Acta Cybernetica **21** (2013) 75–88.

Online Clustering on the Line with Square Cost Variable Sized Clusters

Gabriella Divéki[*]

Abstract

In the online clustering problems, the classification of points into sets (called clusters) is done in an online fashion. Points arrive one by one at arbitrary locations, to be assigned to clusters at the time of arrival without any information about the further points. A point can be assigned to an existing cluster, or a new cluster can be opened for it. Existing clusters cannot be merged or split. We study one-dimensional variants. The cost of a cluster is the sum of a fixed setup cost scaled to 1 and the square of the diameter of the cluster. The goal is to minimize the sum of costs of the clusters used by the algorithm. In this paper we investigate the problem on the line.

We examine two versions, both maintaining the properties that a point which was assigned to a given cluster must remain assigned to this cluster, and clusters cannot be merged. In the strict variant, the size and the exact location of the cluster must be fixed when it is initialized. In the flexible variant, the algorithm can shift the cluster or expand it, as long as it contains all points assigned to it. We consider the online and the semi-online (the input is sorted according to their coordinates from smallest to largest i.e., from left to right) versions of the above two variants.

We present the first online algorithms for the solution of the problem. We describe algorithms for the strict and the flexible variant both for the online and semi-online versions. We also give lower bounds on the possible competitive ratio in all of the cases.

Keywords: online algorithms, competitive analysis, clustering problems

1 Introduction

In clustering problems, we seek for a partitioning of n demand points into k groups, or clusters, while a given objective function, that depends on the distance between points in the same cluster, is minimized. In the online version, the demand points

[*] Subotica Tech - College of Applied Sciences, Marka Oreškovića 16, 24000 Subotica, Serbia, E-mail: diveki.gabriella@gmail.com

DOI: 10.14232/actacyb.21.1.2013.6

are presented to the clustering algorithm one by one. The online clustering algorithm maintains a set of clusters, where a cluster is identified by its name and the set of points already assigned to it. Each point must be assigned to a cluster at the time of arrival; the chosen cluster becomes fixed at this time. The clusters cannot be merged or split.

Usually, the quality of an online algorithm is measured by competitive analysis. An online algorithm for a minimization problem is C-competitive if the algorithm cost is never more than C times the optimal offline cost. (For a good introduction to competitive analysis, see [3, 11, 17].) In the case of clustering problems, the costs are based on the number of clusters and their properties, and they depend on the exact specification of the problem.

In this paper we consider the 1-dimensional variant of the 2-dimensional online clustering with variable sized clusters problem which is presented in [22]. In our model points of the 1-dimensional Euclidean space arrive one by one. After the arrival of a point we have to assign it to an existing cluster or to define a new cluster for it without any information about the further points. The clusters are intervals, the cost of each cluster is the sum of the constant setup cost scaled to 1 and the square of the length of the interval. The goal is to minimize the total cost of the clusters.

We consider two variants, both having property that a point assigned to a given cluster must remain in this cluster, and clusters cannot be merged. In the strict variant, the size and the location of the cluster must be fixed when it is initialized. In the flexible variant, the algorithm can shift the cluster or expand it, as long as it contains all the points assigned to it.

In [6] the one-dimensional variant of our problem is examined (with linear cost), where there is no restriction on the length of a cluster, and the cost of a cluster is the sum of a fixed setup cost and its diameter. Both the strict and the flexible model have been investigated and an intermediate model, where the diameter is fixed in advance but the exact location can be modified is also studied. In [6], tight bounds are given on the competitive ratio of any online algorithm belonging to any of these variants. Tight bounds are given of $1+\sqrt{2} \approx 2.414$ on the competitive ratio for the online problem in the strict model, and tight bounds of 2 in the semi-online version. In the intermediate model, the results of the previous model were extended and it is shown that the same bounds are tight for it as well. Using the flexible model, the best competitive ratio dropped to $\Phi = \frac{1+\sqrt{5}}{2} \approx 1.618$. The semi-online version of this model is solved optimally using a trivial algorithm which is discussed as well in [6].

Several results are known on online clustering with fixed unit sized clusters. A study of online partitioning of points into clusters was presented by Charikar et al. [5]. The problem is called online unit covering. A set of n points needs to be covered by balls of unit radius, and the goal is to minimize the number of balls used. The authors designed an algorithm with 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 deterministic online algorithms in d dimensions. This problem is strictly online: the points arrive one by one, each point has to be assigned to a ball upon arrival, and if it is assigned to a new ball, the exact location of this ball is fixed at this time. The tight bounds on the competitive ratio for $d = 1$ and $d = 2$ are 2 and 4, respectively.

Chan and Zarrabi-Zadeh [4] introduced the unit clustering problem. Here the input and goals are identical to those of unit covering, but the model of online computation is different. This is an online problem as well, but it is more flexible in the sense that the online algorithm is not required to fix the exact position of each ball at the first time the ball is "used". The set of points which is assigned to a ball (cluster) can always be covered by that ball and the ball can be shifted if necessary. The goal is still to minimize the total number of balls used. Unit covering and unit clustering are the same problem when observing in an offline fashion, and the problem is solvable in polynomial time for $d = 1$. In the online model an algorithm for the unit clustering problem has more flexibility because of the optional shifting of a cluster. In [4], the authors showed that standard approaches lead to algorithms of competitive ratio 2 (some of which are valid for unit covering). The lower bound of 2 for unit covering in one dimension is valid even for randomized algorithms. A non-trivial randomized algorithm was presented: a $\frac{15}{8}$-competitive algorithm; also in [21] an $\frac{11}{6}$-competitive randomized algorithm. In [10] an improved deterministic algorithm was given (with competitive ratio $\frac{7}{4}$) and in [8] an algorithm of competitive ratio $\frac{5}{3}$. Currently the best known lower bound is $\frac{8}{5}$ (see [10]).

In [4, 8, 10, 21] the two-dimensional problem is considered using the ℓ_∞ norm instead of the ℓ_2 norm. Thus, "balls" are squares or cubes. The one-dimensional algorithms are used as building blocks in most results in the mentioned papers. This problem has a higher competitive ratio than the one-dimensional case (the best known lower bound is $\frac{13}{6}$ - see [8]). Other variants of the one-dimensional online unit clustering problem were studied in [9].

Our problem is also related to online facility location [7, 12, 13, 14, 15, 19], where the input is a sequence of points and the algorithm has to partition them into clusters and it has to assign a facility to each cluster. On the other hand in facility location the cost of the cluster differs: it is the sum a fixed setup cost and the service cost which is the total distance of the points from a facility assigned to the cluster. In some of the online facility location models it is allowed to merge clusters or to re-assign points.

Our results: We present the first online algorithms for the solution of the problem. We present algorithms for the strict and the flexible variant both for the online and semi-online versions. In this paper, when we refer to a semi-online algorithm we mean an online algorithm for restricted set of inputs in which the points arrive one by one when they are sorted according to their coordinates from smallest to largest (i.e., from left to right).We also give lower bounds on the possible competitive ratio in all of the cases.

We analyze algorithms and give their competitive ratio. We prove that the $GRID_a$ algorithm is 3-competitive in the strict model if we use appropriate size for the cells for the grid. We present the algorithm $SOSM_a$ for the semi-online strict model and prove that it has competitive ratio 2 if the size of the cells are within appropriate bounds. We also give lower bounds in the strict model for both variants: 2.2208 for the online and 1.6481 for the semi-online variant.

We extend the algorithm to the flexible model and we prove that it is 2-competitive if we use appropriate size for the cells. In this model we show that no online algorithm can have smaller competitive ratio than 1.2993. Also, we give the lower bound for the semi-online flexible model: 1.1991144.

In the rest of the paper for an algorithm A and input I we use $A(I)$ to denote the cost of A on input I.

2 The offline problem

As far as we know the offline clustering with this objective function have not been studied yet. Many papers are published on the offline version where the number of clusters is a given constant k. Usually the cost is the sum of the diameters (see [16] and its references for details) but there are also some results on the models where it depends on the powers of the diameters (see [2]), and even for general cost functions (see [18]). All of these problems are NP-hard. If the number of clusters is not fixed and the cost depends on the diameters then the problem is polynomially solvable for trees see [20] and it has not been studied for more general metric spaces yet.

Lemma 1. *The offline problem can be solved optimally by the dynamic programming algorithm DP using the algorithm for the variation of the k-median problem.*

This is an interesting transition: our offline clustering problem on the line with linear objective function can be solved with a simple greedy algorithm with $O(n \cdot logn)$ time complexity (see [6]), the problem on the line with squared cost can be solved by a standard $O(n^3)$ time dynamic programming algorithm. On the other hand the 2-dimensional case seems to be much harder, we conjecture that it is NP-hard.

The input is n request points $(x_1, ..., x_n)$. The dynamic programming algorithm is shown in Algorithm 1.

Algorithm 1 Algorithm DP

- The request points are sorted by their coordinates in ascending order.
- Define the subproblem $F(i, r)$ $(i \geq r)$: the first i request points are divided into r clusters. Then the optimal cost of the clustering problem is $min_r(F(n, r) + r)$.
- The values of $F(i, r)$ can be calculated by the following recursions.

  $F(i, 1) = (x_i - x_1)^2$

  $F(i, r) = min_{j=r}^{i}\{F(j-1, r-1) + (x_i - x_j)^2\}$

The dynamic programming algorithm correctly calculates because if the last cluster is $[x_j, x_i]$ then we have to assign the first $j-1$ request points into $r-1$ clusters optimally.

Based on these steps of the dynamic programming algorithm a 2-dimensional array can be filled sorted by the second dimension r in ascending order. Then one can get the optimal solution from this table. As the algorithm DP fills an $n \times n$ table and an element of the table can be computed in $O(n)$ steps, therefore the time complexity of algorithm DP is $O(n^3)$.

3 The strict model

3.1 The online problem

The $GRID$ algorithm which uses a grid in the 1-dimensional space is defined in [9] for the problem of unit covering with rejection. In [22] the $GRID_a$ algorithm was investigated in 2 dimensions for the strict model; we consider its special 1-dimensional case and also the analysis is similar.

Algorithm $GRID_a$ works as follows. Upon arrival of the first point in the interval $I_k = (ka, (k+1)a]$ for every integer $-\infty < k < \infty$, a new cluster is opened in the interval $[ka, (k+1)a]$ and all future points in this interval are assigned to this cluster. The competitive ratio of $GRID_a$ is determined by the following theorem.

Theorem 1. *The competitive ratio of algorithm* $GRID_a$ *is*

$$\max\{F(\lfloor -2 + \sqrt{4 + \frac{1}{a^2}} \rfloor), F(\lceil -2 + \sqrt{4 + \frac{1}{a^2}} \rceil), 2 + 2a^2\}$$

where $F(k) = \frac{(k+2)(1+a^2)}{1+k^2a^2}$, $k \geq 1$.

Proof. Consider an arbitrary sequence and an optimal solution for it, denoted by OPT. We investigate the clusters of OPT separately. Consider an arbitrary cluster. Let r denote the length of this cluster.

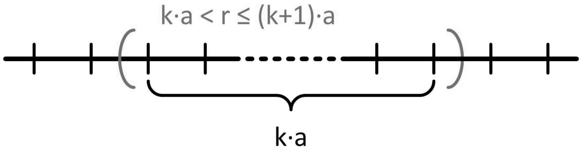

Figure 1: The optimal cluster intersects at most $k+2$ clusters from the grid

Suppose first that $k \cdot a < r \leq (k+1) \cdot a$ for an integer $k \geq 1$. Then this optimal cluster intersects at most $k+2$ clusters from the grid (see Figure 1). Therefore, if we consider only the requests of this optimal cluster then $GRID_a$ has at most

$(1+a^2)(k+2)$ cost. Thus the competitive ratio on this subsequence is at most $\frac{(1+a^2)(k+2)}{r^2+1} < \frac{(1+a^2)(k+2)}{k^2a^2+1} = F(k)$. The derivative of this function is

$$F'(k) = \frac{(1+a^2)\cdot(1-4ka^2-k^2a^2)}{(1+k^2a^2)^2}.$$

$F'(k)$ is 0 at $k^* = -2+\sqrt{4+\frac{1}{a^2}}$. The second derivative of $F(k)$ is

$$F''(k) = \frac{2\cdot(1+a^2)\cdot a^2\cdot(k^3a^2-3k+6k^2a^2-2)}{(k^2a^2+1)^3}$$

while $F''(k^*) < 0$ for every a. Therefore $F'(k)$ is positive before k^*, and it is negative after k^*. This yields that $F(k)$ has maximum at k^*. We have to consider the positive integers, so the maximum is attained either at $k = \lfloor -2+\sqrt{4+\frac{1}{a^2}}\rfloor$ or at $k = \lceil -2+\sqrt{4+\frac{1}{a^2}}\rceil$.

Now suppose that $r \le a$. Then the cluster intersects at most 2 clusters from the grid. Therefore, considering the requests of this cluster $GRID_a$ has at most $2\cdot(1+a^2)$ cost. Thus the competitive ratio on this subsequence is at most $(2+2a^2)/(1+r^2) \le 2+2a^2$.

Now we prove that the analysis is tight. Consider an arbitrary a and let ε be a small positive number. If the request sequence consists of the points $-\varepsilon$ and ε then the optimal solution uses only one cluster and has cost $1+(2\varepsilon)^2$ while the algorithm uses two clusters and has cost $2(1+a^2)$. Since ε can be arbitrarily small we obtain that the competitive ratio is not smaller than $2+2a^2$.

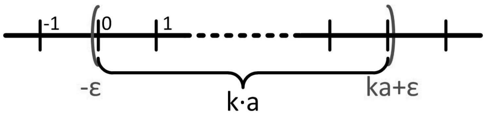

Figure 2: The interval with endpoints $-\varepsilon$ and $ka+\varepsilon$

Now suppose that all the points are requested in the interval with endpoints $-\varepsilon$ and $ka+\varepsilon$ (see Figure 2). If we use only one cluster then the cost is $1+(ka+2\varepsilon)^2$. $GRID_a$ uses $k+2$ cells, thus its cost is $(1+a^2)(k+2)$. This yields that a lower bound on the ratio of the cost of $GRID_a$ and the optimal cost tends to $F(k)$ as ε tends to 0. Therefore we obtained that the competitive ratio of $GRID_a$ is not smaller than $F(k)$ for any positive k, and this shows the tightness of our analysis. □

Corollary 1. *The smallest competitive ratio of $GRID_a$ is obtained if $\frac{1}{2\sqrt{2}} \le a \le \frac{1}{\sqrt{2}}$, then the competitive ratio of the algorithm is* 3.

Proof. First observe that $F(k) = 3$ for $k = 1$ for each value of the parameter a. If $\frac{1}{2\sqrt{2}} \le a \le \frac{1}{\sqrt{2}}$ then $F(k) \le 3$ for all integers $k > 1$ and also $2(1+a^2) \le 3$, thus the algorithm is 3-competitive. If $a > \frac{1}{\sqrt{2}}$ then $2(1+a^2) > 3$. If $a < \frac{1}{2\sqrt{2}}$ then

$F(2) > 3$, therefore if $a \notin [\frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{2}}]$ then the competitive ratio of $GRID_a$ is larger than 3.

□

Remark 1. The competitive ratio of any online algorithm for the strict model is at least 2.2208. The proof of this lower bound in [22] (where the 2-dimensional variant of this problem is studied) uses one dimension so it also applies in our case.

3.2 The semi-online strict model

In the semi-online model the points arrive in ascending order. A possible algorithm to solve this problem is $SOSM_a$.

Algorithm 2 Algorithm $SOSM_a$

```
1. Let $p$ be the new point.
2. If the algorithm has a cluster which contains $p$, then assign $p$ to that cluster.
3. Else, open a new cluster $[p,\, p+a]$ and assign $p$ to the new cluster.
```

Theorem 2. *The competitive ratio of algorithm* $SOSM_a$ *is*

$$\max\{F(\lfloor -1+\sqrt{1+\frac{1}{a^2}}\rfloor), F(\lceil -1+\sqrt{1+\frac{1}{a^2}}\rceil), 1+a^2\}$$

where $F(k) = \frac{(k+1)(1+a^2)}{1+k^2a^2}$*,* $k \geq 1$*.*

Proof. Consider and arbitrary sequence and an optimal solution for it, denoted by OPT. We investigate the clusters of OPT separately. Consider an arbitrary cluster. Let r denote the length of this cluster.

Suppose first that $k \cdot a < r \leq (k+1) \cdot a$ for an integer $k \geq 1$. Then this optimal cluster intersects at most $k+2$ clusters (see Figure 1). We have to consider only the clusters which left endpoint is bigger than or equal to the left endpoint of this optimal cluster. The eventual other cluster which is hanging into this optimal cluster is considered with the optimal cluster on the left. Therefore, if we consider only these clusters then $SOSM_a$ has at most $(1+a^2)(k+1)$ cost. Thus the competitive ratio on this subsequence is at most $\frac{(1+a^2)(k+1)}{r^2+1} < \frac{(1+a^2)(k+1)}{k^2a^2+1} = F(k)$. The derivative of this function is

$$F'(k) = \frac{(1+a^2)\cdot(1-2ka^2-k^2a^2)}{(1+k^2a^2)^2}.$$

$F'(k)$ is 0 at $k^* = -1+\sqrt{1+\frac{1}{a^2}}$. The second derivative of $F(k)$ is

$$F''(k) = \frac{2 \cdot (1+a^2) \cdot a^2 \cdot (k^3a^2 - 3k + 3k^2a^2 - 1)}{(k^2a^2+1)^3}$$

while $F''(k^*) < 0$ for every a (the calculations have been made in MATLAB). This yields that $F(k)$ has maximum at k^*. We have to consider the positive integers, so the maximum is attained at $k = \lfloor -1 + \sqrt{1+\frac{1}{a^2}} \rfloor$ or at $k = \lceil -1 + \sqrt{1+\frac{1}{a^2}} \rceil$.

Now suppose that $r \le a$. Then the cluster intersects at most 2 $SOSM_a$ clusters, but we have to consider only the cluster which left endpoint is bigger than or equal to the left endpoint of this optimal cluster. Therefore, considering the requests of this cluster $SOSM_a$ has at most $1+a^2$ cost. Thus the competitive ratio on this subsequence is at most $(1+a^2)/(1+r^2) \le 1+a^2$.

Now we prove that the analysis is tight. Consider an arbitrary a and let ε be a small positive number. If the request sequence consists of one point then the optimal solution has cost 1 and the algorithm uses one cluster and has cost $1+a^2$. We obtain that the competitive ratio is not smaller than $1+a^2$.

Now suppose that all the points are requested in the interval with endpoints $-\varepsilon$ and $ka+\varepsilon$. If we use only one cluster then the cost is $1+(ka+2\varepsilon)^2$. $SOSM_a$ uses $k+1$ cells, thus its cost is $(1+a^2)(k+1)$. This yields that a lower bound on the ratio of the cost of $SOSM_a$ and the optimal cost tends to $F(k)$ as ε tends to 0. Therefore we obtained that the competitive ratio of $SOSM_a$ is not smaller than $F(k)$ for any positive k, and this shows the tightness of our analysis. □

Corollary 2. *The smallest competitive ratio of $SOSM_a$ is obtained if $\frac{1}{\sqrt{5}} \le a \le 1$, then the competitive ratio of the algorithm is* 2.

Proof: First observe that $F(k) = 2$ for $k = 1$ for each value of the parameter a. If $\frac{1}{\sqrt{5}} \le a \le 1$ then $F(k) \le 2$ for all integers $k > 1$ and also $1+a^2 \le 2$, thus the algorithm is 2-competitive. If $a > 1$ then $1+a^2 > 2$. If $a < \frac{1}{\sqrt{5}}$ then $F(2) > 2$, Therefore if $a \notin [\frac{1}{\sqrt{5}}, 1]$ then the competitive ratio of $SOSM_a$ is larger than 2.

Theorem 3. *The competitive ratio of any semi-online algorithm for the strict model is at least 1.6481.*

Proof. Let the first request point be $p_1 = 0$ and let a_1 be the length of the cluster which is opened by the algorithm. Let the second request point be $p_2 = a_1 + \varepsilon$. Then the online algorithm opens a new cluster with length $a_2 \ge 0$.

- If $a_1 + \varepsilon \le 0.83035$ then
  - if $a_2 \le 0.30817$ then another request point arrives: $p_3 = a_1 + a_2 + 2\varepsilon$.

$$\frac{A(I)}{OPT(I)} \ge \frac{3+a_1^2+a_2^2+a_3^2}{1+(a_1+a_2+2\varepsilon)^2} \to \frac{3+a_1^2+a_2^2+a_3^2}{1+(a_1+a_2)^2} \ge \frac{3+a_1^2+a_2^2}{1+(a_1+a_2)^2}$$

$$\geq \frac{3+0.83035^2+0.30817^2}{1+(0.83035+0.30817)^2} > 1.6481$$

The inequality is valid because the ratio is decreasing both in a_1 and a_2; $\varepsilon \to 0$, $a_1 \leq 0.83035$, $0 \leq a_2 \leq 0.30817$ and $a_3 \geq 0$.

– if $a_2 > 0.30817$ then the request sequence stops and we have:

$$\frac{A(I)}{OPT(I)} \geq \frac{2+a_1^2+a_2^2}{1+(a_1+\varepsilon)^2} \to \frac{2+a_1^2+a_2^2}{1+a_1^2}$$

$$\geq \frac{2+0.83035^2+0.30817^2}{1+0.83035^2} > 1.6481$$

The inequality is valid because the ratio is decreasing both in a_1 and a_2; $\varepsilon \to 0$, $a_1 \leq 0.83035$ and $a_2 > 0.30817$.

- If $a_1 + \varepsilon > 0.83035$ then

  – if $a_2 \leq 0.77894$ then another request point arrives: $p_3 = a_1 + a_2 + 2\varepsilon$. The optimal solution may use 2 clusters ($[p_1, p_1]$ and $[p_2, p_3]$) and the estimation follows:

$$\frac{A(I)}{OPT(I)} \geq \frac{3+a_1^2+a_2^2+a_3^2}{2+(a_2+\varepsilon)^2} \to \frac{3+a_1^2+a_2^2+a_3^2}{2+a_2^2}$$

$$\geq \frac{3+a_1^2+a_2^2}{2+a_2^2} \geq \frac{3+0.83035^2+0.77894^2}{2+0.77894^2} > 1.6481$$

The inequality is valid because the ratio is decreasing both in a_1 and a_2; $\varepsilon \to 0$, $a_1 \leq 0.83035$, $0 \leq a_2 \leq 0.77894$ and $a_3 \geq 0$.

  – if $a_2 > 0.77894$ then the request sequence stops. The optimal solution may use 2 clusters ($[p_1, p_1]$ and $[p_2, p_2]$) and we have:

$$\frac{A(I)}{OPT(I)} \geq \frac{2+a_1^2+a_2^2}{2} \geq \frac{2+0.83035^2+0.77894^2}{2} > 1.6481$$

The inequality is valid because the ratio is decreasing both in a_1 and a_2; $a_1 \leq 0.83035$ and $a_2 > 0.77894$.

□

4 The flexible model

4.1 The online problem

In the case of 1 dimension with the linear cost the ECC (extend closed cluster) algorithm (see [6]) has competitive ratio $\frac{1+\sqrt{5}}{2} \approx 1.618$. In [22] it is extended to 2-dimensions and it is shown that the extended algorithm ECC is not constant competitive. If we consider the 1-dimension variant with the square cost we obtain that it is neither constant competitive. The proof of that claim is the same as in [22]. As the proof shows an algorithm should limit the size of the clusters. The following extension of the $GRID_a$ algorithm satisfies this property.

Algorithm 3 Algorithm $FGRID_a$

```
1. Let $p$ be the new point.
2. If the algorithm has a cluster whose current associated interval contains $p$,
   then assign $p$ to that cluster, and do not modify the associated interval of the
   cluster.
3. Else, consider the cell from the grid which contains $p$.
   a) If this cell does not have a cluster, then open a new cluster and assign
      $p$ to the new cluster. In this case the current cluster consists of a single
      point $p$.
   b) Otherwise, extend the cluster contained in the interval to cover $p$.
```

Theorem 4. *The competitive ratio of algorithm $FGRID_a$ is 2 if $\frac{1}{\sqrt{5}} \leq a \leq 1$.*

Proof. Consider an arbitrary sequence and an optimal solution for it, denoted by OPT. We investigate the clusters of OPT separately. Consider an arbitrary cluster. Let r denote the length of the side of this cluster.

Suppose that $k \cdot a < r \leq (k+1) \cdot a$ for an integer $k \geq 1$. Then the optimal cluster intersects at most $k+2$ cells of the grid. The cells which are not at endpoints of the optimal cluster might be completely covered by $FGRID_a$. Consider now the end cells, denote by A_1 and A_2 the square costs of the intervals covered by the optimal cluster in these end cells and let $A = A_1 + A_2$. At these end cells of the optimal cluster we have two possibilities. If the cell has no intersection with other optimal clusters, then OPT and $FGRID_a$ cover the same parts of the cell. If the cell intersects at least one other optimal cluster, then it might be completely covered by $FGRID_a$ but then its online cost is divided between at least two optimal clusters and we have to consider only the half of this cost here which is $\frac{1}{2} \cdot (1+a^2)$. Therefore we obtained that assigning a total cost $2 \cdot \frac{1}{2}(1+a^2)+A$ from the online cost to these end cells we cover the full online cost by the costs assigned to the optimal clusters. Thus we assigned at most $(1+a^2) \cdot k + 2 \cdot \frac{1}{2}(1+a^2) + A = (k+1)(1+a^2) + A$

cost from $FGRID_a(I)$ to this optimal cluster. The cost of the optimal cluster is at least $1 + k^2a^2 + A$. If we consider the ratio of these costs we obtain that it is

$$\frac{(k+1)(1+a^2)+A}{k^2a^2+1+A} \le \frac{(k+1)(1+a^2)}{k^2a^2+1}.$$

If $k = 1$ then this ratio is 2 for each a. For $k > 1$ and $a > 1$ this ratio is smaller than 2.

Now suppose that $r < a$. Then the optimal cluster intersects at most 2 cells from the grid. At these cells again we have two possibilities. If the cell has no intersection with other optimal clusters, then OPT and $FGRID_a$ cover the same parts of the cell. If the cell intersects at least one other optimal cluster, then it might be completely covered by $FGRID_a$ but then its cost is divided between at least two optimal clusters and we have to consider only the half of this cost here which is $\frac{1}{2}(1 + a^2)$. Therefore we obtained that assigning a total cost $2 \cdot \frac{1}{2}(1 + a^2) + r^2 = 1 + a^2 + r^2$ from $FGRID_a(I)$ to this cluster we cover $FGRID_a(I)$ by the costs assigned to the optimal clusters. The cost of the optimal cluster is at least $1 + r^2$. If we consider the ratio of these costs we obtain that it is

$$\frac{1+a^2+r^2}{1+r^2} \le 1 + a^2.$$

On the other hand $1 + a^2 \le 2$ if $a \le 1$.

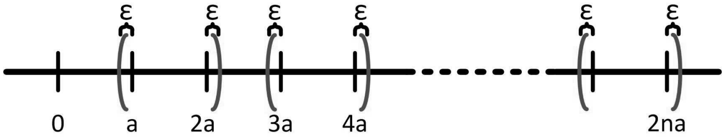

Figure 3: The competitive ratio of $FGRID_a$: the n intervals with endpoints $(2i - 1)a - \varepsilon$ and $2ia + \varepsilon$, $i = 1, ..., n$

Now we prove the tightness. Fix an a and let $\varepsilon > 0$ be a small positive number. Consider the input I_n where all the points in the n intervals with endpoints $(2i - 1)a - \varepsilon$ and $2ia + \varepsilon$, $i = 1, ..., n$ are requested (see Figure 3).

Then a solution can use each such interval as a cluster therefore the cost of OPT is at most $n \cdot (1 + (a + \varepsilon)^2)$. Now investigate the behavior of $FGRID_a$. It covers completely the grid cells with endpoints ia and $(i+1)a$, $i = 1, ..., 2n - 1$ and with ε^2 cost the 2 end cells.

Therefore we obtained that $FGRID_a(I_n) \ge (1 + a^2)(2n - 1) + 2(1 + \varepsilon^2)$. The ratio $FGRID_a(I_n)/OPT(I_n)$ tends to 2 as ε tends to 0 and n tends to ∞, thus we proved that the algorithm is not better than 2-competitive. □

Theorem 5. *The competitive ratio of any online algorithm for the flexible model is at least 1.2993.*

Proof. Suppose that there exists an online algorithm with smaller competitive ratio than 1.2993, denote it by A. Consider the following input sequence. The first two points are $p_1 = 0$ and $p_2 = 0.878$. Now distinguish the following cases.

- If A assigns these points to different clusters then three more points arrive: $p_3 = 0.329$, $p_4 = 0.439$ and $p_5 = 0.549$. The optimal algorithm uses only one cluster and its cost is $1 + 0.878^2 = 1.770884$. The cost of the online algorithm is at least $2 + 0.329^2 + 0.439^2 = 2.300962$ (it is the case when A extends both existing clusters "inward": one to the nearest new point and the other to the second new point – see Figure 4), thus the ratio is at least $2.300962/1.770884 > 1.2993$, which is a contradiction.
- If A assigns the points to one cluster then two more points arrive $p_3 = -0.355$ and $p_4 = 1.233$. Then the optimal algorithm uses two clusters, both of them have size 0.355, thus the optimal cost is $2 \cdot (1 + 0.355^2) = 2.25205$. The cost of A is at least $2 + (0.878 + 0.355)^2 = 3.520289$, thus the ratio is at least $3.520289/2.25205 \approx 1.563149$, which is a contradiction.

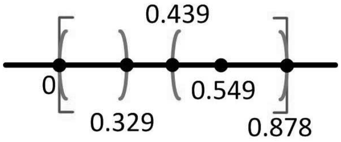

Figure 4: Lower bound in the flexible model: the cost of the online and offline algorithms

We obtained contradiction in both cases, thus we proved the theorem.

□

4.2 The semi-online flexible model

Theorem 6. *The competitive ratio of any semi-online algorithm for the flexible model is at least 1.1991144.*

Proof. Suppose that there exists a semi-online algorithm with smaller competitive ratio than 1.1991144, denote it by A. Let the first two request points be $p_1 = 0$ and $p_2 = 0.81725$.

- If the semi-online algorithm A puts them into one cluster then another request point arrives $p_3 = 1.29147$. The cost of the semi-online algorithm is at least $1 + 1.29147^2 = 2.6678947609$ (it is the case when the algorithm extends the existing cluster to the new point) while the optimal offline algorithm uses two clusters with p_1 in the first and p_2 and p_3 in the second cluster. Its cost is $2 + 0.47422^2 = 2.2248846084$, so we obtain:

$$\frac{A(I)}{OPT(I)} \geq \frac{2.6678947609}{2.2248846084} \approx 1.199116 > 1.1991144$$

.

- If the semi-online algorithm A puts p_1 and p_2 into two clusters, the sequence stops. The offline algorithm puts them into one cluster, so the competitive ratio is:

$$\frac{A(I)}{OPT(I)} \geq \frac{2}{1+0.81725^2} \approx 1.199114409 > 1.1991144$$

.

In both cases we have contradiction so the claim of the theorem holds.

□

Remark 2. We note that a similar modification to the algorithm $SOSM_a$ like in the online case (modification of $GRID_a$ that led to the algorithm $FGRID_a$) does not result in a better competitive ratio than 2 (like in the online case with algorithm $FGRID_a$).

References

[1] Anagnostopoulos, A., Bent, R., Upfal, E., and Van Hentenryck, P. A simple and deterministic competitive algorithm for online facility location. *Information and Computation*, **194(2)**, 175–202, 2004.

[2] Bilo, V., Caragiannis, I., Kaklamanis, C., and Kanellopoulos, P. Geometric Clustering to Minimize the Sum of Cluster Sizes. *ESA '05, LNCS 3669*, pp. 460–471, 2005.

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

[4] Chan, T. M. and Zarrabi-Zadeh, H. A randomized algorithm for onine unit clustering. *Theory of Computing Systems*, **45(3)**, 486–496, 2009.

[5] Charikar, M., Chekuri, C., Feder, T., and Motwani, R. Incremental clustering and dynamic information retrieval. *SIAM Journal on Computing*, **33(6)**, 1417–1440, 2004.

[6] Csirik, J., Epstein, L., Imreh, Cs., and Levin, A. Online Clustering with Variable Sized Clusters. *Algorithmica*, DOI: 10.1007/s00453-011-9586-2, 2011

[7] Divéki, G. and Imreh, Cs. Online facility location with facility movements. *Central European Journal on Operations Research*, **19(2)**, 191–200, 2011.

[8] Ehmsen, M. R. and Larsen, K. S. Better bounds on online unit clustering. *In Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT2010)*, pages 371–382, 2010.

[9] Epstein, L., Levin, A., and van Stee, R. Online unit clustering: Variations on a theme. *Theoretical Computer Science*, **407(1-3)**, 85–96, 2008.

[10] Epstein, L. and van Stee, R. On the online unit clustering problem. *ACM Transactions on Algorithms*, **7(1)**, Article 7 (18 pages), 2010.

[11] Fiat, A., Woeginger, G. J., editors. *Online algorithms: The State of the Art, LNCS 1442.* Springer-Verlag Berlin, 1998.

[12] Fotakis, D. Incremental Algorithms for Facility Location and k-Median. *Theoretical Computer Science*, **361**, 275–313, 2006.

[13] Fotakis, D. A Primal-Dual Algorithm for Online Non-Uniform Facility Location. *Journal of Discrete Algorithms*, **5**, 141–148, 2006.

[14] Fotakis, D. Memoryless Facility Location in One Pass. *Proceedings of STACS '06, LNCS 3884*, 608–620, 2006.

[15] Fotakis, D. On the Competitive Ratio for Online Facility Location *Algorithmica*, **50(1)**, 1–57, 2008.

[16] Gibson, M., Kanade, G., Krohn, E., Pirwani, I. A., and Varadarajan, K. On Metric Clustering to Minimize the Sum of Radii. *Algorithmica*, **57**, 484–498, 2010.

[17] Imreh, Cs. Competitive analysis. In *Algorithms of Informatics Volume 1*, ed. Antal Iványi, mondAt, Budapest 2007, 395–428.

[18] Levin, A. A generalized minimum cost k-clustering. *ACMTrans. Algorithms*, **5(4)**, Article 36, 2009.

[19] Meyerson, A. Online facility location. *In Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS2001)*, pages 426–431, 2001.

[20] Shah, R. and Farach-Colton, M. Undiscretized dynamic programming: faster algorithms for facility location and related problems on trees. *In Proc. of the* 13^{th} *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2002)*, pp. 108–115, 2002.

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

[22] Divéki, G. and Imreh, Cs. An online 2-dimensional clustering problem with variable sized clusters *Submitted to OPTE, 2011.*