(i) \in \{-1, 0, 1\}^d$ to be later defined.

(ii) The algorithm may create new clusters or expand existing clusters to cover S_i.

(iii) If $i < \lfloor d/2 \rfloor$, the adversary computes $\sigma(i+1)$ from the clusters that cover S_i.

In the first round, the adversary presents points of the integer lattice; namely $S_1 = [K]^d$, where $[K] = \{x \in \mathbb{Z} : 1 \leq x \leq K\}$. In round $i = 2, \ldots, \lfloor d/2 \rfloor$, the point set S_i will depend on the clusters created by the algorithm in previous rounds. We say that a cluster *expires in round i* if it contains some points from S_i but no additional points can (or will) be added to it in any subsequent round. We show that over $\lfloor d/2 \rfloor$ rounds, $\Omega(d \cdot \mathsf{OPT})$ clusters expire, which readily implies $\varrho = \Omega(d)$.

Optimal solutions. For $i = 1, \ldots, \lfloor d/2 \rfloor$, denote by OPT_i the offline optimum for the set $\bigcup_{j=1}^{i} S_j$ of points presented up to (and including) round i. Since $S_1 = [K]^d$ and K is even, $\mathsf{OPT}_1 = K^d/2^d$. The optimum solution for S_1 is unique, and each cluster in the optimum is a Cartesian product $\prod_{i=1}^{d}\{a_i, a_i+1\}$, where $a_i \in [K]$ is odd for $i = 1, \ldots, d$ (Fig. 1(a)).

Consider $2^d - 1$ additional near-optimal solutions for S_1 obtained by translating the optimal clusters by a d-dimensional $0-1$ vector, and adding new clusters along the boundary of the cube $[K]^d$. We shall argue that the points inserted in round i, $i \geq 2$, can be added to some but not all of these solutions. To make this precise, we define these solutions a bit more carefully. First we define an infinite set of hypercubes

$$\mathcal{Q} = \left\{ \prod_{i=1}^{d} [a_i, a_i+1] : a_i \in \mathbb{Z} \text{ is odd for } i = 1, \ldots, d \right\}.$$

For a point set $S \subset \mathbb{R}^d$ and a vector $\tau \in \{0,1\}^d$, let the clusters be the subsets of S that lie in translates $Q + \tau$ of hypercubes $Q \in \mathcal{Q}$, that is, let

$$C(S, \tau) = \{S \cap (Q + \tau) : Q \in \mathcal{Q}\}.$$

Since S_1 is an integer grid, the clusters $C(S_1, \tau)$ contain all points in S_1 for all $\tau \in \{0,1\}^d$. See Fig. 1(a–d) for examples. Due to the boundary effect, the number of clusters in $C(S_1, \tau)$ is

$$\frac{K^d + O(dK^{d-1})}{2^d} = \mathsf{OPT}_1 \cdot \left(1 + O\left(\frac{d}{K}\right)\right) = (1 + o(1))\,\mathsf{OPT}_1,$$

if K is sufficiently large with respect to d.

In round $i = 2, \ldots, \lfloor d/2 \rfloor$, the point set S_i is a perturbation of the integer grid S_1 (as described below). Further, we ensure that the final point set $S = \bigcup_{i=1}^{\lfloor d/2 \rfloor} S_i$ is covered by the clusters $C(S, \tau)$ for at least one vector $\tau \in \{0,1\}^d$. Consequently,

$$\mathsf{OPT}_i = \mathsf{OPT}_1(1 + o(1)) = (1 + o(1))\frac{K^d}{2^d}, \text{ for all } i = 1, \ldots, \lfloor d/2 \rfloor.$$

At the end, we have $\mathsf{OPT} = \mathsf{OPT}_{\lfloor d/2 \rfloor} = (1 + o(1))\frac{K^d}{2^d}$.

Perturbation. A perturbation of the integer grid S_1 is encoded by a vector $\sigma \in \{-1, 0, 1\}^d$, that we call the *signature* of the perturbation. Let $\varepsilon \in (0, \frac{1}{2})$. For an integer point $p = (p_1, \ldots, p_d) \in S_1$ and a signature σ, the perturbed point p' is defined as follows; see Fig. 1(e–h) for examples in the plane: For $j = 1, \ldots, d$, let p'_j be

- p_j when $\sigma_j = 0$;
- $p_j + \varepsilon$ if p_j is odd, and $p_j - \varepsilon$ if p_j is even when $\sigma_j = -1$;
- $p_j - \varepsilon$ if p_j is odd, and $p_j + \varepsilon$ if p_j is even when $\sigma_j = 1$.

For $i = 2, \ldots, \lfloor d/2 \rfloor$, the point set S_i is a perturbation of S_1 with signature $\sigma(i)$, for some $\sigma(i) \in \{-1, 0, 1\}^d$. The signature of S_1 is $\sigma(1) = (0, \ldots 0)$ (and so S_1 can be viewed as a null perturbation of itself). At the end of round $i = 1, \ldots, \lfloor d/2 \rfloor - 1$, we compute $\sigma(i+1)$ from $\sigma(i)$ and from the clusters that cover S_i. The signature $\sigma(i)$ determines the set S_i, for every $i = 2, \ldots, \lfloor d/2 \rfloor$. Note the following relation between the signatures $\sigma(i)$ and the clusters $C(S_i, \tau)$.

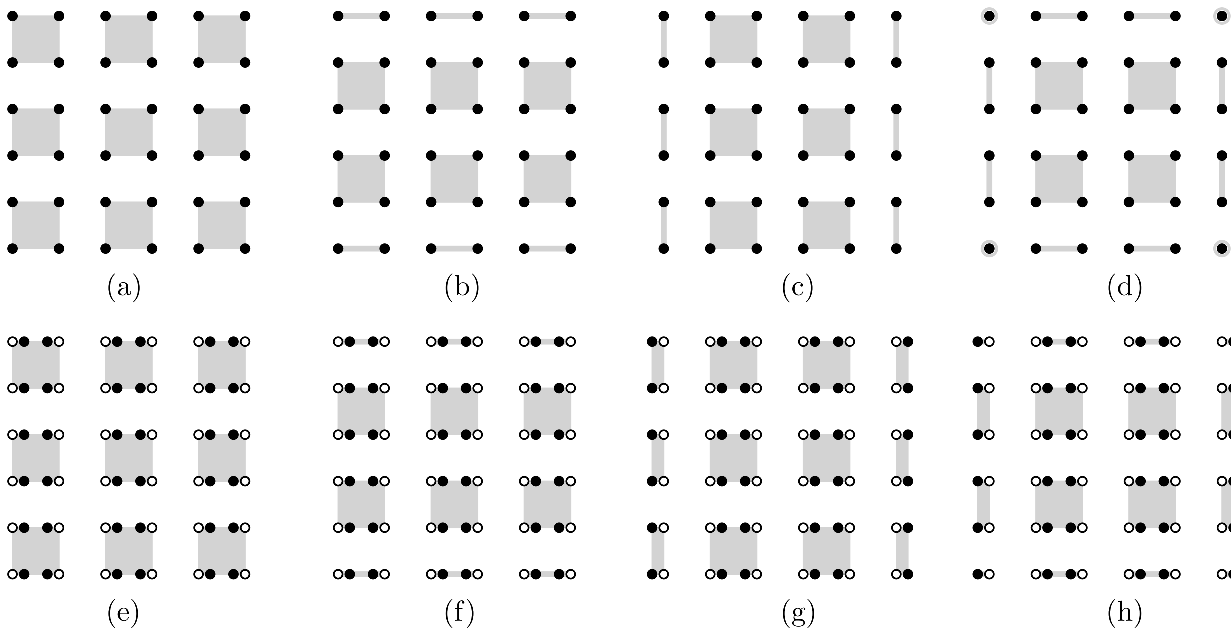

Figure 1: (a) A 6×6 section of the integer grid and $\mathsf{OPT}_1 = 9$ clusters. (b–d) Near-optimal solutions $C(S_1, \tau)$ for $\tau = (0,1)$, $(1,0)$, and $(1,1)$. (e–f) The perturbation with signature $\sigma = (-1,0)$, and clusters $C(S,\tau)$ for $\tau = (0,0)$ and $\tau = (0,1)$, where S is the union of the perturbed points (full dots), and grid points (empty circles). (g–h) The perturbation with signature $\sigma = (1,0)$ and clusters $C(S,\tau)$ for $\tau = (1,0)$ and $\tau = (1,1)$ and the same S.

Observation 1. *Consider a point set S_i with signature $\sigma(i) \in \{-1,0,1\}^d$. The clusters $C(S_i,\tau)$ cover S_i if and only if for all $j = 1, \dots d$,*

- $\sigma_j(i) = 0$, *or*
- $\sigma_j(i) = -1$ *and* $\tau_j = 0$, *or*
- $\sigma_j(i) = 1$ *and* $\tau_j = 1$.

It follows from Observation 1 that the final point set $S = \bigcup_{i=1}^{\lfloor d/2 \rfloor} S_i$ is covered by the clusters $C(S,\tau)$ for at least one vector $\tau \in \{0,1\}^d$.

Adversary strategy. At the end of round $i = 1, \dots, \lfloor d/2 \rfloor - 1$, we compute $\sigma(i+1)$ from $\sigma(i)$ by changing a 0-coordinate to -1 or $+1$. Note that every point in S_i, $i = 1, 2, \dots, \lfloor d/2 \rfloor$, has $i-1$ perturbed coordinates and $d-i+1$ unperturbed coordinates. For all points in S_i, all unperturbed coordinates are integers. The algorithm covers S_i with at most $\varrho \cdot \mathsf{OPT}_i$ clusters. Project these clusters to the subspace $\mathbb{Z}^{d+1-i}$ corresponding to the unperturbed coordinates. We say that a cluster is

- *small* if its projection to $\mathbb{Z}^{d+1-i}$ contains at most $2^{d-i}/\varrho$ points, and
- *big* otherwise.

Note that we distinguish small and big clusters in round i based on how they cover the set S_i (in particular, a small cluster in round i may become large in another round, or vice versa).

Since the L_∞-diameter of a cluster is at most 1, a small cluster contains at most $(2^{d-i}/\varrho) \cdot 2^{i-1} = 2^d/(2\varrho)$ points of S_i (by definition, it contains at most $2^{d-i}/\varrho$ points in the projection to $\mathbb{Z}^{d+1-i}$, each of these points is the projection of K^{i-1} points of S_i; since S_i is a perturbation of the integer grid, any cluster contains at most 2^{i-1} of these preimages). The total number of points in S_i that lie in small clusters is at most

$$(\varrho \cdot \mathsf{OPT}_i)\, \frac{2^d}{2\varrho} = \mathsf{OPT}_i \cdot 2^{d-1} = \left(\frac{1}{2} + o(1)\right) K^d.$$

Consequently, the remaining $\left(\frac{1}{2} - o(1)\right) K^d$ points in S_i are covered by big clusters. For a big cluster C, let $s(C)$ denote the number of unperturbed coordinates in which its extent is 1. Then the number of points in C satisfies

$$\begin{aligned} 2^{d-i}/\varrho &\leq 2^{s(C)} \\ d - i - \log_2 \varrho &\leq s(C). \end{aligned}$$

We say that a big cluster C *expires* if no point can (or will) be added to C in the future. Consider the following experiment: choose one of the zero coordinates of the signature $\sigma(i)$ uniformly at random (i.e., all $d+1-i$ choices are equally likely), and change it to -1 or $+1$ with equal probability 1/2. If the j-th extent of a cluster C is 1, then it cannot be expanded in dimension j. Consequently, a big cluster C expires with probability at least

$$\frac{s(C)}{d+1-i} \cdot \frac{1}{2} = \frac{d - i - \log_2 \varrho}{2(d+1-i)} \geq \frac{d - \lfloor d/2 \rfloor - \log_2 d}{2d} = \Omega(1), \tag{1}$$

as $i \leq \lfloor d/2 \rfloor$ and we assume $\varrho \leq d$. It follows that there exists an unperturbed coordinate j, and a perturbation of the j-th coordinate such that

$$\Omega(1) \cdot \left(\frac{1}{2} - o(1)\right) \frac{K^d}{2^d} = \Omega(\mathsf{OPT})$$

big clusters expire in (at the end of) round $i = 1, \ldots, \lfloor d/2 \rfloor - 1$. The adversary makes this choice and the corresponding perturbation. In round $i = \lfloor d/2 \rfloor$, all clusters that cover any point in $S_{\lfloor d/2 \rfloor}$ expire, because no point will be added to any of these clusters. Since $S_{\lfloor d/2 \rfloor}$ is a perturbation of S_1, at least $\mathsf{OPT}_1 = \Omega(\mathsf{OPT})$ clusters expire in the last round, as well.

If a cluster expires in round i, then it contains some points of S_i but does not contain any point of S_j for $j > i$. Consequently, each cluster expire in at most one round, and the total number of expired clusters over all $\lfloor d/2 \rfloor$ rounds is $\Omega(d \cdot \mathsf{OPT})$. Since each of these cluster was created by the algorithm in one of the rounds, we have $\varrho \cdot \mathsf{OPT} = \Omega(d \cdot \mathsf{OPT})$, which implies $\varrho = \Omega(d)$, as claimed.

Randomized Algorithm. We modify the above argument to establish a lower bound of $\Omega(d)$ for a randomized algorithm with an oblivious randomized adversary. The adversary starts with the integer grid $S_1 = [K]^d$, with signature $\sigma(1) = \mathbf{0}$ as before. At the end of round $i = 1, \ldots, \lfloor d/2 \rfloor - 1$, it chooses an unperturbed coordinate of $\sigma(i)$ uniformly at random, and switches it to -1 or $+1$ with equal probability (independently of the clusters created by the algorithm) to obtain $\sigma(i+1)$. By (1), the expected number of big clusters that expire in round i, $1 \leq i < \lfloor d/2 \rfloor$, is $\Omega(\mathsf{OPT}_i) = \Omega(\mathsf{OPT})$; and all $(1-o(1))\mathsf{OPT}_{\lfloor d/2 \rfloor} = \Omega(\mathsf{OPT})$ big clusters expire in round $\lfloor d/2 \rfloor$. Consequently, the expected number of clusters created by the algorithm is $\Omega(d \cdot \mathsf{OPT})$, which implies $\varrho = \Omega(d)$, as required. $\square$

3 Lower bounds and algorithms for online Unit Covering

The following theorem extends a result from [7] from $d = 1$ to higher dimensions.

Theorem 2. *The competitive ratio of every deterministic online algorithm (with an adaptive deterministic adversary) for* Unit Covering *in* $\mathbb{R}^d$ *under* L_∞ *norm is at least* $d+1$ *for every* $d \geq 1$.

Proof. We construct an input sequence $p_1, \ldots, p_{d+1} \in \mathbb{Z}^d$ for which $\mathsf{OPT} = 1$ and $\mathsf{ALG} = d+1$ using an adaptive adversary (which knows the actions of the algorithm). We construct such a sequence inductively, so that

- each new point p_i requires a new cube, $Q_i \subset \mathbb{R}^d$, and
- all points presented can be covered by one integer unit cube incident to the origin.

Let $x_1, \ldots, x_d$ be the d coordinate axes in $\mathbb{R}^d$; and x_{d+1} be the new axis in $\mathbb{R}^{d+1}$. The induction basis is $d = 1$. We may assume for concreteness that $p_1 = 0$, and suppose that the algorithm opens a unit interval $[x, x+1]$ to cover this point. If $x = -1$, let $p_2 = 1$, else let $p_2 = -1$. The algorithm now opens a new unit interval to cover p_2. It is easily seen that $p_1, p_2 \in \mathbb{Z}$ and $\{p_1, p_2\}$ define a unit interval.

For the induction step, assume the existence of a sequence $\sigma = p_1, \ldots, p_{d+1} \in \mathbb{Z}^d$ that forces the algorithm to open a new unit cube, $Q_i \subset \mathbb{R}^d$, to cover each new point p_i, $i = 1, \ldots, d+1$ (and so $\mathsf{ALG} = d+1$), while $\mathsf{OPT} = 1$ with σ being covered by a single cube $U_d \subset \mathbb{Z}^d$. Present the following sequence of $d+2$ points to the algorithm in $\mathbb{R}^{d+1}$: $(p_1, 0), \ldots, (p_{d+1}, 0)$. The algorithm must use $d+1$ cubes, say, $Q_1, \ldots, Q_{d+1} \subset \mathbb{R}^{d+1}$ to cover these points. As such, the $d+1$ unit cubes $\pi(Q_1), \ldots, \pi(Q_{d+1}) \subset \mathbb{R}^d$, cover $p_1, \ldots, p_{d+1} \in \mathbb{Z}^d$, where $\pi(Q_i)$ is the projection onto the first d coordinates of Q_i; moreover, the unit cubes $\pi(Q_1), \ldots, \pi(Q_d)$ do not cover $(p_{d+1}, 0)$. Only $\pi(Q_d)$ contains p_{d+1}, but the cube Q_d cannot contain both $(p_{d+1}, -1)$ and $(p_{d+1}, 1)$. Consequently, $(p_{d+1}, -1)$ or $(p_{d+1}, 1)$ is not covered by $\bigcup_{i=1}^{d+1} Q_i$. The adversary presents such an uncovered point, which requires a new cube Q_{d+2}. (Note that the points $p_1, \ldots, p_{d+2}$ form a lattice path, where p_i and p_{i+1} differ in the $(i+1)$-th coordinate.) This completes the inductive step, and thereby the proof of the theorem. □

Remark. Since the proof of Theorem 2 uses integer points, the result also holds for Unit Covering in $\mathbb{Z}^d$ under L_∞ norm.

Online algorithm for Unit Covering over $\mathbb{Z}^d$. Note that the lower bound construction used sequences of integer points (i.e., points in $\mathbb{Z}^d$). We substantially improve on the trivial 2^d upper bound on the competitive ratio of Unit Covering over $\mathbb{Z}^d$ (or the $2^{d-1} + \frac{1}{2}$ upper bound of the greedy algorithm, see Section 4).

The online algorithm by Buchbinder and Naor [6] for Set Cover, for the unit covering problem over $\mathbb{Z}^d$, yields an algorithm with $O(d \log(n/\mathsf{OPT}))$ competitive ratio under the assumption that a set of n possible integer points is given in advance. Recently, Gupta and Nagarajan [19] gave an online randomized algorithm for a broad family of combinatorial optimization problems that can be expressed as sparse integer programs. For unit covering over the integers in $[n]^d$, their results yield a competitive ratio of $O(d^2)$, where $[n] = \{1, 2, \ldots, n\}$. The competitive ratio does not depend on n, but the algorithm must know n in advance.

We now remove the dependence on n so as to get a truly online algorithm for Unit Covering over $\mathbb{Z}^d$. Consider the following randomized algorithm.

`Algorithm Iterative Reweighing.` Let $P \subset \mathbb{Z}^d$ be the set of points presented to the algorithm and $\mathcal{C}$ the set of cubes chosen by the algorithm; initially $P = \mathcal{C} = \emptyset$. The algorithm chooses cubes for two different reasons, and it keeps them in sets $\mathcal{C}_1$ and $\mathcal{C}_2$, where $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$. It also maintains a third set of cubes, $\mathcal{B}$, for bookkeeping purposes;

initially $\mathcal{B} = \emptyset$. In addition, the algorithm maintains a weight function on all integer unit cubes. Initially $w(Q) = 2^{-(d+1)}$ for all integer unit cubes (this is the default value for all cubes that are disjoint from P).

We describe one iteration of the algorithm. Let $p \in \mathbb{Z}^d$ be a new point; put $P \leftarrow P \cup \{p\}$. Let $\mathcal{Q}(p)$ be the set of 2^d integer unit cubes that contain p.

1. If $p \in \bigcup \mathcal{C}$, then do nothing.
2. Else if $p \in \bigcup \mathcal{B}$, then let $Q \in \mathcal{B} \cap \mathcal{Q}(p)$ be an arbitrary cube and put $\mathcal{C}_1 \leftarrow \mathcal{C}_1 \cup \{Q\}$.
3. Else if $\sum_{Q \in \mathcal{Q}(p)} w(Q) \geq 1$, then let Q be an arbitrary cube in $\mathcal{Q}(p)$ and put $\mathcal{C}_2 \leftarrow \mathcal{C}_2 \cup \{Q\}$.
4. Else, the weights give a probability distribution on $\mathcal{Q}(p)$. Successively choose cubes from $\mathcal{Q}(p)$ at random with this distribution in $2d$ independent trials and add them to $\mathcal{B}$. Let $Q \in \mathcal{B} \cap \mathcal{Q}(p)$ be an arbitrary cube and put $\mathcal{C}_1 \leftarrow \mathcal{C}_1 \cup \{Q\}$. Double the weight of every cube in $\mathcal{Q}(p)$.

Theorem 3. *The competitive ratio of* `Algorithm Iterative Reweighing` *for* UNIT COVERING *in $\mathbb{Z}^d$ under L_∞ norm is $O(d^2)$ for every $d \in \mathbb{N}$.*

Proof. Suppose that a set P of n points is presented to the algorithm sequentially, and the algorithm created unit cubes in $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2$. Note that $\mathcal{C}_1 \subseteq \mathcal{B}$. We show that $\mathbb{E}[|\mathcal{B}|] = O(d^2 \cdot \mathsf{OPT})$ and $\mathbb{E}[|\mathcal{C}_2|] = O(\mathsf{OPT})$. This immediately implies that $\mathbb{E}[|\mathcal{C}|] \leq \mathbb{E}[|\mathcal{C}_1|] + \mathbb{E}[|\mathcal{C}_2|] \leq \mathbb{E}[|\mathcal{B}|] + \mathbb{E}[|\mathcal{C}_2|] = O(d^2 \cdot \mathsf{OPT})$.

First consider $\mathbb{E}[|\mathcal{B}|]$. New cubes are added to $\mathcal{B}$ in step 4. In this case, the algorithm places at most $2d$ cubes into $\mathcal{B}$, and doubles the weight of all 2^d cubes in $\mathcal{Q}(p)$ that contain p. Let $\mathcal{C}_{\mathsf{OPT}}$ be an offline optimum set of unit cubes. Each point $p \in P$ lies in some cube $Q_p \in \mathcal{C}_{\mathsf{OPT}}$. The weight of Q_p is initially $2^{-(d+1)}$, and it never exceeds 2; indeed, since $Q_p \in \mathcal{Q}(p)$, its weight before the last doubling must have been at most 1 in step 4 of the algorithm; thus its weight is doubled in at most $d+2$ iterations. Consequently, the algorithm invokes step 4 in at most $(d+2)\mathsf{OPT}$ iterations. In each such iteration, it adds at most $2d$ cubes to $\mathcal{B}$. Overall, we have $|\mathcal{B}| \leq (d+2) \cdot 2d \cdot \mathsf{OPT} = O(d^2 \cdot \mathsf{OPT})$, as required.

Next consider $\mathbb{E}[|\mathcal{C}_2|]$. A new cube is added to $\mathcal{C}_2$ in step 3. In this case, none of the cubes in $\mathcal{Q}(p)$ is in $\mathcal{B}$ and $\sum_{Q \in \mathcal{Q}(p)} w(Q) \geq 1$ when point p is presented, and the algorithm increments $|\mathcal{C}_2|$ by one. At the beginning of the algorithm, we have $\sum_{Q \in \mathcal{Q}(p)} w(Q) = \sum_{Q \in \mathcal{Q}(p)} 2^{-(d+1)} = 2^d \cdot 2^{-(d+1)} = 1/2$. Assume that the weights of the cubes in $\mathcal{Q}(p)$ were increased in t iterations, starting from the beginning of the algorithm, and the sum of weights of the cubes in $\mathcal{Q}(p)$ increases by $\delta_1, \ldots, \delta_t > 0$ (the weights of several cubes may have been doubled in an iteration). Since $\sum_{Q \in \mathcal{Q}(p)} w(Q) = 1/2 + \sum_{i=1}^{t} \delta_i$, then $\sum_{Q \in \mathcal{Q}(p)} w(Q) \geq 1$ implies $\sum_{i=1}^{t} \delta_i \geq 1/2$. For every $i = 1, \ldots, t$, the sum of weights of some cubes in $\mathcal{Q}(p)$, say, $\mathcal{Q}_i \subset \mathcal{Q}(p)$, increased by δ_i in step 4 of a previous iteration. Since the weights doubled, the sum of the weights of these cubes was δ_i at the beginning of that iteration, and the algorithm added one of them into $\mathcal{B}$ with probability at least δ_i in one random draw, which was repeated $2d$ times independently. Consequently, the probability that the algorithm did *not* add any cube from $\mathcal{Q}_i$ to $\mathcal{B}$ in that iteration is at most $(1 - \delta_i)^{2d}$. The probability that none of the cubes in $\mathcal{Q}(p)$ has been added to $\mathcal{B}$ before point p arrives is (by independence) at most

$$\prod_{i=1}^{t} (1 - \delta_i)^{2d} \leq e^{-2d \sum_{i=1}^{t} \delta_i} \leq e^{-d}.$$

The total number of points p for which step 3 applies is at most $|P|$. Since each unit cube contains at most 2^d points, we have $|P| \leq 2^d \cdot \mathsf{OPT}$. Therefore $\mathbb{E}[|\mathcal{C}_2|] \leq |P| e^{-d} \leq (2/e)^d\, \mathsf{OPT} \leq \mathsf{OPT}$, as claimed. □

The above algorithm applies to Unit Clustering of integer points in $\mathbb{Z}^d$ with the same competitive ratio:

Corollary 1. *The competitive ratio of* `Algorithm Iterative Reweighing` *for* Unit Clustering *in $\mathbb{Z}^d$ under L_∞ norm is $O(d^2)$ for every $d \in \mathbb{N}$.*

4 Lower bound for `Algorithm Greedy` for Unit Clustering

Chan and Zarrabi-Zadeh [7] showed that the greedy algorithm for Unit Clustering on the line ($d = 1$) has competitive ratio of 2 (this includes both an upper bound on the ratio and a tight example). Here we show that the competitive ratio of the greedy algorithm is unbounded. We first recall the algorithm:

> `Algorithm Greedy.` For each new point p, if p fits in some existing cluster, put p in such a cluster (break ties arbitrarily); otherwise open a new cluster for p.

Theorem 4. *The competitive ratio of* `Algorithm Greedy` *for* Unit Clustering *in $\mathbb{R}^d$ under L_∞ norm is unbounded for every $d \geq 2$.*

Proof. It suffices to consider $d = 2$; the construction extends to arbitrary dimensions $d \geq 2$. The adversary presents $2n$ points in pairs $\{(1 + i/n, i/n), (i/n, 1 + i/n)\}$ for $i = 0, 1, \ldots, n - 1$. Each pair of points spans a unit square that does not contain any subsequent point. Consequently, the greedy algorithm will create n clusters, one for each point pair. However, $\mathsf{OPT} = 2$ since the clusters $C_1 = \{(1 + i/n, i/n) : i = 0, 1, \ldots, n - 1\}$ and $C_2 = \{(i/n, 1 + i/n) : i = 0, 1, \ldots, n - 1\}$ are contained in the unit squares $[1, 2] \times [0, 1]$ and $[0, 1] \times [1, 2]$, respectively. $\square$

When we restrict `Algorithm Greedy` to integer points, its competitive ratio is exponential in d.

Theorem 5. *The competitive ratio of* `Algorithm Greedy` *for* Unit Clustering *in $\mathbb{Z}^d$ under L_∞ norm is at least 2^{d-1} and at most $2^{d-1} + \frac{1}{2}$ for every $d \geq 1$.*

Proof. We first prove the lower bound. Consider an integer input sequence implementing a a barycentric subdivision of the space, as illustrated in Fig. 2. Let K be a sufficiently large positive multiple of 4 (that depends on d). We present a point set S, where $|S| = (2 + o(1))(K/2)^d$ points to the algorithm, and show that it creates $(1 + o(1))\, 2^{d-1}\mathsf{OPT}$ clusters.

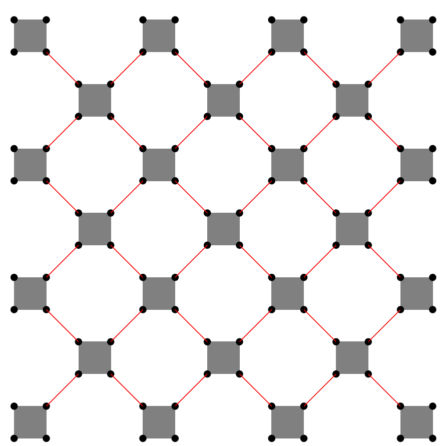

Figure 2: A planar instance for the greedy algorithm with $K = 12$; the edges in E are drawn in red.

Let $S = B \cup D$, where

$$\begin{aligned}
A &= \{(x_1, \ldots, x_d) \mid x_i \equiv 0 \pmod 4,\ 0 \le x_i \le K,\ i = 1, \ldots, d\}, \\
B &= A + \{0,1\}^d, \\
C &= \{(x_1, \ldots, x_d) \mid x_i \equiv 2 \pmod 4,\ 0 \le x_i \le K,\ i = 1, \ldots, d\}, \\
D &= C + \{0,1\}^d, \\
E &= \{\{u, v\} : u \in B, v \in D, ||u - v||_\infty \le 1\}.
\end{aligned}$$

Note that each element of C is the barycenter (center of mass) of 2^d elements of A, namely the vertices of a cell of $(4\mathbb{Z})^d$ containing the element. Here E is a set of pairs of lattice points (edges) that can be put in one-to-one correspondence with the points in D. As such, we have

$$\begin{aligned}
|A| &= \left(\frac{K}{4} + 1\right)^d,\ |B| = 2^d |A| = (1 + o(1))\,\frac{K^d}{2^d}, \\
|C| &= \left(\frac{K}{4}\right)^d,\ |D| = 2^d |C| = (1 + o(1))\,\frac{K^d}{2^d}, \\
|E| &= |D| = (1 + o(1))\,\frac{K^d}{2^d}, \\
\mathsf{OPT} &= |A \cup C| = |A| + |C| = (2 + o(1)) \left(\frac{K}{4}\right)^d.
\end{aligned}$$

It follows that $|E| = (1 + o(1))\, 2^{d-1}\mathsf{OPT}$. The input sequence presents the points in pairs, namely those in E. The greedy algorithm makes one new non-extendable cluster for each such "diagonal" pair (each cluster is a unit cube), so its competitive ratio is at least 2^{d-1} for every $d \ge 2$.

An upper bound of 2^d follows from the fact that each cluster in OPT contains at most 2^d integer points; we further reduce this bound. Let $\Gamma_1, \ldots, \Gamma_k$ be the clusters of an optimal partition ($k = \mathsf{OPT}$). Assume that the algorithm produces m clusters of size at least 2 and s singleton clusters. Since each cluster of OPT contains at most one singleton cluster created by the algorithm, we have

$$\begin{aligned}
\mathsf{ALG} = m + s &\le \frac{(k - s)2^d + s(2^d - 1)}{2} + s = \frac{k\, 2^d - s}{2} + s \\
&= k\, 2^{d-1} + \frac{s}{2} \le k\, 2^{d-1} + \frac{k}{2} = k\left(2^{d-1} + \frac{1}{2}\right),
\end{aligned}$$

as required. $\square$

5 Conclusion

Our results suggest several directions for future study. For instance, the gaps between the linear lower bounds and the exponential upper bounds in the competitive ratios for Unit Covering and Unit Clustering are intriguing. We conclude by listing a few specific questions of interest.

Problem 4. *Is there a lower bound on the competitive ratio for* Unit Covering *or* Unit Clustering *that is exponential in d? Is there a superlinear lower bound?*

Problem 5. *Do our lower bounds for* Unit Clustering *in $\mathbb{R}^d$ and $\mathbb{Z}^d$ under the L_∞ norm carry over to the L_2 norm (or the L_p norm for $1 \le p < \infty$)?*

Problem 6. *Is there an online algorithm for* Unit Clustering *whose competitive ratio is subexponential in d?*

Problem 7. *Are there online algorithms for* Unit Clustering *that do not fit into the paradigm of "lifting" the one-dimensional algorithm?*

References

[1] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor, The online set cover problem, *SIAM J. Comput.* **39(2)** (2009), 361–370.

[2] Yossi Azar, Niv Buchbinder, T.-H. Hubert Chan, Shahar Chen, Ilan Reuven Cohen, Anupam Gupta, Zhiyi Huang, Ning Kang, Viswanath Nagarajan, Joseph Naor, and Debmalya Panigrahi, Online algorithms for covering and packing problems with convex objectives, in *Proc. 57th IEEE Symposium on Foundations of Computer Science (FOCS)*, IEEE, 2016, pp. 148–157.

[3] Yossi Azar, Umang Bhaskar, Lisa Fleischer, and Debmalya Panigrahi, Online mixed packing and covering, in *Proc. 24th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, SIAM, 2013, pp. 85–100.

[4] Yossi Azar, Ilan Reuven Cohen, and Alan Roytman, Online lower bounds via duality, in *Proc. 28th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, SIAM, 2017, pp. 1038–1050.

[5] Allan Borodin and Ran El-Yaniv, *Online Computation and Competitive Analysis*, Cambridge University Press, Cambridge, 1998.

[6] Niv Buchbinder and Joseph Naor, Online primal-dual algorithms for covering and packing, *Math. Oper. Res.* **34(2)** (2009), 270–286.

[7] Timothy M. Chan and Hamid Zarrabi-Zadeh, A randomized algorithm for online unit clustering, *Theory Comput. Syst.* **45(3)** (2009), 486–496. A preliminary version in *Proc. 5th Workshop on Approximation and Online Algorithms (WAOA)*, LNCS 4368, Springer, 2006, pp. 121-131.

[8] Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani, Incremental clustering and dynamic information retrieval, *SIAM J. Comput.* **33(6)** (2004), 1417–1440.

[9] Marek Chrobak, SIGACT news online algorithms column 13, *SIGACT News Bulletin* **39(3)** (2008), 96–121.

[10] János Csirik, Leah Epstein, Csanád Imreh, and Asaf Levin, Online clustering with variable sized clusters, *Algorithmica* **65(2)** (2013), 251–274.

[11] Gabriella Divéki and Csanád Imreh, An online 2-dimensional clustering problem with variable sized clusters, *Optimization and Engineering* **14(4)** (2013), 575–593.

[12] Gabriella Divéki and Csanád Imreh, Grid based online algorithms for clustering problems, in *Proc. 15th IEEE Int. Sympos. Comput. Intel. Infor. (CINTI)*, IEEE, 2014, pp. 159.

[13] Martin R. Ehmsen and Kim S. Larsen, Better bounds on online unit clustering, *Theoret. Comput. Sci.* **500** (2013), 1–24.

[14] Leah Epstein, Asaf Levin, and Rob van Stee, Online unit clustering: variations on a theme, *Theoret. Comput. Sci.* **407(1-3)** (2008), 85–96.

[15] Leah Epstein and Rob van Stee, On the online unit clustering problem, *ACM Trans. Algorithms* **7(1)** (2010), 1–18. A preliminary version in *Proc. 5th Workshop on Approximation and Online Algorithms (WAOA)*, LNCS 4927, Springer, 2007, pp. 193–206.

[16] Tomás Feder and Daniel H. Greene, Optimal algorithms for approximate clustering, in *Proc. 20th Annual ACM Symposium on Theory of Computing (STOC)*, 1988, pp. 434–444.

[17] Robert J. Fowler, Mike Paterson, and Steven L. Tanimoto, Optimal packing and covering in the plane are NP-complete, *Inform. Process. Lett.* **12(3)** (1981), 133–137.

[18] Teofilo F. Gonzalez, Clustering to minimize the maximum intercluster distance, *Theoret. Comput. Sci.* **38** (1985), 293–306.

[19] Anupam Gupta and Viswanath Nagarajan, Approximating sparse covering integer programs online, *Math. Oper. Res.* **39(4)** (2014), 998–1011.

[20] Dorit S. Hochbaum and Wolfgang Maass, Approximation schemes for covering and packing problems in image processing and VLSI, *J. ACM* **32(1)** (1985), 130–136.

[21] Jun Kawahara and Koji M. Kobayashi, An improved lower bound for one-dimensional online unit clustering, *Theoret. Comput. Sci.* **600** (2015), 171–173.

[22] Nimrod Megiddo and Kenneth J. Supowit, On the complexity of some common geometric location problems, *SIAM J. Comput.* **13(1)** (1984), 182–196.

[23] Vijay Vazirani, *Approximation Algorithms*, Springer Verlag, New York, 2001.

[24] David P. Williamson and David B. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, 2011.

[25] Hamid Zarrabi-Zadeh and Timothy M. Chan, An improved algorithm for online unit clustering, *Algorithmica* **54(4)** (2009), 490–500.