Computational Geometry Column 68

Adrian Dumitrescu[*]

Abstract

This column is devoted to geometric clustering and covering problems in $\mathbb{R}^d$. As exact solutions for these problems are usually out of reach (unless $d = 1$), one is forced to deal with approximations. Here we mostly consider online algorithms, as the online setting introduces additional difficulty due to uncertainty about the future. One representative problem is the following (so-called Unit Covering): given a set of n points in $\mathbb{R}^d$, cover the points by balls of unit diameter, so as to minimize the number of balls used.

Keywords: online algorithm, unit covering, unit clustering, competitive ratio, lower bound, Newton number.

1 Introduction

Covering and clustering are fundamental problems in the theory of algorithms, computational geometry, data structures for databases systems, optimization, etc. They arise in a wide range of applications, such as facility location, database systems, information retrieval, maintenance of dynamic structures, spatial data mining, robotics, wireless networks, and others. Such problems can be asked in any metric space, however here we focus on geometric versions due to their intrinsic appeal; moreover, this restriction generally allows sharper bounds, particularly for approximation algorithms and online algorithms. Throughout this column, the ambient space is $\mathbb{R}^d$ equipped with the L_2 norm or the L_∞ norm.

Consider the following four optimization problems listed below. By no means they exhaust the list of clustering problems; other variants, including online algorithms and variable sized clusters have been studied in [7, 8, 9, 13], and the list can continue.

The diameter of a cluster is defined to be the maximum inter-point distance between points in the cluster. The last two problems (below) are dual to the first two in the sense that the cluster diameter is given and the goal is to minimize the number of clusters (or balls).

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

(II) k-Clustering. Given a set of n points in $\mathbb{R}^d$ and a positive integer k, partition the points into k clusters so as to minimize the maximum cluster diameter.

(III) 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.

(IV) 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.

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

All four problems are easily solved in polynomial time for points on the line, i.e., for $d = 1$; but all problems become NP-hard already in the Euclidean plane [16, 21]. Factor 2 approximations are known for k-Center in any metric space (and so for any dimension) [15, 17]; see also [23, Ch. 2], while polynomial-time approximation schemes are known for Unit Covering for any fixed dimension [18]. However, these algorithms are notoriously inefficient and thereby impractical; see also [1] for a summary of results and different time vs. ratio trade-offs.

Unit Covering and Unit Clustering are identical in the offline setting, since one can go from clusters to balls in a straightforward way; and conversely, one can arbitrarily assign multiply covered points to unique balls. However, there exist key differences between the two problems in the *online* setting (see Sections 4 and 5 below).

The performance of an approximation algorithm is measured by comparing it to an optimal solution using the standard notion of *approximation ratio* [22, 23]. 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* [3, 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 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 $\sup_\sigma \frac{E[\mathsf{ALG}(\sigma)]}{\mathsf{OPT}(\sigma)}$. If there is no danger of confusion, we use ALG to refer to an algorithm or the cost of its solution, as needed.

2 Euclidean k-Center

Throughout this section, the ambient space is $\mathbb{R}^d$ equipped with the L_2 norm. Euclidean k-Center has been shown to be NP-hard by Gonzalez [17], who also gave a 2-approximation algorithm (for any metric space). Moreover, the author showed that computing a $(\sqrt{3} - \varepsilon)$-approximation in the plane and a $(2 - \varepsilon)$-approximation in $\mathbb{R}^3$ are NP-hard. An alternative 2-approximation algorithm was obtained by Hochbaum and Shmoys around the same time [19].

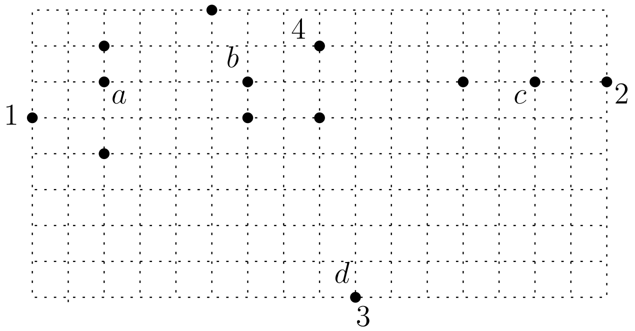

Figure 1: An planar instance of k-Center with integer points and $k = 4$. Points $1, 2, 3, 4$ are selected by the algorithm, whereas points a, b, c, d are an optimal solution; $r_{\mathsf{ALG}} = 4$, $r_{\mathsf{OPT}} = \sqrt{5}$.

Let S be the input point set. The 2-approximation algorithm by Gonzalez [17], called `Farthest-Point Clustering`, proceeds in a greedy manner. It first picks an arbitrary point, say, $p_1 \in S$ and puts it in the set C of cluster centers (while removing it from S). It then picks a second point, say, $p_2 \in S$ as far away as possible from p_1, that is, $\text{dist}(p_2, p_1) = \text{dist}(p_2, C)$ is maximized. The algorithm continues in this way, that is, by selecting p_i, $i = 3, \ldots, k$, such that $\text{dist}(p_i, C)$ is maximized, until it has selected k centers, that is, $|C| = k$. After all k centers have been chosen, the corresponding partition of S is: the jth cluster ($j = 1, \dots, k$) consists of all points that are closer to p_j than to other centers, with ties broken arbitrarily. An execution of the algorithm (with $k = 4$) is shown in Fig. 1.

Problem 1. *Can the ratio 2 approximation for k-Center in the plane be improved?*

If cluster centers can be arbitrary points (not necessarily points in the set), and the covering is still by congruent balls whose diameter is to be minimized, we have another variant, called *central clustering*; for this problem, Feder and Greene [15] proved a hardness of approximation threshold of $(1 + \sqrt{7})/2 = 1.8228\dots$; see also [2].

3 Euclidean k-Clustering and Incremental Clustering

The k-Clustering problem is also known as *pairwise clustering* [2, 15]. Throughout this section, the ambient space is $\mathbb{R}^d$ equipped with the L_2 norm. The `Farthest-Point Clustering Algorithm` of Gonzalez [17] gives a 2-approximation also for the offline version of k-Clustering. On the other hand, Feder and Greene [15] showed that it is NP-hard to approximate k-Clustering in the plane with an approximation ratio smaller than $2\cos(10^\circ) = 1.9696\dots$; see also [2].

In the online version of k-Clustering, points arrive one by one and the algorithm needs to maintain a collection of k clusters such that as each input point is presented, either it is assigned to one of the current k clusters or it starts off a new cluster while two existing clusters are merged into one. The competitive ratio of an online clustering algorithm is then defined as the supremum over all update sequences of the ratio of its maximum cluster diameter to that of the optimal k-clustering for the input points that can be obtained by an optimal offline algorithm that processes the points in the same order.

The Incremental Clustering problem, introduced by Charikar et al. [6], is defined similarly but with one key difference regarding the performance. The performance ratio of an incremental clustering algorithm is defined as the supremum over all update sequences of the ratio of its maximum cluster diameter to that of the optimal (offline) k-clustering for the input points.

The authors have avoided labeling their model as the online clustering problem or referring to the performance ratio as a competitive ratio for several reasons. One reason is the following: while in an online setting, one would compare the performance of an algorithm to that of an adversary which knows the update sequence in advance but must process the points in the *order of arrival*, the incremental algorithm has a stronger requirement in that it is compared to an adversary which does not need to respect the input ordering, i.e., the clustering computed by the incremental algorithm is compared to an optimal clustering of the final point set where no intermediate clustering need be maintained.

Charikar et al. [6] have proposed several algorithms for Incremental Clustering. Their algorithms work for any metric space, but their performance is usually better in Euclidean space. In particular, their `Doubling Algorithm` achieves a performance ratio of 8, while a randomized version achieves a performance ratio of $2e = 5.4365\dots$ (where $e = \sum_{i=0}^{\infty} 1/i!$ is the base of the natural logarithm). Also, their `Clique Algorithm` achieves a performance ratio of $4(1 + \sqrt{d/(2d+2)})$ in $\mathbb{R}^d$; this is 6 for $d = 1$, and converges to $4(1 + \sqrt{1/2}) = 6.8284\dots$ as d goes to infinity.

The authors also showed that even for points on the line, one cannot expect a performance ratio better than 2 against an adaptive deterministic adversary[1]. Consider the case $k = 2$: $n = 4$ consecutive points from among $\{0, 1, 2, 3, 4\}$ will be presented, so that $\mathsf{OPT} = 1$. Start with the

[1] An *adaptive adversary* constructs the next input point online, based on the previous actions of the algorithm.

three points $1, 2, 3$. Since there are three points and only two clusters, there is a cluster C with two consecutive points (two nonconsecutive points in a cluster immediately imply a lower bound of 2 on the maximum cluster diameter). If $1, 2 \in C$, the next point is 0, and then 0 or 3 is merged with C, which gives diameter 2; the situation is analogous if $2, 3 \in C$. The argument can be extended for any $k \geq 2$: let $n = k + 2$, where the extra points are positive multiples of 10, namely $10i$, where $i = 1, \ldots, k - 2$. Notice that again $\mathsf{OPT} = 1$ (the extra points make singleton clusters in an optimal solution). A lower bound of $2 - 2^{-\lfloor k/2 \rfloor}$ on the performance ratio of any randomized algorithm for Incremental Clustering in one dimension (against an oblivious adversary[2]) can be deduced along the same lines [6].

Problem 2. *What is the optimal competitive ratio of an online algorithm (deterministic and resp., randomized) for k-*Clustering *of points on the line (or in the plane)?*

Problem 3. *What is the optimal competitive ratio of an online algorithm (deterministic and resp., randomized) for* Incremental Clustering *of points on the line (or in the plane)?*

4 Unit Clustering and Unit Covering in the L_∞ Norm

The online version of Unit Clustering was introduced by Chan and Zarrabi-Zadeh [5]. As mentioned earlier, Unit Covering and Unit Clustering are identical in the offline setting; however, there is a key difference between the two problems in the *online* setting: as a point p arrives, the unit clustering problem only requires the algorithm to decide on the choice of the cluster containing p; the point cannot subsequently be reassigned to another cluster, but the cluster may expand (subject to the constraint).

Consider the ambient space $\mathbb{R}^d$ equipped with the L_∞ norm. A simple deterministic algorithm (`Algorithm Grid` below) that assigns points to a predefined set of unit cubes that partition $\mathbb{R}^d$ can be easily proved to be 2^d-competitive for both Unit Covering and Unit Clustering. Since in $\mathbb{R}^d$ each cluster of OPT can be split into at most 2^d grid-cell clusters created by the algorithm, its competitive ratio is at most 2^d, and this analysis is tight. See Fig. 2 for an example.

> `Algorithm Grid.` Build a uniform grid in $\mathbb{R}^d$ where cells are unit cubes of the form $\prod_{j=1}^{d} [i_j, i_j + 1)$, 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.

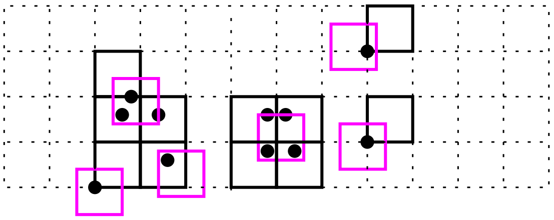

Figure 2: Example for `Algorithm Grid` in the plane; here $\mathsf{ALG} = 11$ and $\mathsf{OPT} = 6$.

[2] An *oblivious adversary* must construct the entire input sequence in advance, without having access to the actions of the algorithm.

Presently there is no online algorithm for UNIT CLUSTERING in $\mathbb{R}^d$ under the L_∞ norm with a competitive ratio $o(2^d)$. The best known algorithm has ratio $2^d \cdot \frac{5}{6}$ for every $d \geq 1$, which is only marginally better than the 2^d ratio.

Another standard approach for UNIT CLUSTERING is the deterministic algorithm `Centered`. It works for every $d \geq 1$ and it has a competitive ratio 2 for both UNIT COVERING and UNIT CLUSTERING of points on the line ($d = 1$).

> `Algorithm Centered.` For each new point p, if p is covered by an existing unit ball, do nothing; otherwise place a new unit ball centered at p.

In fact, Chan and Zarrabi-Zadeh [5] showed that no online algorithm (deterministic or randomized) for UNIT COVERING can have a competitive ratio better than 2 in one dimension ($d = 1$). However, for UNIT CLUSTERING, it is possible to get better results. Specifically, they developed randomized algorithms with competitive ratios of 15/8 and 11/6 [5, 24]. Two deterministic algorithms for this problem, with ratios of 7/4 and 5/3, were subsequently developed by Epstein and van Stee [14] and Ehmsen and Larsen [12], respectively. On the other hand, the lower bound for deterministic algorithms has evolved from 3/2 in [5] to 8/5 in [14], and then to 13/8 in [20]. Whence a small gap for the competitive ratio of deterministic algorithms for the one-dimensional case of UNIT CLUSTERING remains, namely $\frac{5}{3} - \frac{13}{8} = \frac{1}{24}$. The lower bound for randomized algorithms has evolved from 4/3 in [5] to 3/2 in [14], and so the current gap for this class of algorithms is $\frac{5}{3} - \frac{3}{2} = \frac{1}{6}$. As such, even the simplest, one-dimensional case of online UNIT CLUSTERING poses unmet challenges.

Problem 4. *What is the optimal competitive ratio of an online algorithm (deterministic and resp., randomized) for* UNIT CLUSTERING *of points on the line?*

Chan and Zarrabi-Zadeh [5] have also considered the greedy algorithm for UNIT CLUSTERING, and showed—by providing an upper bound on the ratio and a tight example—that it has competitive ratio 2 for points on the line ($d = 1$). In contrast, its competitive ratio becomes unbounded as soon as $d \geq 2$; see item (i) below.

> `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.

Recently, the following results have been obtained by Dumitrescu and Tóth [10]. First, on the negative side, some lower bounds:

(i) The competitive ratio of `Algorithm Greedy` for UNIT CLUSTERING in $\mathbb{R}^d$ under the L_∞ norm is unbounded for every $d \geq 2$. Fig. 3 illustrates the case $d = 2$ (the lower bound example extends to arbitrary $d \geq 2$). 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, $\mathsf{GREEDY} = n$, while $\mathsf{OPT} = 2$ 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.

(ii) Answering a question of Epstein and van Stee, it was shown that the competitive ratio of every online algorithm (deterministic or randomized) for UNIT CLUSTERING in $\mathbb{R}^d$ under the L_∞ norm is $\Omega(d)$ for every $d \geq 2$. Until then there was no evidence that the competitive ratio must grow with the dimension [14, Sec. 4].

(iii) The competitive ratio of every deterministic online algorithm (with an adaptive deterministic adversary) for UNIT COVERING in $\mathbb{R}^d$ under the L_∞ norm is at least $d + 1$ for every $d \geq 1$. This generalizes a result of Chan and Zarrabi-Zadeh [5] for $d = 1$ to higher dimensions.

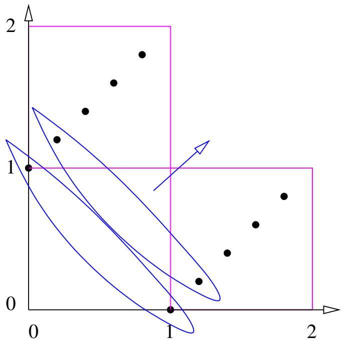

Figure 3: Example for the greedy algorithm.

And then, on the positive side, some upper bounds (and mixed results):

(iv) There exists a randomized algorithm with competitive ratio $O(d^2)$ for UNIT COVERING in $\mathbb{Z}^d$, $d \in \mathbb{N}$, under the L_∞ norm. The algorithm applies to UNIT CLUSTERING in $\mathbb{Z}^d$, $d \in \mathbb{N}$, with the same competitive ratio.

(v) The competitive ratio of `Algorithm Greedy` for UNIT CLUSTERING in $\mathbb{Z}^d$ under the L_∞ norm is at least 2^{d-1} and at most $2^{d-1} + \frac{1}{2}$ for every $d \geq 2$.

It is hard to ignore the gaps between the linear lower bounds and the exponential upper bounds in the competitive ratios for UNIT CLUSTERING. One can formulate two questions of definite interest:

Problem 5. *Is there a lower bound on the competitive ratio for* UNIT CLUSTERING *in* $\mathbb{R}^d$ *under the* L_∞ *norm that is exponential in* d*? Is there a superlinear lower bound?*

Problem 6. *Is there an online algorithm for* UNIT CLUSTERING *in* $\mathbb{R}^d$ *under the* L_∞ *norm whose competitive ratio is* $o(2^d)$*? Is there one whose competitive ratio is* $O((2-\delta)^d)$*, where* $\delta > 0$ *is an absolute constant?*

5 Euclidean Unit Covering

In relation to its sister problem, UNIT CLUSTERING, some (easy) remarks are in order. Any lower bound on the competitive ratio of an online algorithm for UNIT CLUSTERING applies to the competitive ratio of the same type of algorithm for UNIT COVERING. Conversely, if we have an online algorithm for UNIT COVERING, this is also an online algorithm for UNIT CLUSTERING with the same competitive ratio.

Throughout this section, the ambient space is $\mathbb{R}^d$ equipped with the L_2 norm. Recently, Euclidean UNIT COVERING was revisited by Dumitrescu, Ghosh, and Tóth [11], who obtained the following results:

(i) The competitive ratio of `Algorithm Centered` for online UNIT COVERING in $\mathbb{R}^d$, $d \in \mathbb{N}$, under the L_2 norm is bounded by the Newton number of the Euclidean ball in the same dimension[3]. In particular, it follows that this ratio is $O(1.321^d)$. This improves on the ratio of the previous best algorithm due to Charikar et al. [6], $O(2^d d \log d)$, by an exponential factor. The competitive ratios

[3] For a convex body $C \subset \mathbb{R}^d$, the *Newton number* (a.k.a. *kissing number*) of C is the maximum number of nonoverlapping congruent copies of C that can be arranged around C so that they each touch C [4, Sec. 2.4].

of `Algorithm Centered` are 5 in the plane and 12 in 3-space, improving the earlier ratios of 7 and 21, respectively.

(ii) From the other direction, the competitive ratio of every deterministic online algorithm (with an adaptive deterministic adversary) for Unit Covering in $\mathbb{R}^d$ under the L_2 norm is at least $d+1$ for every $d \geq 1$. This greatly improves the previous best lower bound, $\Omega(\log d / \log\log\log d)$, due to Charikar et al. [6].

(iii) The competitive ratio of any deterministic algorithm (with an adaptive deterministic adversary) for Unit Covering in $\mathbb{R}^2$ is at least 4; and in $\mathbb{R}^3$ is at least 5. The previous best lower bounds were both 3.

(iv) There exist deterministic online algorithms for Unit Covering with an optimal competitive ratio of 3 for input point sequences that are subsets of the infinite square or hexagonal lattices.

In illustration of item (iv) above for the square lattice $\mathbb{Z}^2$, refer to Fig. 4. Partition the lattice points using unit disks as shown in the figure. When a point arrives, cover it with the disk it belongs to in the partition. For the analysis, consider a disk D from an optimal cover. It is not difficult to show that D can cover points that belong to at most three disks used for partitioning the lattice, and so the algorithm has competitive ratio 3. A rather short argument shows that this ratio cannot be improved.

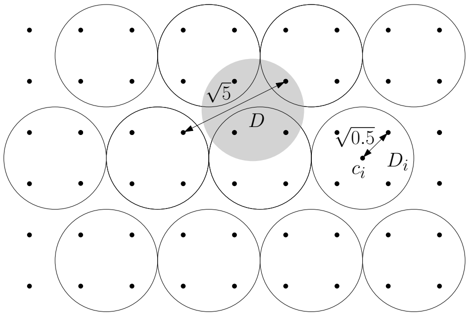

Figure 4: Covering integer lattice points; the disk D is shaded. The figure is reproduced from [11].

In conclusion, we again note the large gaps between the linear lower bounds and the exponential upper bounds in the competitive ratios, this time for Unit Covering. It is therefore natural to ask:

Problem 7. *Is there a lower bound on the competitive ratio for* Unit Covering *that is exponential in d? Is there a superlinear lower bound?*

Problem 8. *Can the online algorithm for* Unit Covering *of integer points (with ratio* 3 *in the plane) be extended to higher dimensions, i.e., for covering points in* $\mathbb{Z}^d$*? What ratio can be obtained for this variant?*

References

[1] Ahmad Biniaz, Peter Liu, Anil Maheshwari, and Michiel Smid, Approximation algorithms for the unit disk cover problem in 2D and 3D, *Comput. Geom.* **60** (2017), 8–18.

[2] Marshall Bern and David Eppstein, Approximation algorithms for geometric problems, in *Approximation Algorithms for NP-hard Problems (Dorit S. Hochbaum, editor)*, PWS Publishing Company, Boston, 1997, pp. 296–345.

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

[4] Peter Brass, William Moser, and János Pach, *Research Problems in Discrete Geometry*, Springer, New York, 2005.

[5] Timothy M. Chan and Hamid Zarrabi-Zadeh, A randomized algorithm for online unit clustering, *Theory Comput. Syst.* **45(3)** (2009), 486–496.

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

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

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

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

[10] Adrian Dumitrescu and Csaba D. Tóth, Online unit clustering in higher dimensions, *Proc. 15th International Workshop on Approximation and Online Algorithms (WAOA)*, LNCS 10787, Springer, Cham, 2017, pp. 238–252.

[11] Adrian Dumitrescu, Anirban Ghosh, and Csaba D. Tóth, Online unit covering in Euclidean space, *Proc. 12th Annual International Conference on Combinatorial Optimization and Applications* (COCOA 2018), LNCS, Springer, Cham, 2018, to appear.

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

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

[14] Leah Epstein and Rob van Stee, On the online unit clustering problem, *ACM Trans. Algorithms* **7(1)** (2010), 1–18.

[15] 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.

[16] 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.

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

[18] 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.

[19] Dorit S. Hochbaum and David B. Shmoys, A best possible heuristic for the k-center problem, *Math. Oper. Res.* **10(2)** (1985), 180–184.

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

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

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

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

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