University of Szeged
Department of Computer Algorithms and Artificial Intelligence

Online algorithms for clustering problems

Ph.D. Thesis

Gabriella Divéki

Supervisor:
Dr. Csanád Imreh

University of Szeged
PhD School in Computer Sciences

Szeged, 2014

Acknowledgments

I would like to thank all of those who helped me, one way or another, to get where I am now in my scientific and professional career.

First of all, I would like to express my gratitude to my supervisor, Dr. Csanád Imreh for his professional and personal support, who introduced me to the field of online algorithms and competitive analysis. He gave me invaluable instructions and advices from the beginning of my Ph.D. studies. Without him this thesis would not have been possible.

I also received a lot of assistance from many professors and colleagues at the Institute of Informatics of the University of Szeged. I would also like to thank the anonymous reviewers of my papers for their useful comments and suggestions.

I would like to express my gratitude to my mother, mother-in-law and father-in-law for caring for my children while I was writing this thesis, and thanks to my children Feri and Hajni for letting me "work" instead of spending time with them. Last, but not least I would like to thank my husband, Szabolcs, for his support and for just being there for me.

Contents

1 Introduction

1.1 Online problems

In the practice often arise such optimization problems where the input – the numbers which define the problem – is known piece by piece, it is also unknown if there are any more input. These problems are called *online problems* and the so called *online algorithms* which solve them can process the input in a serial fashion without having the entire input available from the start and they are not able to "see the future". In contrast, the optimal offline algorithm can view the sequence of requests in advance. An online algorithm is forced to make decisions that may later turn out not to be optimal.

The first results in the field of online algorithms originate from the 1970's, then from the beginning of the 1990's more and more researchers pay attention to this field and commence to study its problems. Numerous subfields are formed and nowadays on most of the important conferences dealing with algorithms and operational research, a lot of new results are presented which means that the field is very popular.

The study of online algorithms has focused on the quality of decision-making that is possible in this setting.

Two basic methods are used to measure the effectiveness of online algorithms. One of the possibilities is analyzing an average case. We have to assume some kind of probability distribution on the possible input and for this distribution the expected value of the algorithm is examined. The main disadvantage of this method is that the distribution of the input is usually unknown. Generally this problem is solved by using uniform distribution.

In some cases it is very difficult to calculate the expected value or we

do not know the input distribution but we have real input data. In these cases using the experimental analysis of the algorithms (on randomly generated or real data) can be a useful tool to analyze the behavior of the algorithms in the average case. The results often show that one can find significantly better algorithms than the algorithms which have the smallest known competitive ratio. Such experiments are presented in the case of scheduling in [4] where it is shown that the simple greedy list algorithm is better than many more sophisticated algorithms.

In the case of the online data acknowledgment problem parameter learning algorithms with experimental results on real data are presented in [44] and [48]. The extension of these parameter learning algorithms to a clustering problem in real time locating systems and their analysis can be found in [49]. In the problems studied in this thesis we do not have real data, but we executed some experimental tests on randomly generated data.

Sleator and Tarjan [53] suggested to evaluate the effectiveness of an online algorithm using *competitive analysis*. It compares the relative performance of an online and offline algorithm for the same problem instance. Specifically, the *competitive ratio* of an algorithm, is defined as the worst-case ratio of its cost divided by the optimal cost, over all possible inputs. The competitive ratio of an online problem is the best competitive ratio achieved by an online algorithm. Intuitively, the competitive ratio of an algorithm gives a measure on the quality of solutions produced by this algorithm on any input. (For a good introduction to competitive analysis, see [3, 8, 28, 43].)

Formally, many online problems can be described as follows (see [2]). A *request sequence* $I = I(1), I(2), ..., I(m)$ is presented to an online algorithm A. The requests $I(t)$, $1 \leq t \leq m$, must be served in their order of occurrence. More specifically, when serving request $I(t)$, algorithm

A does not know any request $I(t')$ with $t' > t$. Serving requests incurs cost, and the goal is to minimize the total cost paid on the entire input sequence. This setting can also be regarded as a *request-answer* game: An adversary generates requests and an online algorithm has to serve them one at a time.

Given a request sequence I, let $A(I)$ denote the cost incurred by the deterministic online algorithm A and let $OPT(I)$ denote the cost incurred by an optimal offline algorithm OPT. The algorithm A is called c-competitive if there exists a constant c such that

$$A(I) \leq c \cdot OPT(I)$$

for all request sequences I. The factor c is called the competitive ratio of A if c is the least such number.

1.2 Clustering problems

1.2.1 Clustering problems with the cost depending on the diameter of the cluster

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 distances between points in the same cluster, is minimized. In the online version, the demand points 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. 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 thesis the 1-dimensional and the 2-dimensional variants of the online clustering with variable sized clusters problem are considered which are presented in [17], [19] and [20]. In our model points of the 1-dimensional and 2-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 request points. The clusters are intervals (squares in 2D, since we use the l_∞ norm), 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 (side of the square in 2D). 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 or split. 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.

The similar offline problem was studied on trees (see [42, 45, 51, 55]), where the authors showed that the problem is polynomially solvable. A variant of the offline problem was studied in [12, 21], where the number of clusters is constant and the goal is to minimize the sum of diameters or the sum of radii, and a more generalized cost function was studied in [46].

A related offline problem in two dimensions was studied in [13]. In this paper clusters are rectangles which have a fixed area, with their lower edges placed on a common baseline. The goal is to cover the set of request points above the base line with a minimal number of clusters.

In [15] 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. An intermediate model, where the diameter is fixed in advance but the exact location can be modified is also studied. In [15] tight bounds are given on the competitive ratio of any online algorithm belonging to any of these variants. In the strict model tight bounds are given of $1 + \sqrt{2} \approx 2.414$ on the competitive ratio for the online problem, and tight bounds of 2 in the semi-online version (points are presented sorted by their location). In the intermediate model, the results of the strict 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 [15].

Several results are known on online clustering with fixed unit sized clusters. A study of online partitioning of points into unit sized clusters was presented by Charikar et al. [11]. 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 its arrival, and if it is assigned to a new ball, the exact location of this new 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 [10] 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) must always be covered by that ball but 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 [10], the authors showed that standard approaches lead to algorithms of competitive ratio 2 (some of which are valid for unit covering, too). 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 [56] an $\frac{11}{6}$-competitive randomized algorithm. In [26] an improved deterministic algorithm was given (with competitive ratio $\frac{7}{4}$) and in [23] an algorithm of competitive ratio $\frac{5}{3}$. Currently the best known lower bound is $\frac{8}{5}$ (see [26]).

In [10, 23, 26, 56] the two-dimensional problem is considered using the ℓ_∞ norm instead of the ℓ_2 norm. Thus, the "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 [23]). Other variants of the one-dimensional online unit clustering problem were studied in [25].

In [35] the multidimensional extension of the problem is studied with linear cost function, where again the cost of a cluster is the sum of a fixed setup cost and its radius. In this paper an $O(\log n)$-competitive algorithm is given for arbitrary metric spaces, and it is also proved that no online algorithm exists with smaller competitive ratio than $\Omega(\log n)$. The lower bound is also valid for the 2-dimensional Euclidean spaces. These bounds hold for both (strict and flexible) models. It is interesting that in the case

of linear cost function no constant competitive algorithm exists while in the case of square cost function we have constant competitive algorithms.

1.2.2 Online facility location – clustering with the cost: sum of the distances to the facility

In the facility location problem a metric space is given with a multiset of demand points (elements of the space). The goal is to find a set of facility locations in the metric space which minimizes the sum of the facility costs and assignment costs (the cost of assigning a request point to a given facility). The offline version is a well-known problem in combinatorial optimization.

The facility location and the closely related k-median problems have been widely studied in both computer science and operations research [6, 9, 22, 40]. In the facility location problem, usually the position of the facilities is fixed. A variant where facilities are movable has been introduced as the mobile facility location in [16, 36] as a generalization of the standard k-median and facility location problems. In [1] recently was presented a $(3 + \varepsilon)$-factor approximation algorithm, matching the best-known approximation factor for the k-median problem (see [6]). In [36] an approximation-preserving reduction was presented which shows that the mobile facility location generalizes the k-median problem.

In some applications (building a computer or sensor network, some clustering problems) the set of request points is not known in advance, demand points arrive one at a time and after their arrival the algorithm has to decide whether to open a new facility or to assign the demand to an existing facility without any information about the further demand points. This online version of facility location problem without facility movements is defined by Meyerson in [47]. In [47] it is not allowed to move a facility which is opened by the algorithm. In the paper an $O(\log n)$-

competitive randomized algorithm is presented for uniform facility cost and it is also showed that the algorithm is constant competitive against uniformly ordered input sequences (this second statement is also valid for the case of nonuniform facility cost). Moreover, it is proved that no constant-competitive online algorithm exists for the solution of the problem.

In this thesis a further version is considered which can also be used for the applications mentioned above. In this model the algorithm is allowed to move the facilities to other positions after the arrival of a demand point but the already opened facilities cannot be closed so increasing the number of facilities is an irrevocable decision that may later turn out to be a bad choice.

In [32] a similar model is investigated with the restriction that only one facility is allowed to move into new position after the arrival of each demand point. In [32] a 13.66-competitive memoryless algorithm is presented for uniform facility cost. The nonuniform facility cost is also investigated (changing the position of a facility changes its cost as well), in this general case a 48.6-competitive memoryless algorithm is given.

In [33] the problem is further investigated and an $O(\frac{\log n}{\log \log n})$-competitive deterministic algorithm is presented for uniform and nonuniform facility cost. Also, it is proved that no deterministic or randomized algorithm exists with smaller competitive ratio than $\Omega(\frac{\log n}{\log \log n})$.

In [5] a much simpler $O(\log n)$-competitive algorithm called *Partition* is presented for uniform facility cost and Eucledian spaces. That paper contains the first probabilistic and the first experimental analysis for the problem. It is shown that algorithm *Partition* is $O(1)$-competitive with high probability for any arrival order when customers are uniformly distributed or when they follow a distribution satisfying a smoothness property. A simple $O(\log n)$-competitive algorithm for nonuniform facility

cost and arbitrary metric spaces is presented in [31].

In [50] the authors propose the Online Connected Facility Location problem (OCFL), which is an online version of the Connected Facility Location problem (CFL). The CFL is a combination of the Uncapacitated Facility Location problem (FL) and the Steiner Tree problem (ST). In this paper a randomized $O(log_2 n)$-competitive algorithm is given for the OCFL via the sample-and-augment framework of Gupta, Kumar, Pál, and Roughgarden and previous algorithms for Online Facility Location (OFL) and Online Steiner Tree (OST). Also, the authors showed that the same algorithm is a deterministic $O(logn)$-competitive algorithm for a special case of the OCFL.

The incremental facility location is also a connected field to the online facility location. Namely, while in the online facility location the decisions of opening a facility at a particular location and of assigning a demand to some facility are irrevocable, in the incremental variant the algorithm can also merge existing facilities (and the corresponding demand clusters) with each other, and only the decision of assigning some demands to the same facility is fixed. In [30] a constant competitive algorithm is given for the incremental facility location problem. In a recent survey ([34]) the author discuss the previous work on online and incremental algorithms for facility location. The main results are: the competitive ratio for the online variant is $\theta(\frac{logn}{loglogn})$, where n is the number of demands, and that the competitive ratio for the incremental variant is $O(1)$.

A restricted variant of incremental facility location is presented in [39]. In this variant, similarly as in our problem, the facilities can be moved to reduce the overall cost. However unlike in our paper, moving a facility is not for free. The authors gave a deterministic online algorithm, which for parameters $\alpha \geq 1$ and $\beta \geq 0$ guarantees that at all times, the service cost is within a multiplicative factor α and an additive term β of the

optimal service cost and where also the movement cost is within a small multiplicative factor and an additive term of the optimal overall cost.

An interesting new approach to the facility location is recently presented in [24]. The authors propose to use information on the dynamics of the data to find stable partitions of the network into groups. For that purpose, they introduced a time-dependent, dynamic version of the facility location problem, that includes a switching cost when a client's assignment changes from one facility to another. This might provide a better representation of an evolving network, emphasizing the abrupt change of relationships between subjects rather than the continuous evolution of the underlying network. The authors showed that in realistic examples this model yields better fitting solutions than optimizing every snapshot independently.

2 Clustering problems in 1 dimension

2.1 The problem

In our model request points of the 1-dimensional Eucledian space arrive in an online fashion and the algorithm has to assign them to an already opened cluster or to a new one. The cost of each cluster is the sum of the constant setup cost scaled to 1 and the square of the length of the cluster (interval) and the goal is to minimize the total cost of all intervals.

We consider first the offline problem where the whole input is given in advance and we offer a simple dynamic programming algorithm to solve the problem optimally. The next subsection deals with the strict model where the size and the place of the cluster is irrevocable. Here we study the pure online and semi-online variant as well where the input is ordered. The fourth section in this chapter considers the flexible model where the algorithm can shift the cluster or expand it, as long as it contains all the points assigned to it. Both previously mentioned online variants are dealt with in this model, too.

2.2 The offline problem

As far as we know the offline clustering with the objective function considered in this thesis has 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 of the clusters (see [38] and its references for details) but there are also some results on the models where it depends on the powers of the diameters (see [7]), and even for general cost functions (see [46]). 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

[51]) 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.*

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 [15]), the problem on the line with square 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 uses an algorithm for the variation of the k-median problem and 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 the values 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 $O(n^3)$.

2.3 The strict model

In the strict variant of the clustering problem, the size and the exact location of the cluster must be fixed when it is initialized. We consider both the online and semi-online versions. "Semi-online" usually means that the algorithm knows something about the future demand points. In our case the points are sorted in ascending order.

2.3.1 The online model

The $GRID$ algorithm which uses a grid in the 1-dimensional space is defined in [25] for the problem of unit covering with rejection. For every integer $-\infty < k < \infty$, $GRID$ considers points arriving in the interval $I_k = (k, k+1]$ separately and independently from other points. Upon arrival of the first point in I_k, a new cluster is opened in the interval $[k, k+1]$ and all future points in this interval are assigned to this cluster.

In this work we propose the algorithm $GRID_a$: the size of the intervals is a parameter a. The 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.

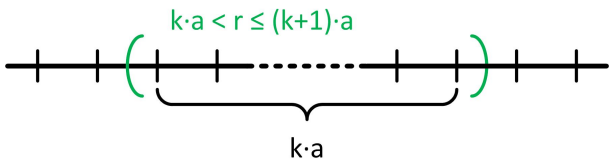

Figure 1: The optimal cluster intersects at most $k+2$ clusters from the grid

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

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

$F''(k^*) < 0$ and is concave for every a. $F'(k)$ 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 $r \leq a$. Then the optimal 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 $(2+2a^2)/(1+r^2) \leq 2+2a^2$.

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 this results that the competitive ratio is not smaller than $2 + 2a^2$.

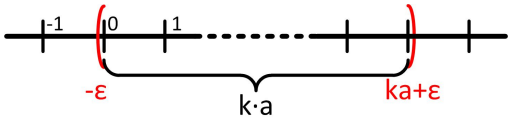

Figure 2: The interval with endpoints $-\varepsilon$ and $ka + \varepsilon$

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 achieved that the competitive ratio of $GRID_a$ is not less 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 $\frac{1}{2\sqrt{2}} \leq a \leq \frac{1}{\sqrt{2}}$, 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 $\frac{1}{2\sqrt{2}} \leq a \leq \frac{1}{\sqrt{2}}$ then $F(k) \leq 3$ for all integers $k > 1$ and also $2(1 + a^2) \leq 3$, 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.

□

Theorem 2 *The competitive ratio of any online algorithm for the strict model is at least* 2.3243.

Proof: Let us suppose that there exists an online algorithm A which has strictly less competitive ratio than X. Let the first request point be $p_1 = 0$ and $[x_1, x_2]$ the cluster (interval) which is opened by the

algorithm. Let $x = min\{-x_1, x_2\}$. Without loss of generality, it is possible to assume that $x = x_2$. From this time on, the sequence of requests is increasing (if $x = -x_1$, then a decreasing sequence should be used instead). Let $a = x_2 - x_1$ and the second request point be $x_2 + \varepsilon$. Then the online algorithm opens a new interval with diameter b (the end point of the second interval is x_3). Then the cost of the optimal solution is $OPT(I_2) \leq 1 + (\frac{a}{2} + \varepsilon)^2 \to 1 + \frac{a^2}{4}$, if $\varepsilon \to 0$ while the cost of the online algorithm is $A(I_2) = 2 + a^2 + b^2$. Then

$$X > \frac{A(I_2)}{OPT(I_2)} \to \frac{2 + a^2 + b^2}{1 + (\frac{a}{2})^2}.$$

The third request point arrives at ε distance from the second cluster (i.e. at $x_3 + \varepsilon$) and the online algorithm opens for it a new cluster with diameter c (the rightmost point that the third interval covers is x_4). Then

$$X > \frac{A(I_3)}{OPT(I_3)} \geq \frac{3 + a^2 + b^2 + c^2}{1 + (\frac{a}{2} + b + 2\varepsilon)^2} \to \frac{3 + a^2 + b^2 + c^2}{1 + (\frac{a}{2} + b)^2}$$

We follow the pattern to the k-th request point, so we obtain a problem with k variables:

$$\frac{k + {x_1}^2 + {x_1}^2 + \cdots + {x_k}^2}{1 + (x_1/2 + x_2 + \cdots + x_{k-1})^2} < X$$

where we have to minimize the maximum of the above problem for $x_k \geq 0, k = 2, 3, ...$ With MATLAB's NLP solver $fminimax$ we proved that the the lower bound is 2.3243 if 5 request points are used and only a small further increasing can be obtained if we continue adding new points.

□

2.3.2 The semi-online model

In the semi-online strict 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 3 *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 greater than or equal to the left endpoint of this optimal cluster. The possible 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 $F''(k^*)<0$ and is concave 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 $r \leq a$. Then the optimal cluster intersects at most 2 $SOSM_a$ clusters, but we have to consider only the cluster which left endpoint is greater 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 $\frac{1+a^2}{1+r^2} \leq 1 + a^2$.

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 $1 + a^2$ so the competitive ratio is not less 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 $1 + (ka + 2\varepsilon)^2$. Algorithm $SOSM_a$ 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. As a result we obtained that the competitive ratio of $SOSM_a$ is not less 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 accomplished if $\frac{1}{\sqrt{5}} \leq a \leq 1$, 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 $\frac{1}{\sqrt{5}} \leq a \leq 1$ then $F(k) \leq 2$ for all integers $k > 1$ and also $1 + a^2 \leq 2$, 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 4 *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 $a_2 \geq 0$.

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

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

$$\geq \frac{3 + a_1^2 + a_2^2}{1 + (a_1 + a_2)^2} \geq \frac{3 + 0.83035^2 + 0.30817^2}{1 + (0.83035 + 0.30817)^2} > 1.6481$$

    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:

$$\frac{A(I)}{OPT(I)} \geq \frac{2 + a_1^2 + a_2^2}{1 + (a_1 + \varepsilon)^2} \to \frac{2 + a_1^2 + a_2^2}{1 + a_1^2} \geq$$
$$\geq \frac{2 + 0.83035^2 + 0.30817^2}{1 + 0.83035^2} > 1.6481$$

    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 $a_2 > 0.77894$.

We have discussed all the possibilities, therefore the theorem holds.

□

2.4 The flexible model

2.4.1 The online problem

In the case of 1 dimension with the linear cost the ECC (extend closed cluster) algorithm (see [15]) has competitive ratio $\frac{1+\sqrt{5}}{2} \approx 1.618$. It is a straightforward idea to use this algorithm which worked in the case of linear cost function. The modified algorithm ECC for the square cost is as follows:

Observation 1 *The algorithm* ECC *is not constant competitive if the cost is the length of the interval squared.*

Proof. Consider the following sequence of requests which contains $n + 1$ points. We denote the sequence by I_n. Start with 0, and let the $(i + 1)$-th request be $\sqrt{i}$. Then it is easy to see by induction that the algorithm extends the cluster in each step, thus it ends with one cluster of size $\sqrt{n}$

Algorithm 3 Algorithm ECC

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, let Q be the cluster which can be extended with the smallest cost to cover p.
   (a) If the cost of this extension is at most 1, then extend Q, assign p to it.
   (b) Otherwise, open a new cluster and assign p to the new cluster. In this case this new cluster consists of a single point p.

and $ECC(I_n) = 1 + n$. On the other hand an offline algorithm can cover the requests by $\lceil\sqrt{n}\rceil$ unit sized clusters, therefore $OPT(I_n) \leq 2 \cdot \lceil\sqrt{n}\rceil$. $ECC(I_n)/OPT(I_n)$ tends to ∞ as n increases and this completes the proof. □

As the proof shows an algorithm should limit the size of the clusters. The following extension of the algorithm $GRID_a$ satisfies this property.

Algorithm 4 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 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. This new cluster consists of a single point p.
   (b) Otherwise, extend the cluster contained in the interval to cover p.

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

Proof. Consider an arbitrary request 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 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 costs of the intervals covered by the optimal cluster in these end cells (square of the length of the interval) 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 covers at least that interval which $FGRID_a$ covers in the given end 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 $\frac{1}{2} \cdot (1+a^2)$. As a result we achieved 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 $1 + k^2a^2 + A$. The ratio of these costs is:

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

If $k = 1$ then this ratio is 2 for each a. For $k > 1$ and $a > \frac{1}{\sqrt{5}}$ this ratio is less 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 covers at least the interval which $FGRID_a$ covers. 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 $\frac{1+a^2}{2}$. 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 $1 + r^2$. The ratio of these costs is:

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

On the other hand $1 + a^2 \leq 2$ if $a \geq \frac{1}{\sqrt{5}}$.

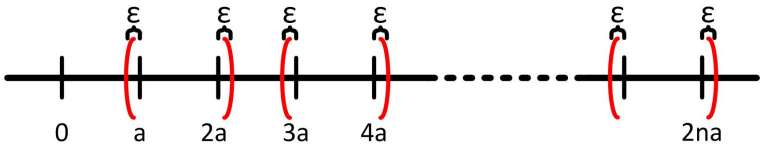

Figure 3: The competitive ratio of $FGRID_a$: the n intervals with endpoints $(2i-1)a-\varepsilon$ and $2ia+\varepsilon$, $i = 1, ..., n$

Now we prove the tightness of the competitive ratio. 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 $FGRID_a(I_n) \geq (1+a^2)(2n-1)+2(1+\varepsilon^2)$. Since the ratio $FGRID_a(I_n)/OPT(I_n)$ tends to 2 as ε tends to 0 and n tends to ∞, this results that the algorithm is not better than 2-competitive.

□

Theorem 6 *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 less 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": the first cluster 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 also a contradiction.

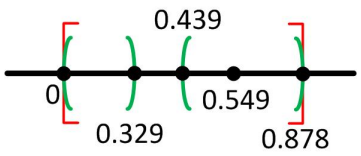

Figure 4: Lower bound in the flexible model: the cost of the online and offline algorithms

We conclude that in both cases the competitive ratio of the online algorithm is at least 1.2993.

□

2.4.2 The semi-online model

Theorem 7 *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 less 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 the points 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: $[p_1, p_1]$ and $[p_2, p_3]$. The cost is $2 + 0.47422^2 = 2.2248846084$, so we conclude:

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

  .

- 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 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 $FGRID_a$).

3 Clustering problems in 2 dimensions

3.1 Introduction

In the model considered in this thesis request points of the 2-dimensional Eucledian space arrive one by one and the algorithm has to define a cluster for them: an already opened cluster or a new one. The cost of each cluster is the sum of the constant setup cost scaled to 1 and the square of the length of the side of the cluster (square) and the objective function is to minimize the total cost of all clusters. The clusters are squares since we use the l_∞ norm.

In the first and second section the strict and flexible models are studied with the same properties as seen in the previous chapter. The third section in this chapter presents an experimental analysis on the grid parameter. The fifth section mentions a relaxation of the objective function: the cost is the p-th power rather than the square of the length of the diameter of the cluster, also a generalization on the dimension of the space is given.

3.2 The strict model

The $GRID$ algorithm which uses a grid in the 1-dimensional space is defined in [10] and [25] for the unit clustering problems, and later it is studied in [17] for the 1-dimensional version of our problem. In the 2-dimensional case the algorithm works as follows. It covers the space with a square-grid of $a \times a$ sized cells. If a request arrives which is not covered by any of the existing clusters then the algorithm creates a new one which is the closed cell containing the request.

3.2.1 The improved algorithm

We investigate a more sophisticated algorithm with a better competitive ratio which we call SHIFT$(1/3)$GRID$_a$. This algorithm is also based on a grid of $a \times a$ cells which cover the space, but here each row is shifted right by $a/3$ compared to the row below. Thus the cells in the grid are:

- the cells with the corners $(ja, 3ia), (ja, (3i+1)a), ((j+1)a, 3ia)$, $((j+1)a, (3i+1)a)$ for the integers i and j,
- the cells with the corners $((j+1/3)a, (3i+1)a), ((j+1/3)a, (3i+2)a)$, $((j+4/3)a, (3i+1)a), ((j+4/3)a, (3i+2)a)$ for the integers i and j,
- the cells with the corners $((j+2/3)a, (3i+2)a)$, $((j+2/3)a, (3i+3)a)$, $((j+5/3)a, (3i+2)a)$, $(j+5/3)a, (3i+3)a)$ for the integers i and j.

If a request arrives which is not covered by any of the clusters then the algorithm creates a new cluster which is the closed cell containing the request. The competitive ratio of SHIFT$(1/3)$GRID$_a$ is determined by the following theorem.

Theorem 8 *If $\sqrt{1/2} \leq a \leq \sqrt{27/29}$, then the competitive ratio of algorithm* SHIFT$(1/3)$GRID$_a$ *is* 7.

Proof. Suppose that $\sqrt{1/2} \leq a \leq \sqrt{27/29}$ holds. Consider an arbitrary sequence and an optimal solution for it, denote this solution by OPT. We investigate the clusters of OPT separately. Consider an arbitrary cluster. Let r denote the length of the side of this square. We distinguish the following cases depending on r.

Case 1 Suppose that $r \leq a/3$. Then this optimal cluster can intersect at most two rows. If it intersects two cells in one of these rows then it can intersect only one in the other, thus it cannot intersect more than 3 cells from the grid. Therefore the cost of the online algorithm on the

requests of this cluster is at most $3(1+a^2)$. The optimal cost is at least 1, thus in this case the competitive ratio is at most $3(1+a^2)$ which is at most 7 if $a \leq \sqrt{4/3}$.

Case 2 Suppose that $a/3 < r \leq a$. Then this optimal cluster cannot intersect more than two rows and it can intersect two cells in both of them. Therefore the cost of the online algorithm on the requests of this cluster is at most $4(1+a^2)$. The optimal cost is at least $1+a^2/9$, therefore in this case the competitive ratio is at most $4(1+a^2)/(1+a^2/9)$ which is at most 7 if $a \leq \sqrt{27/29}$.

Case 3 Suppose that $ka \leq r \leq (k+1/3)a$ for a $k \geq 1$. Then the optimal cluster can intersect at most $k+2$ rows from the grid. If at some row it intersects $k+2$ cells then in the neighboring rows it can intersect only $k+1$ cells. Therefore all together it intersects at most $\lceil (k+2)/3 \rceil (k+2) + (k+2-\lceil (k+2)/3 \rceil)(k+1)$ cells. This yields that the online cost is at most $\big(\lceil (k+2)/3 \rceil (k+2) + (k+2-\lceil (k+2)/3 \rceil)(k+1)\big)(1+a^2)$. On the other hand the optimal offline cost is at least $1+(ka)^2$. To analyze the ratio of the costs we distinguish the following subcases.

Case 3.1 Suppose that $k = 3t+1$ for some $t \geq 0$. Then the ratio of the online and optimal costs is at most

$$\frac{((t+1)(3t+3)+(2t+2)(3t+2))(1+a^2)}{1+((3t+1)a)^2}.$$

If $t = 0$ then this ratio is 7. If $t \geq 1$, then this ratio is less than 7 if and only if

$$a^2 \geq \frac{9t+16}{54t+26}$$

is valid. The right side of the inequality is maximal on $t \geq 1$ if $t = 1$. Therefore if $a \geq \sqrt{25/80} = \sqrt{5/16}$ the inequality holds because $\sqrt{5/16} < \sqrt{1/2}$. This results in the upper bound of 7 on the competitive ratio of the algorithm in this case.

Case 3.2 Suppose that $k = 3t + 2$ for some $t \geq 0$. Then the ratio of the online and optimal costs is at most

$$\frac{((t+2)(3t+4) + (2t+2)(3t+3))(1+a^2)}{1 + ((3t+2)a)^2}.$$

This ratio is at most 7 if and only if

$$a^2 \geq \frac{9t^2 + 22t + 7}{54t^2 + 62t + 14}$$

is valid. The right side of the inequality is $1/2$ if $t = 0$ and it is less than $1/2$ if $t \geq 1$, thus we achieve that the algorithm is 7 competitive in this case if $a \geq \sqrt{1/2}$.

Case 3.3 Suppose that $k = 3t$ for some $t \geq 1$. Then the ratio of the online and optimal costs is at most

$$\frac{((t+1)(3t+2) + (2t+1)(3t+1))(1+a^2)}{1 + (3ta)^2}.$$

This ratio is at most 7 if and only if

$$a^2 \geq \frac{9t^2 + 10t - 4}{54t^2 - 10t - 3}$$

is valid. The right side of the inequality is maximal on $t \geq 1$ if $t = 1$ where it is $15/41$, therefore if $a \geq \sqrt{1/2}$ the algorithm is 7-competitive.

Case 4 Suppose that $(k + 1/3)a < r \leq (k + 2/3)a$ for a $k \geq 1$. Then again the optimal cluster can intersect at most $k+2$ rows from the grid. If at some row it intersects $k + 2$ cells then in one of the neighboring rows it can intersect only $k + 1$ cells. Therefore all together it intersects at most $\lfloor (k+2)/3 \rfloor (k+1) + (k+2 - \lfloor (k+2)/3 \rfloor)(k+2)$ cells. Therefore the online cost is at most $\big(\lfloor (k+2)/3 \rfloor (k+1) + (k+2 - \lfloor (k+2)/3 \rfloor)(k+2)\big)(1+a^2)$ the optimal cost is at least $1 + ((k + 1/3)a)^2$. To analyze the ratio of the costs we distinguish the following subcases.

Case 4.1 Suppose that $k = 3t + 1$ for some $t \geq 0$. Then the ratio of the online and optimal costs is at most

$$\frac{((t+1)(3t+2) + (2t+2)(3t+3))(1+a^2)}{1 + ((3t+4/3)a)^2}.$$

This ratio is at most 7 if and only if

$$a^2 \geq \frac{9t^2 + 17t + 1}{54t^2 + 39t + 40/9}$$

is valid. The right side of the inequality is $9/40$ if $t = 0$ which is less than $1/2$ and if $t \geq 1$ the expression is also less than $1/2$, therefore we obtain the algorithm is 7-competitive in this case.

Case 4.2 Suppose that $k = 3t + 2$ for some $t \geq 0$. Then the ratio of the online and optimal costs is at most

$$\frac{((t+1)(3t+3) + (2t+3)(3t+4))(1+a^2)}{1 + ((3t+7/3)a)^2}.$$

The upper bound of this ratio is 7 if and only if

$$a^2 \geq \frac{9t^2 + 23t + 8}{54t^2 + 75t + 208/9}$$

is valid. If $t = 0$ the right side of the inequality is the greatest: $\frac{72}{208} < \frac{1}{2}$ and it is also less than $1/2$ if $t \geq 1$, thus we conclude that the algorithm is 7 competitive in this case.

Case 4.3 Suppose that $k = 3t$ for some $t \geq 1$. Then the ratio of the online and optimal costs is at most

$$\frac{(t(3t+1) + (2t+2)(3t+2))(1+a^2)}{1 + ((3t+1/3)a)^2}.$$

This ratio is at most 7 if and only if

$$a^2 \geq \frac{9t^2 + 11t - 3}{54t^2 + 3t - 29/9}$$

is valid. The right side of the inequality is less than $1/2$ if $t \geq 1$, therefore if $a \geq \sqrt{1/2}$ we attain that the algorithm is again 7-competitive.

Case 5 Suppose that $(k + 2/3)a < r \leq (k + 1)a$ for some $k \geq 1$. Then again the optimal cluster can intersect at most $k + 2$ rows from the grid and in each row it can intersect at most $k + 2$ cells. Therefore the cost of the online algorithm is at most $(k + 2)^2(1 + a^2)$ the optimal cost is at least $1 + ((k + 2/3)a)^2$, thus the competitive ratio is at most

$$\frac{(k + 2)^2(1 + a^2)}{1 + ((k + 2/3)a)^2}.$$

If and only if

$$a^2 \geq \frac{k^2 + 4k - 3}{6k^2 + \frac{16}{3}k - \frac{8}{9}}$$

is valid, this ratio is at most 7. This inequality holds for $k \geq 1$ thus the result follows in this case, too.

We considered all of the possible values for r, therefore we proved that the algorithm is 7-competitive.

It is easy to see that the competitive ratio of the algorithm is not better than 7. Let $\varepsilon \geq 0$ be a very small positive number. Consider the following demand points $(-\varepsilon a, -\varepsilon a)$, $(-\varepsilon a, a/2)$, $(-\varepsilon a, (1 + \varepsilon)a)$, $(a/2, a/2)$, $((1 + \varepsilon)a, -\varepsilon a)$, $((1 + \varepsilon)a, a/2)$, $((1 + \varepsilon)a, (1 + \varepsilon)a)$. We show this set of demand points on Figure 5. These points are located in 7 different cells, therefore the cost of the online algorithm is $7(1 + a^2)$. The optimal solution can use 1 square with sides $(1 + 2\varepsilon)a$ to cover all of the requests, thus the optimal cost is at most $1 + (1 + 2\varepsilon)^2 a^2$. If ε tends to 0 this example shows that the algorithm cannot be better than 7 competitive.

We also note that the bounds on the parameter a are tight as well. If $a \leq \sqrt{1/2}$ then we can construct an example based on Case 3.2 where

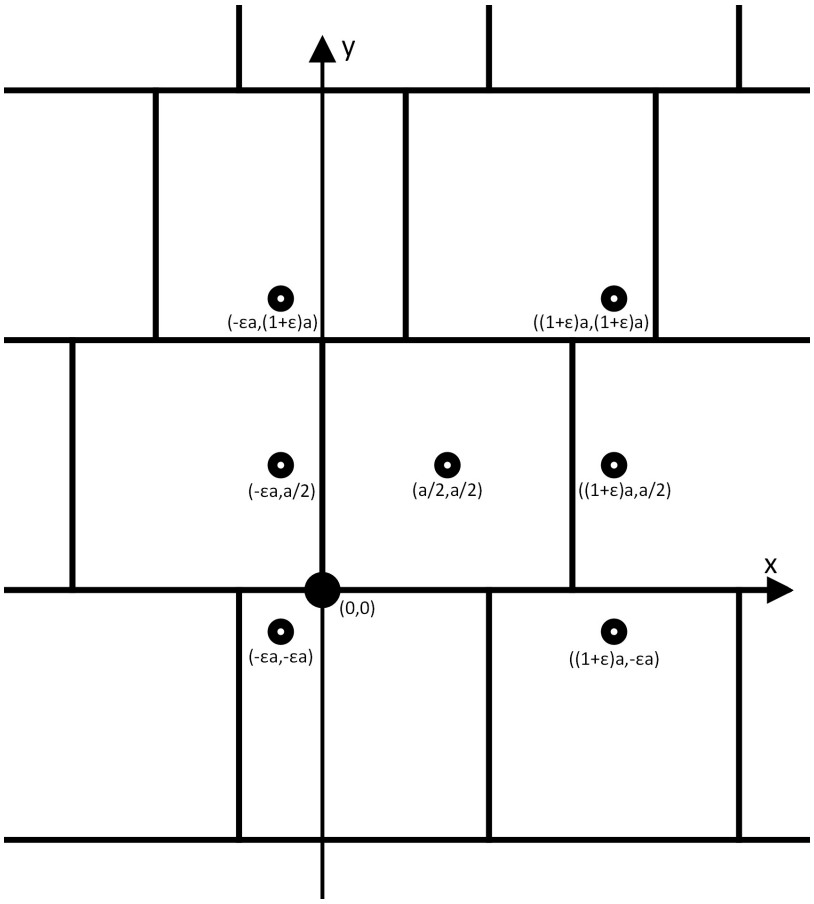

Figure 5: The tightness example for theorem 8

the algorithm is worse than 7-competitive, if $a \geq \sqrt{27/29}$ then we can use Case 2 to construct an example where the algorithm is also not 7-competitive.

□

Remark A similar but slightly simpler analysis shows that using a grid without shifts yields an algorithm which is 9-competitive for the best choice of the parameter a: $\sqrt{7/20} \leq a \leq \sqrt{5}/2$. One can also prove that using shifts of $a/2$ in the grid gives an 8-competitive algorithm for the best choice of a: $\sqrt{1/3} \leq a \leq \sqrt{5/3}$. But using smaller shifts than $a/3$ does not improve the competitive ratio further.

3.2.2 Lower bound

Theorem 9 *The competitive ratio of any online algorithm for the strict model is at least* 2.768.

Proof. Assume by contradiction that there exists an online algorithm A which has strictly less competitive ratio than 2.768. We will investigate the possible decisions of the algorithm. In each case the smallest possible competitive ratio is analyzed with the help of the function $fminimax$ in MATLAB, with the constraints that all variables a, b,..., h are nonnegative.

Let the first request point be $p_1 = (0, 0)$ and let (x_1, y_1) be the lower left and (x_2, y_2) the upper right corner of the cluster (square) which is opened by the algorithm. Let $a = x_2 - x_1$. Thus the competitive ratio of the online algorithm is at least $R_1 = 1 + a^2$. Let $x = min\{-x_1, x_2\}$ and $y = min\{-y_1, y_2\}$. Then $x \leq a/2$ and $y \leq a/2$ and without loss of generality, it is possible to assume that $x = x_2$ and $y = y_2$. From this time on, the x or y coordinates in the sequence of requests are increasing (if $x = -x_1$ and/or $y = -y_1$, then decreasing sequences of request points should be used instead). Let the second request point be $(a/2 + \varepsilon, 0)$ and the third request point be $(0, a/2 + \varepsilon)$. Then the cost of the optimal solution is at most $1 + (\frac{a}{2} + \varepsilon)^2 \to 1 + \frac{a^2}{4}$, if $\varepsilon \to 0$. The online algorithm A has not covered the new points with its first cluster, therefore it has two possibilities:

1. Suppose that A opens two new clusters to cover these points with sides b and c. Then the cost of the online algorithm is $3 + a^2 + b^2 + c^2$. In this case the sequence ends and the limit of the ratio of the online and optimal cost tends to a number which is at least 3. This leads to contradiction.

2. Suppose that the algorithm opens only one new cluster which covers

both of the new request points. Then its side must be larger than $a/2$, suppose it is $\frac{a}{2}+b$. The cost of the online algorithm is $2+a^2+(\frac{a}{2}+b)^2$. Thus we obtained that if ε tends to 0 then the competitive ratio is at least

$$R_2 = \frac{2 + a^2 + (\frac{a}{2} + b)^2}{1 + \frac{a^2}{4}}.$$

Now another two points arrive at ε distance from the last cluster, one on the x and the other on the y axis. They arrive at the points $(a/2+b+\varepsilon, 0)$ and $(0, a/2+b+\varepsilon)$. The optimal offline algorithm can cover all points by one cluster thus its cost is at most $1+(\frac{a}{2}+b+\varepsilon)^2 \to 1 + (\frac{a}{2} + b)^2$. The online algorithm covers none of these new points thus it has again two possibilities:

(a) Suppose that it opens two clusters to cover the new points with sides c and d. Then the cost of the online algorithm is $4 + a^2 + (\frac{a}{2} + b)^2 + c^2 + d^2$. The competitive ratio is at least

$$R_3 = \frac{4 + a^2 + (\frac{a}{2} + b)^2 + c^2 + d^2}{1 + (\frac{a}{2} + b)^2}.$$

Without loss of generality we can suppose that $c \geq d$. Now another two points are given at $(a/2+b+c+2\varepsilon, 0)$ and $(0, a/2+b+c+2\varepsilon)$. Then the cost of the optimal offline solution is at most $1 + (\frac{a}{2} + b + c + 2\varepsilon)^2 \to 1 + (\frac{a}{2} + b + c)^2$. The new points are not covered by the online algorithm therefore it has the following two cases:

i. Suppose that A opens two new clusters. Denote by e the sides of the larger cluster and by f the sides of the smaller one. Then the competitive ratio is at least

$$R_4 = \frac{6 + a^2 + (\frac{a}{2} + b)^2 + c^2 + d^2 + e^2 + f^2}{1 + (\frac{a}{2} + b + c)^2}.$$

Now the request sequence continues with two more points at $(a/2+b+c+e+3\varepsilon, 0)$ and $(0, a/2+b+c+e+3\varepsilon)$. The limit of the cost of the optimal algorithm is at most $1+(\frac{a}{2}+b+c+e)^2$ and algorithm A has the following possibilities:

A. Suppose that it opens two more clusters to cover these points with sides g and h. The clusters of the online algorithm are presented on Figure 6. We note that in the figure the cluster of size e is on the x-axis and the cluster of size f is on the y axis, but as we do not use this in the proof, they could be in the opposite way. Then the cost of A is $8 + a^2 + (\frac{a}{2} + b)^2 + c^2 + d^2 + e^2 + f^2 + g^2 + h^2$ and the competitive ratio is at least

$$R_5 = \frac{8 + a^2 + (\frac{a}{2} + b)^2 + c^2 + d^2 + e^2 + f^2 + g^2 + h^2}{1 + (\frac{a}{2} + b + c + e)^2}.$$

If we minimize $\max\{R_1, R_2, R_3, R_4, R_5\}$ on the nonnegative variables $a, \ldots, h$ using $d \leq c$ and $f \leq e$, we obtain that the objective value is not less than 2.768, contradiction.

B. Suppose that one new cluster is opened with a side of at least $\frac{a}{2} + b + c + e + 3\varepsilon$ to cover these requests. The clusters of the online algorithm are presented on Figure 7. Then the competitive ratio is at least

$$R_6 = \frac{7 + a^2 + (\frac{a}{2} + b)^2 + c^2 + d^2 + e^2 + f^2 + (\frac{a}{2} + b + c + e)^2}{1 + (\frac{a}{2} + b + c + e)^2}$$

.

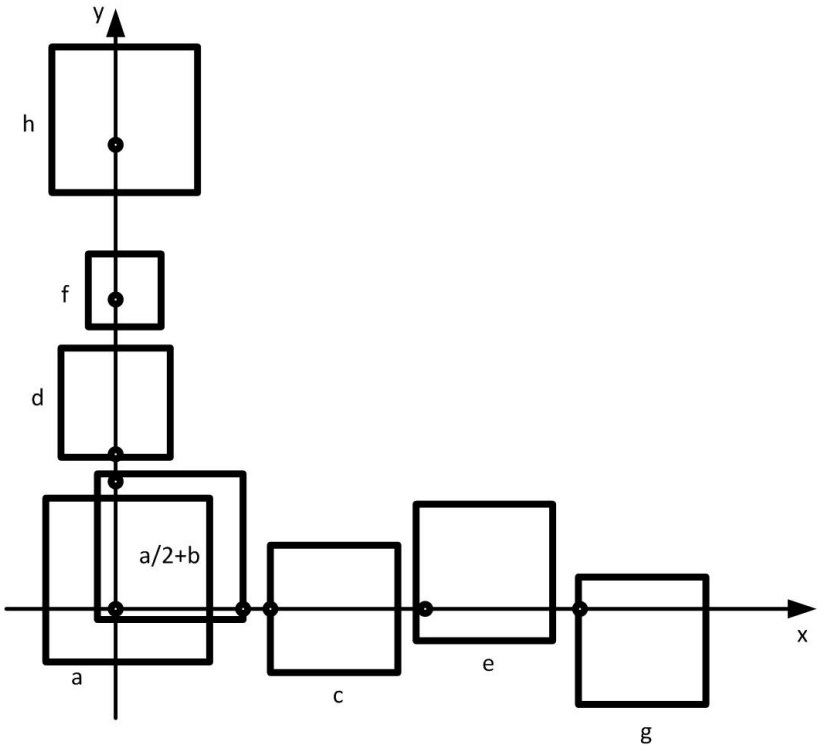

Figure 6: The clusters in case 2/a/i/A

If we minimize $\max\{R_1, R_2, R_3, R_4, R_6\}$ on the nonnegative variables $a, \ldots, h$ using $d \leq c$ and $f \leq e$, the objective value is not less than 2.7727. Therefore we achieved a contradiction in this case as well.

ii. Suppose that A opens one new cluster with sides at least $\frac{a}{2} + b + c + 2\varepsilon$ to serve the points $(a/2 + b + c + 2\varepsilon, 0)$ and $(0, a/2 + b + c + 2\varepsilon)$. Its cost is at least $5 + a^2 + (\frac{a}{2} + b)^2 + c^2 + d^2 + (\frac{a}{2} + b + c)^2$ so the competitive ratio is at least

$$R_7 = \frac{5 + a^2 + (\frac{a}{2} + b)^2 + c^2 + d^2 + (\frac{a}{2} + b + c)^2}{1 + (\frac{a}{2} + b + c)^2}$$

Minimizing $\max\{R_1, R_2, R_3, R_7\}$ on the nonnegative variables a, b, c and d using $d \leq c$, we accomplish that the objective value is at least 2.7894, contradiction.

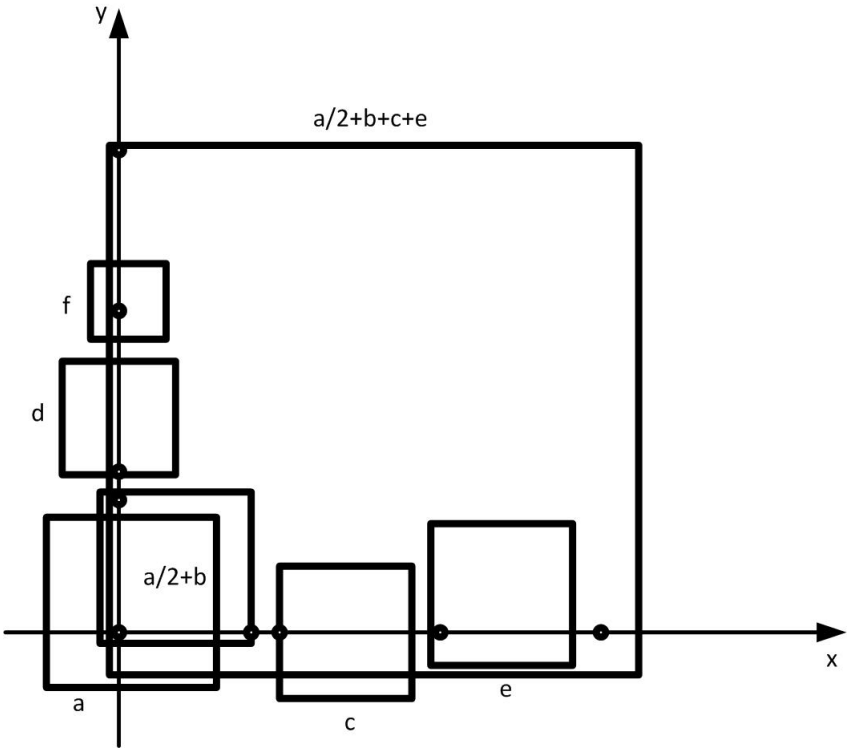

Figure 7: The clusters in case 2/a/i/B

(b) Suppose that A opens one new cluster with sides $\frac{a}{2}+b+c$ to serve the points $(a/2+b+\varepsilon, 0)$ and $(0, a/2+b+\varepsilon)$. Then the competitive ratio is at least

$$R_8 = \frac{3+a^2+(\frac{a}{2}+b)^2+(\frac{a}{2}+b+c)^2}{1+(\frac{a}{2}+b)^2}$$

In this case two more points arrive at $(a/2+b+c+\varepsilon, 0)$ and $(0, a/2+b+c+\varepsilon)$. The cost of the optimal algorithm is at most $1+(\frac{a}{2}+b+c+\varepsilon)^2 \to 1+(\frac{a}{2}+b+c)^2$. The online algorithm has two possibilities:

i. Suppose that A opens two new clusters for the new request points with sides d and e. Then the competitive ratio is at least

$$R_9 = \frac{5 + a^2 + (\frac{a}{2} + b)^2 + (\frac{a}{2} + b + c)^2 + d^2 + e^2}{1 + (\frac{a}{2} + b + c)^2}$$

If we minimize $\max\{R_1, R_2, R_8, R_9\}$ on the nonnegative variables $a, \ldots, e$, it follows that the objective value is at least 2.7693, contradiction.

ii. Suppose that A opens one new cluster which covers both new points with the sides at least $\frac{a}{2} + b + c + d$. Then the competitive ratio is at least

$$R_{10} = \frac{4 + a^2 + (\frac{a}{2} + b)^2 + (\frac{a}{2} + b + c)^2 + (\frac{a}{2} + b + c + d)^2}{1 + (\frac{a}{2} + b + c)^2}$$

which is not less than 3 which is a contradiction.

We investigated all of the possible decisions of the algorithm and we received a contradiction in all cases, therefore we proved the theorem. □

Remark: A small further increasing of the value (in the fourth decimal) can be obtained by continuing to add new points.

3.3 The flexible model

3.3.1 Algorithms

In the case of 1 dimension the ECC (extend closest cluster) algorithm (see [15]) has competitive ratio $\frac{1+\sqrt{5}}{2} \approx 1.618$. It is a straightforward idea to extend it to 2-dimensions. In the section 2.4.1 it is shown that the extended algorithm ECC is not constant competitive with the square cost, therefore we conclude that it is not constant competitive in 2 dimensions either.

The following extension of the algorithm SHIFT$(1/3)$GRID$_a$ is again based on the shifted grid of $a \times a$ cells defined in Section 3.2. The difference

is that we do not define the cluster immediately as the whole cell, the clusters are only subsquares of the cells of the grid or consist of a single point.

Algorithm 5 Algorithm $Shift(1/3)FGRID_a$

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, consider the cell from the grid which contains p.
   (a) If this cell does not contain a cluster, then open a new cluster and assign p to this new cluster. In this case this cluster consists of a single point p.
   (b) Otherwise, extend the cluster contained in the cell to cover p.

Theorem 10 *Let* $x = (63 + 9\sqrt{521})/118 \approx 2.2748$ *be the positive root of the following equation*

$$\frac{2+2x}{1+x/9} = \frac{6+44x/9}{1+x}.$$

If $a = \sqrt{x} \approx 1.508$, *then* SHIFT$(1/3)$FGRID$_a$ *is* C*-competitive where*

$$C = \frac{2+2x}{1+x/9} = \frac{6+44x/9}{1+x} \approx 5.228.$$

Proof. Suppose that $a = \sqrt{x}$. Consider an arbitrary sequence and an optimal solution for it, denoted by OPT. The clusters of OPT will be investigated separately. Consider an arbitrary cluster. Let r denote the length of the side of this square. We will prove that the online cost which is assigned to this optimal cluster is at most C times $1 + r^2$. We will conclude the proof by a case disjunction as it is done in the proof of Theorem 8. The only difference is that in some cases we handle separately the corner cells of the optimal cluster. These cells are the leftmost and the rightmost cells in the top and lowermost rows of the shifted grid which

are intersected by the optimal cluster. They are important since they are usually not completely covered by the online algorithm. If a corner cell is intersected only by one optimal cluster then the part which is covered by the online algorithm depends on the size of this intersection. If the cell is intersected by more optimal clusters, then the whole cell could be covered by the online algorithm but in this case we divide this $1 + a^2$ cost between at least two optimal clusters, therefore we have to count only $(1 + a^2)/2$ cost in the analysis. Now consider the following cases.

Case 1 Suppose that $r \leq a/3$. Then this optimal cluster can intersect at most two rows. If it intersects two cells in one of these rows then it can intersect only one in the other, thus it cannot intersect more than 3 cells from the grid. Then all of these three cells are corner cells. If such a cell has no intersection with other optimal clusters then only a subsquare of the cell of size $a/3 \times a/3$ can be covered by the online algorithm and its cost is $1 + a^2/9$. If a cell intersects more than one optimal clusters then we have to count $(1 + a^2)/2 \geq 1 + a^2/9$ cost here. Therefore the cost of the online algorithm assigned to the requests of this cluster is at most $3(1 + a^2)/2$. Since the optimal cost is at least 1, in this case the competitive ratio is at most $3(1 + a^2)/2$ which is less than C.

Case 2 Suppose that $a/3 < r \leq a$. Then this optimal cluster cannot intersect more than two rows and it can intersect in both of them at most two cells. All of them are corner cells. Suppose that $r = a/3 + y$ for some $0 \leq y \leq 2a/3$. In each of these cells the cost which is assigned to the cell from the online algorithm is at most $\max\{1 + (a/3 + y)^2, (1 + a^2)/2\}$. The first value is achieved if only one optimal cluster is intersected, the second is attained if more optimal clusters are intersected. On the other hand the optimal cost is $1 + (a/3 + y)^2$. Now examine how the ratio of the online and offline costs is changing if a positive y is used. The increase in the denominator is $2ay/3 + y^2$ and the increase in the numerator is at

most $4(2ay/3 + y^2)$. Since we would like to prove that the ratio is at most C and $C > 4$ it follows that we can assume that $y = 0$: if the ratio with $y = 0$ is less than C, then it is less for every y. If $y = 0$ the online cost is at most $4(1 + a^2)/2 = 2(1 + a^2)$, the optimal cost is $1 + a^2/9$. Therefore the competitive ratio is at most

$$\frac{2 + 2a^2}{1 + a^2/9} = C.$$

Case 3 Suppose that $ka \leq r \leq (k + 1/3)a$ for some $k \geq 1$. Then the optimal cluster can intersect at most $k + 2$ rows from the grid. If at some row it intersects $k + 2$ cells then in the neighboring rows it can intersect only $k + 1$ cells. Therefore all together it intersects at most $\lceil (k + 2)/3 \rceil (k + 2) + (k + 2 - \lceil (k + 2)/3 \rceil)(k + 1)$ cells. To analyze the ratio of the costs we distinguish the following subcases.

Case 3.1 Suppose that $k = 3t + 1$ for some $t \geq 0$. Then the online cost for the cells which are not corner is at most $((t + 1)(3t + 3) + (2t + 2)(3t + 2) - 4)(1 + a^2)$. Now consider the corner cells. Suppose that the size of the optimal cluster is $ka + y$ for some $0 \leq y \leq a/3$. Then the costs in two corner cells are $\max\{1 + (a/3 + y)^2, (1 + a^2)/2\}$ and in the other corner cells $\max\{1 + (2a/3 + y)^2, (1 + a^2)/2\}$. The optimal cost is $1 + (ka + y)^2$. Again observe that if the ratio with $y = 0$ is less than C, then it is less for every y. (Changing from 0 to a positive y the increase in the denominator is $2aky + y^2$ and the increase in the numerator is at most $4(4ay/3 + y^2)$.) Therefore we can suppose that $y = 0$ and in this case the online cost is at most $\big((t+1)(3t+3)+(2t+2)(3t+2)-4\big)(1+a^2)+2(1+a^2)/2+2(1+(2a/3)^2)$. Therefore the ratio of the online and offline costs is at most

$$\frac{((t + 1)(3t + 3) + (2t + 2)(3t + 2) - 4 + 1)(1 + a^2) + 2 + 8a^2/9}{1 + ((3t + 1)a)^2}.$$

If $t = 0$ then this ratio is $(6 + 44a^2/9)/(1 + a^2) = C$. If $t \geq 1$, then this

ratio is less than $5 \leq C$ if and only if

$$a^2 \geq \frac{9t^2 + 16t + 1}{36t^2 + 14t + 1/9}$$

is valid. The right side of the inequality is less than 1 for $t \geq 1$, therefore the algorithm is C-competitive.

Case 3.2 Suppose that $k = 3t + 2$ for some $t \geq 0$. In this case we can use weaker bounds in the calculation, we do not have to handle the corner cells separately. Then the ratio of the online and optimal costs is at most

$$\frac{((t+2)(3t+4) + (2t+2)(3t+3))(1+a^2)}{1 + ((3t+2)a)^2}.$$

This ratio is at most $5 \leq C$ if and only if

$$a^2 \geq \frac{9t^2 + 22t + 9}{36t^2 + 38t + 6}$$

is valid. The right side of the inequality is $3/2$ if $t = 0$ and it is less than 1 if $t \geq 1$, thus the algorithm is C competitive in this case if $a^2 \approx 2.2748$.

Case 3.3 Suppose that $k = 3t$ for some $t \geq 1$. Then again we can use the simpler bounds where all of the corner cells are fully covered. Then the ratio of the online and optimal costs is at most

$$\frac{((t+1)(3t+2) + (2t+1)(3t+1))(1+a^2)}{1 + (3ta)^2}.$$

This ratio is at most $5 \leq C$ if and only if

$$a^2 \geq \frac{9t^2 + 10t - 2}{36t^2 - 10t - 3}$$

is valid. The right side of the inequality is less than 1 if $t \geq 1$, therefore if $a^2 \approx 2.2748$ then the algorithm is C-competitive.

Case 4 Suppose that $(k + 1/3)a < r \leq (k + 2/3)a$ for a $k \geq 1$. Then again the optimal cluster can intersect at most $k + 2$ rows from the grid. If at some row it intersects $k + 2$ cells then in one of the neighboring rows

it can intersect only $k+1$ cells. Therefore all together it intersects at most $\lfloor (k+2)/3 \rfloor (k+1) + (k+2-\lfloor (k+2)/3 \rfloor)(k+2)$ cells. To analyze the ratio of the costs we distinguish the following subcases.

Case 4.1 Suppose that $k = 3t+1$ for some $t \geq 0$. Then the number of the fully covered cells which are not corner cells is at most $(t+1)(3t+2) + (2t+2)(3t+3) - 4$. If $r = (3t+4/3+y)a$ for some $0 \leq y \leq 1/3$, then we obtain that in each corner cell the cost is at most $\max\{1 + (2/3+y)^2a^2, (1+a^2)/2\}$. We conclude in the same way as in Case 2 or Case 3.1 that we can suppose $y = 0$. Then the online cost is at most $\big((t+1)(3t+2) + (2t+2)(3t+3) - 4\big)(1+a^2) + 4(1+4a^2/9)$, the optimal offline cost is $1 + ((3t+4/3)a)^2$, thus the ratio of the online and optimal costs is at most

$$\frac{\big((t+1)(3t+2) + (2t+2)(3t+3) - 4\big)(1+a^2) + 4(1+4a^2/9)}{1 + ((3t+4/3)a)^2}.$$

This ratio is at most $5 \leq C$ if and only if

$$a^2 \geq \frac{9t^2+17t+3}{36t^2+23t+28/9}$$

is valid. The right side of the inequality is less than 1 if $t = 0$ and also if $t \geq 1$, therefore the algorithm is C-competitive in this case.

Case 4.2 Suppose that $k = 3t+2$ for some $t \geq 0$. Then the ratio of the online and optimal costs is at most

$$\frac{((t+1)(3t+3) + (2t+3)(3t+4))(1+a^2)}{1 + ((3t+7/3)a)^2}.$$

The upper bound of this ratio is 7 if and only if

$$a^2 \geq \frac{9t^2+23t+10}{36t^2+47t+110/9}$$

is valid. The right side of the inequality is $9/11 \leq 1$ if $t = 0$ and it is also

less than 1 if $t \geq 1$, thus we conclude that the algorithm is C competitive in this case.

Case 4.3 Suppose that $k = 3t$ for some $t \geq 1$. Then the ratio of the online and optimal costs is at most

$$\frac{(t(3t+1)+(2t+2)(3t+2))(1+a^2)}{1+((3t+1/3)a)^2}.$$

This ratio is at most $5 < C$ if and only if

$$a^2 \geq \frac{9t^2+11t-3}{36t^2-t-31/9}$$

is valid. The right side of the inequality is less than 1 if $t \geq 1$, therefore if $a^2 \approx 2.2748$ then the algorithm is C-competitive.

Case 5 Suppose that $(k+2/3)a < r \leq (k+1)a$ for some $k \geq 1$. Then again the optimal cluster can intersect at most $k+2$ rows from the grid and in each row it can intersect at most $k+2$ cells. Therefore the cost of the algorithm is at most $(k+2)^2(1+a^2)$ the optimal cost is at least $1+((k+2/3)a)^2$, thus the competitive ratio is at most

$$\frac{(k+2)^2(1+a^2)}{1+((k+2/3)a)^2}.$$

We obtain that this ratio is at most $5 \leq C$ if and only if

$$a^2 \geq \frac{k^2+4k-1}{6k^2+8/3k-16/9}$$

is valid. This inequality holds if $k \geq 1$.

We conclude that in all cases the competitive ratio of the online algorithm is C.

□

3.3.2 Lower bound

Theorem 11 *The competitive ratio of any online algorithm for the flexible model is at least* $C = (x+4)/(x+1) \approx 1.743$*, where* $x \approx 3.0351$ *is a root of the equation* $4x^3 + 4x^2 - 48x - 3 = 0$.

Proof. Suppose that there exists an online algorithm with less competitive ratio than C, denote it by A. Let $a = \sqrt{x} \approx 1.742$. Consider the following input sequence. The first two points are $p_1 = (0,0)$ and $p_2 = (a,0)$. If A uses only one cluster its cost is $1 + x$, the optimal cost is 2 and the algorithm is not C-competitive. Therefore we can suppose that it uses two clusters for these points. Now a new request arrives at $p_3 = (0,a)$. Then the algorithm has the following possibilities to cover this point.

1. Suppose that A extends one of the clusters to cover this point. Then this extended cluster has a side at least a. We can suppose that the cluster which contained p_1 is extended, the other case can be handled in the same way. Then we request a very dense subset from the square which is defined by the vertices $(0,0)$, $(0,-1/(2a))$, $(-1/(2a),-1/(2a))$, $(-1/(2a),0)$, the algorithm has to cover the full square – see Figure 8. If A covers this square with a new cluster then it has cost $3 + a^2 + 1/(2a)^2$. If it extends the cluster to cover these points as well then it has a cost of $2+(a+1/(2a))^2 = 3+a^2+1/(2a)^2$. The optimal algorithm can use 3 clusters to cover all points with sizes 0 (the point $(a,0)$), 0 (the point $(0,a)$) and $1/(2a)$ (all the other points), therefore the competitive ratio in this case is at least

$$\frac{3 + a^2 + 1/(2a)^2}{3 + 1/(2a)^2} \approx 1.985 > C.$$

2. Suppose that A defines a new cluster for p_3. Then a further request arrives at $p_4 = (a,a)$. The algorithm has the following possibilities

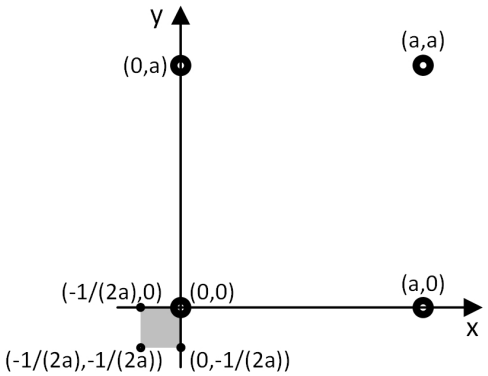

Figure 8: The request points in case 1.

to cover this point.

(a) Suppose that A defines a fourth cluster to cover this point. Then we request a very dense subset from the square defined by the vertices $(0,0)$, $(0,a)$, (a,a), $(a,0)$. The algorithm has to cover all of these points and it cannot do better then covering the full square thus it will use a total area of a^2 and its cost is at least $4+a^2$. The optimal cost is at most $1+a^2$, thus the competitive ratio is at least $(4+a^2)/(1+a^2) = C$.

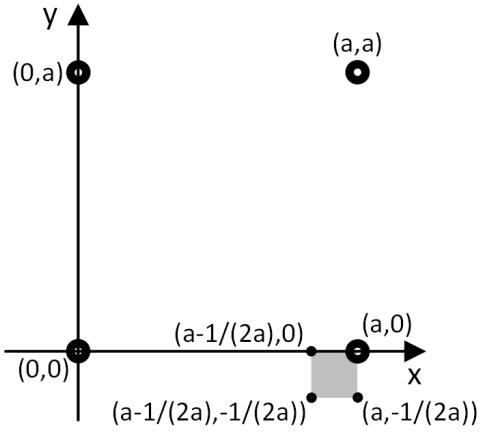

Figure 9: The request points in case 2/b

(b) Suppose that A extends one of the clusters to cover this point,

then the extended cluster has a side of least a. We can suppose that the cluster which contained p_2 is extended, the other cases can be handled in the same way. Then we request a very dense subset from the square which is defined by the vertices $(a, 0)$, $(a, -1/(2a))$, $(a - 1/(2a), -1/(2a))$, $(a - 1/(2a), 0)$. A has to cover the full square – see Figure 9. If it covers the square with a new cluster then it has cost $4 + a^2 + 1/(2a)^2$, if it extends the cluster to cover these points as well then it has a cost of $3 + (a + 1/(2a))^2 = 4 + a^2 + 1/(2a)^2$. The optimal algorithm can use 4 clusters to cover all points with sizes 0 ((the point $(0, 0)$)), 0 ((the point $(0, a)$), 0 (the point (a, a)) and $1/(2a)$ (all of the other points) respectively, therefore the competitive ratio in this case is at least

$$\frac{4 + a^2 + 1/(2a)^2}{4 + 1/(2a)^2} = C.$$

The last equality follows from $4a^6 + 4a^4 - 48a^2 - 3 = 0$.

We obtained that the competitive ratio is at least C in each case, therefore the result follows.

□

3.4 Experimental tests of the grid parameter

In the competitive analysis it is obtained that algorithm SHIFT(1/3)GRID$_a$ achieves its best competitive ratio for an interval of the parameter a. We studied how the average behavior of the algorithm depends on this parameter. Random request sequences of different length are generated with uniform distribution in a square of size 30×30. Both SHIFT(1/3)GRID$_a$ and SHIFT(1/3)FGRID$_a$ are investigated on the same sequences with the parameter a in the interval where the competitive

ratio of SHIFT(1/3)GRID$_a$ is minimal and also with slightly smaller and larger parameter values. We divided the optimal parameter interval into equal parts and this gave us the parameter values in the tests. Each test is performed 10 times, the average values are shown in the tables 1 and 2. In the top line of the tables the value of the parameter a is presented and the first column gives the number of request points. Bold is used in each line to denote the best objective function value. We note that the tables with 10 columns are broken into two parts, each part containing 5 columns.

#points, a	0.6702	0.7071	0.7439	0.7808	0.8176
50	**71.0144**	73.8000	76.2738	79.0309	81.2537
100	**141.0144**	146.8500	150.3729	156.6131	162.0068
200	**274.2026**	282.7500	289.4056	299.8666	310.3324
300	**403.0431**	415.5000	426.7296	434.7502	449.1477
400	**526.6661**	536.2500	547.8978	562.2297	575.9502
500	**639.5646**	654.7500	665.3378	682.949	701.9184
1000	**1141.8829**	1155.0000	1166.011	1172.9081	1187.7721
5000	2690.4321	2582.8500	2456.9186	2347.5868	2228.5535
10000	2934.4899	2761.3500	2593.3105	2447.0595	2301.7986

#points, a	0.8544	0.8912	0.9281	0.9649	1.0017
50	83.5604	88.1011	91.7631	94.0414	97.7689
100	166.0829	173.1518	179.0590	185.1862	192.5326
200	316.9415	331.5901	339.3187	352.4138	361.6247
300	458.1119	478.0066	487.8521	501.4897	517.8947
400	588.5562	611.3245	617.5861	634.9241	649.9228
500	710.1773	736.7473	743.5975	755.6138	774.7384
1000	1189.5686	1205.4234	1209.2999	1209.2138	1213.6967
5000	2149.2163	2056.2895	1986.7734	1913.0759	**1826.3552**
10000	2205.7882	2091.4581	2022.8830	1943.7793	**1841.5815**

Table 1: The costs of $Shift(1/3)GRID_a$

Based on the above results the following set of conclusions are drawn:

- Observe that the best value of the parameter in the case of

#points, a	0.6702	0.7071	0.7439	0.7808	0.8176
50	49.1747	49.3102	49.2376	49.2296	48.9567
100	97.6479	98.2301	97.3883	97.7089	97.6102
200	190.2726	189.8077	188.2612	188.2476	188.2599
300	280.4192	279.8541	278.4045	274.5088	274.1086
400	367.4212	362.6804	359.8828	356.7944	354.5127
500	448.1340	444.4284	438.8660	435.8802	434.0661
1000	811.1618	798.1739	784.5582	767.6279	759.0505
5000	2124.0052	2020.4392	1912.2135	1819.7368	1730.2603
10000	2520.1382	2368.5762	2227.6816	2107.8043	1992.9301
#points, a	0.8544	0.8912	0.9281	0.9649	1.0017
50	**48.6660**	49.3952	49.4539	49.0958	49.0646
100	**96.7520**	97.3106	97.1682	96.9823	96.9636
200	185.8145	187.4794	186.1791	186.6144	**185.2483**
300	271.0262	272.8910	270.7076	269.1366	**268.6732**
400	350.6717	352.2294	346.4836	345.1840	**343.0134**
500	426.3726	428.0001	421.4842	416.3530	**414.7936**
1000	741.8113	734.4657	721.7623	707.1137	**696.8139**
5000	1665.6461	1598.6024	1548.2784	1496.4183	**1442.8930**
10000	1911.6143	1823.1600	1765.3234	1697.4163	**1629.3094**

Table 2: The costs of $Shift(1/3)FGRID_a$

SHIFT(1/3)GRID$_a$depends on the number of points which is related to their density. As the density is increasing it is better to use larger grid cells. The straightforward idea for its reason is that when the density is small then there are many large cells, which contain only a few demand points, and using smaller cells would be more effective. When there are many points then most of the cells are used, and this means that almost the full area is cost, but using larger cells there are fewer cells and smaller setup cost is paid. In the case of $Shift(1/3)FGRID_a$ there is still a dependence between the density and the better parameter values but it is not so strong. The reason could be that $Shift(1/3)FGRID_a$ does not use the whole

cells. This dependence on the density can be very useful if an a priori information about the points is known.

- In most cases the best value of the average objective function is outside the interval where the optimal values are from the competitive analysis. The reason of this "anomaly" could be that the worst cases in the competitive analysis have very special structure and they do not occur in randomly generated inputs.

- As it is expected $FGRID_a$ also has significantly smaller cost in the average case than $GRID_a$ not only its competitive ratio is less.

3.5 Extensions: d-dimensional space and general power instead of square of the side of a cluster

3.5.1 Introduction

In this section a more general version of the variable sized clustering problem is considered: d-dimensional Euclidean spaces and a more general cost function are studied. In our model the cost of a cluster is a unit setup cost plus the p-th power of the side of the cube. We examine the strict version of the problem where the exact location of the cluster is fixed when it is created.

The simplest grid based algorithm is $GRID_a$ which can be described in d-dimensional Euclidean spaces as follows. It covers the space with a hypercube-grid of a^d sized cells. The clusters will be the closed cells of this grid. If a new request is not covered by the already defined clusters then the algorithm creates a new cluster for it which is the closed cell containing the request. Such grid based algorithm is studied for 1-dimensional unit clustering problems in [10] and [25].

In 2-dimensional spaces a better algorithm called SHIFT$(1/3)$GRID$_a$ is defined and described in detail in the section 3.2.1.

We analyze the algorithm $GRID_a$ for the general case and we prove that it is not constant competitive if $p < d$ and 3^d-competitive if $p \geq d$. In two dimensional Euclidean space we investigate the more difficult SHIFT$(1/3)$GRID$_a$ algorithm and we prove that it is 7-competitive for an adequate choice of the parameter a.

3.5.2 Multidimensional cases

First we consider the general d-dimensional case. The best competitive ratio of $GRID_a$ depends on the ratio of p and d and it is determined in the two theorems below.

Theorem 12 *If $p < d$ then the algorithm $GRID_a$ is not constant competitive.*

Proof. Let n be a large integer and consider an input which contains the points with coordinates $((i_1 + 0.5)a, (i_2 + 0.5)a, \ldots, (i_d + 0.5)a)$, for each integer $0 \leq i_j \leq n-1$, $j \in 1, \ldots, d$. Then the algorithm $GRID_a$ uses n^d clusters and its cost is $n^d(1 + a^p)$. On the other hand one can cover all requests with one cluster of side $(n-1)a$ therefore the optimal cost is at most $1 + ((n-1)a)^p \leq 1 + n^p a^p$. In the case of $p < d$ the ratio of the costs of the algorithm $GRID_a$ and the optimal solution tends to ∞ as n grows. □

Theorem 13 *If $p \geq d$ there exists a parameter* a *such that $GRID_a$ is 3^d-competitive. Moreover, $GRID_a$ has never smaller competitive ratio than 3^d.*

Proof. First suppose that the input contains 3^d points. The points are the d-dimensional vectors which can be constructed from the points of the multiset $\{-\varepsilon, a/2, a + \varepsilon\}$ where ε is a very small positive number. Then $GRID_a$ uses a new cell to cover each request point therefore its cost is

$3^d(1+a^p)$. On the other hand these points can be covered by one cube with sides $a+2\varepsilon$ therefore the optimal cost is at most $1+(a+2\varepsilon)^p$. The ratio tends to 3^d as ε tends to 0 thus the competitive ratio of $GRID_a$ cannot be smaller than 3^d.

Now let $a = \sqrt[p]{(3/2)^d - 1)}$, we will prove that in this case $GRID_a$ is 3^d-competitive. Consider an arbitrary input and its optimal solution. We analyze the optimal clusters separately. Suppose that we have a cluster which is a hypercube with sides r. Distinguish the following two cases depending on r.

First suppose that $r \leq a$. Then this cluster cannot intersect more than 2^d cells. As a result the cost of the algorithm $GRID_a$ on this cluster is at most $2^d(1+a^p)$, the optimal cost is at least 1 thus their ratio is at most $2^d(1+a^p) = 3^d$.

Now suppose that $r > a$. Then $k \cdot a < r \leq (k+1) \cdot a$ for an integer $k \geq 1$. This yields that this optimal cluster intersects at most $(k+2)^d$ cells from the grid. Therefore, considering only the requests of this optimal cluster, $GRID_a$ has at most $(1+a^p)(k+2)^d$ cost. The optimal cost is at least $1+k^p a^p$. Thus to prove the competitive ratio we have to show that

$$\frac{(1+a^p)(k+2)^d}{1+k^p a^p} \leq 3^d.$$

If $k = 1$ then equality holds between the two sides, thus the statement is valid. If $k > 1$ then the inequality is equivalent to

$$(k+2)^d - 3^d \leq a^p(3^d k^p - (k+2)^d).$$

Since $p \geq d$, it is enough to prove that

$$(k+2)^d - 3^d \leq a^p((3k)^d - (k+2)^d).$$

Substituting $(3/2)^d - 1$ into a^p we obtain that equality holds if $d = 1$ and a simple calculation shows that this inequality is valid for $d > 1$ and $k > 1$.

□

Remark In smaller dimensions a more careful calculation can determine the interval of the parameter a where $GRID_a$ is 3^d-competitive. In 1-dimensional Euclidean space the algorithm is 3-competitive if

$$\frac{1}{\sqrt[p]{3 \cdot 2^p - 4}} \leq a \leq \frac{1}{\sqrt[p]{2}}.$$

In 2-dimensional Euclidean space the algorithm is 9-competitive if

$$\sqrt[p]{\frac{7}{9 \cdot 2^p - 16}} \leq a \leq \sqrt[p]{5/4}.$$

3.5.3 Two dimensional version

In this section the algorithm SHIFT$(1/3)$GRID$_a$ is analyzed. We prove the following theorem.

Theorem 14 *Algorithm* SHIFT$(1/3)$GRID$_a$ *is 7-competitive if*

$$\max\left\{\sqrt[p]{\frac{1}{2^p - 2}}, \sqrt[p]{\frac{1}{7 \cdot (4/3)^p - 8}}\right\} \leq a \leq \sqrt[p]{\frac{27}{29}}.$$

$\sqrt[p]{\frac{1}{2^p-2}}$ *is greater for* $2 < p \leq x$ *and* $\sqrt[p]{\frac{1}{7\cdot(4/3)^p-8}}$ *is greater for* $p > x$ *where* $x \approx 4.0257$ *is the root of the equation:*

$$\frac{1}{2^p - 2} = \frac{1}{7 \cdot (4/3)^p - 8}$$

.

Proof. Consider an arbitrary input and an optimal solution for it. Again we analyze the optimal clusters separately. Let r denote the length of the side of a cluster. We distinguish the following cases depending on r.

Case 1 Suppose that $r \leq a/3$. Then this cluster can intersect at most two rows from the grid. If it intersects two cells in one of these rows then it can intersect only one in the other, thus in this case the optimal cluster can not intersect more than 3 cells from the grid. Therefore the cost of the online cost on this cluster is at most $3(1+a^p)$. The optimal cost is at least 1, thus we obtain that the competitive ratio is at most $3(1+a^p)$ which is at most 7 if $a \leq \sqrt[p]{4/3}$. Since $\sqrt[p]{27/29} < \sqrt[p]{4/3}$ we obtain that the statement of the theorem holds in this case.

Case 2 Suppose that $a/3 < r \leq a$. Then the cluster cannot intersect more than 4 cells. Therefore the online cost is at most $4(1+a^p)$. From $a/3 < r$ it follows that the optimal cost is at least $1 + a^p/9$, thus in this case the competitive ratio is at most $4(1+a^p)/(1+a^p/9)$ which is at most 7 if $a \leq \sqrt[p]{27/29}$.

Case 3 Suppose that $ka \leq r \leq (k+1/3)a$ for a $k \geq 1$. Then the optimal cluster can intersect at most $k+2$ rows from the grid. If at some row it intersects $k+2$ cells then in the neighboring rows it can intersect only $k+1$ cells. Therefore the total number of the cells the optimal cluster might intersect is at most $\lceil (k+2)/3 \rceil (k+2) + (k+2-\lceil (k+2)/3 \rceil)(k+1)$. Thus we obtain that the optimal online cost is at most $\big(\lceil (k+2)/3 \rceil (k+2) + (k+2-\lceil (k+2)/3 \rceil)(k+1)\big)(1+a^p)$. On the other hand the optimal cost is at least $1 + (ka)^p$. Now distinguish the following subcases.

Case 3.1 Suppose that $k = 3t+1$ for some $t \geq 0$. By replacing k with $3t+1$ we obtain that the bound on the competitive ratio is at most

$$\frac{(9t^2+16t+7)(1+a^p)}{1+(3t+1)^p a^p}.$$

If $t = 0$ then this ratio is 7. Otherwise, this ratio is smaller than 7 if and only if

$$a^p \geq \frac{9t^2 + 16t}{7 \cdot (3t+1)^p - (9t^2 + 16t + 7)}$$

is valid. The right side of the inequality is maximal on $t \geq 1$ if $t = 1$. Therefore if $a \geq \sqrt[p]{\frac{25}{7 \cdot 4^p - 32}}$ then the inequality holds. This expression is less than $\sqrt[p]{1/(2^p - 2)}$ for every $2 < p \leq x$ and is less than $\sqrt[p]{1/(7 \cdot (4/3)^p - 8)}$ for every $p > x$. Thus we obtained that the algorithm is 7-competitive in this case.

Case 3.2 Suppose that $k = 3t + 2$ for some $t \geq 0$. Substituting k with $3t + 2$ we achieve that the bound on the competitive ratio is at most

$$\frac{(9t^2 + 22t + 14)(1 + a^p)}{1 + (3t+2)^p a^p}.$$

This ratio is at most 7 if and only if

$$a^p \geq \frac{9t^2 + 22t + 7}{7 \cdot (3t+2)^p - (9t^2 + 22t + 14)}$$

is valid. The right side of the inequality is $1/(2^p - 2)$ if $t = 0$ and it is smaller if $t \geq 1$. The function $1/(2^p - 2)$ is less than the function $1/(7 \cdot (4/3)^p - 8)$ if $p > x$, thus we obtain that the algorithm is 7 competitive in this case.

Case 3.3 Suppose that $k = 3t$ for some $t \geq 1$. Using $3t$ instead of k we obtain that the bound on the competitive ratio is at most

$$\frac{(9t^2 + 10t + 3)(1 + a^p)}{1 + (3t)^p a^p}.$$

This ratio is at most 7 if and only if

$$a^p \geq \frac{9t^2 + 10t - 4}{7 \cdot (3t)^p - (9t^2 + 10t + 3)}$$

is valid. The right side of the inequality is maximal on $t \geq 1$ if $t = 1$ where it is

$$\frac{15}{7 \cdot 3^p - 22}$$

which is smaller than the function $1/(2^p - 2)$ if $2 < p \leq x$ also than $1/(7 \cdot (4/3)^p - 8)$ for every $p > x$, therefore we conclude that the algorithm is 7-competitive in this case.

Case 4 Suppose now that $(k + 1/3)a < r \leq (k + 2/3)a$ for a $k \geq 1$. Then again the optimal cluster can intersect at most $k + 2$ rows from the grid. If at some row it intersects $k + 2$ cells then in one of the neighboring rows it can intersect only $k + 1$ cells. Therefore the total number of the cells the optimal cluster might intersect is at most $\lfloor (k+2)/3 \rfloor (k+1) + (k+2 - \lfloor (k+2)/3 \rfloor)(k+2)$. This yields that the online cost is at most $\big(\lfloor (k+2)/3 \rfloor (k+1) + (k+2 - \lfloor (k+2)/3 \rfloor)(k+2)\big)(1 + a^p)$. On the other hand the optimal cost is at least $1 + ((k + 1/3)a)^p$. We distinguish the following subcases.

Case 4.1 Suppose that $k = 3t + 1$ for some $t \geq 0$. By replacing k with $3t + 1$ we obtain that the bound on the competitive ratio is at most

$$\frac{(9t^2 + 17t + 8)(1 + a^p)}{1 + (3t + 4/3)^p a^p}.$$

We obtain that this ratio is at most 7 if and only if

$$a^p \geq \frac{9t^2 + 17t + 1}{7 \cdot (3t + 4/3)^p - (9t^2 + 17t + 8)}$$

is valid. If $t = 0$ the right side of the inequality is:

$$\frac{1}{7 \cdot (4/3)^p - 8}$$

which is less than $1/(2^p - 2)$ if $2 < p \leq x$. If $t = 1$ we obtain

$$\frac{27}{7 \cdot (13/3)^p - 34}$$

which is also less than $1/(2^p - 2)$ for $2 < p \leq x$ and less than $1/(7 \cdot (4/3)^p - 8$ for every $p > 2$. If $t = 2, 3, ...$ the functions are even smaller so the theorem holds in this case, too.

Case 4.2 Suppose that $k = 3t + 2$ for some $t \geq 0$. Substituting k with $3t + 2$ the bound on the competitive ratio is at most

$$\frac{(9t^2 + 23t + 15)(1 + a^p)}{1 + (3t + 7/3)^p a^p}.$$

We obtain that this ratio is at most 7 if and only if

$$a^p \geq \frac{9t^2 + 23t + 8}{7 \cdot (3t + 7/3)^p - (9t^2 + 23t + 15)}$$

is valid.

The right side of the inequality is

$$\frac{8}{7 \cdot (7/3)^p - 15} \leq 1/(2^p - 2)$$

if $t = 0$ and it is also smaller if $t \geq 1$, thus we obtain that the algorithm is 7 competitive in this case.

Case 4.3 Suppose that $k = 3t$ for some $t \geq 1$. Using $3t + 2$ instead of k we obtain that the bound on the competitive ratio is at most

$$\frac{(9t^2 + 11t + 4)(1 + a^p)}{1 + (3t + 1/3)^p a^p}.$$

We obtain that this ratio is at most 7 if and only if

$$a^p \geq \frac{9t^2 + 11t - 3}{7 \cdot (3t + 1/3)^p - (9t^2 + 11t + 4)}$$

is valid. If $t = 1$

$$a^2 \geq \frac{17}{7 \cdot (10/3)^p - 24}$$

This expression is less than the function $1/(2^p - 2)$ if $2 < p \leq x$ and also than $1/(7 \cdot (4/3)^p - 8)$ for every $p > x$. For $t > 1$ the expression is even less than for $t = 1$. Therefore we obtain that the algorithm is 7-competitive.

Case 5 Suppose that $(k+2/3)a < r \leq (k+1)a$ for some $k \geq 1$. Then again the optimal cluster can intersect at most $k+2$ rows from the grid and in each row it can intersect at most $k+2$ cells. Therefore the online cost is at most $(k+2)^2(1+a^p)$. On the other hand the optimal cost is at least $1+((k+2/3)a)^p$, thus the competitive ratio is at most

$$\frac{(k+2)^2(1+a^p)}{1+((k+2/3)a)^p}.$$

We obtain that this ratio is at most 7 if and only if

$$a^p \geq \frac{k^2+4k-3}{7\cdot(k+2/3)^p-(k^2+4k+4)}$$

is valid. For all of the values of k the above expression is smaller than $1/(2^p-2)$ if $2 < p \leq x$ and also than $1/(7\cdot(4/3)^p-8)$ if $p > x$.

Since we considered all the possible values for r, thus we proved that the algorithm is 7-competitive. □

Remark It is easy to see that the above analysis is tight in the sense that SHIFT$(1/3)$GRID$_a$ is not better than 7-competitive. The same example which was used to prove the tightness in [19] in case of $p = 2$ also works for arbitrary p.

3.5.4 Summary and further questions

We studied the online variable sized clustering problem in d-dimensional Euclidean spaces where the cost of a cluster is the sum of a unit setup cost and the p-th power of the side of the cluster. We analyzed grid based algorithms and proved that these algorithms are constant competitive only when $p \geq d$.

An important open question relating to our model is whether a more sophisticated algorithm can be constant competitive for $p < d$ or not. We conjecture that no other constant competitive algorithm exists. This conjecture is proved in [35] for $p = 1$ and $d = 2$ but it seems to be very

hard to extend the lower bound proof to the general case. A further interesting question could be to investigate the flexible model where the algorithm is allowed to change the size and location of the cluster with the cost function defined in this section.

4 Online facility location

4.1 Notations and the OFW algorithm

In the facility location problem a metric space $\mathbf{M} = (M, d)$ is given, M is the set of points. The distance function d is non-negative, symmetric, and satisfies the triangle inequality. The input is a sequence $s_1, \ldots, s_n$ of requests, each request is a point of M. To serve the requests facilities should be opened at the points of the set M. We restrict our attention to the special case of uniform facility cost, where the cost of opening a facility, denoted by f, is the same for all points. If a solution SOL opens facilities at the points $a_1, \ldots, a_k$ the total cost of this solution is

$$c(SOL) = k \cdot f + \sum_{i=1}^{n} \min_{j=1,\ldots,k} d(s_i, a_j).$$

The goal is to find the solution which minimizes this cost. The first part $(k \cdot f)$, which is the cost of opening the facilities is called the facility opening cost, the second part is called the service cost. If the value of k is fixed then the problem is to minimize the service cost. This problem is called the k-median problem.

In the online facility location problem the requests points arrive one by one. After the arrival of a request point the algorithm is allowed to open a new facility and to move the opened facilities into new positions in the metric space. The algorithm has to make these decisions without any information about the further parts of the input. For the request sequence $I = s_1, \ldots, s_n$, we denote the prefix $s_1, \ldots, s_i$ by I_i. For any algorithm A let $C_A(I)$ denote the total cost, $S_A(I)$ the service cost and $F_A(I)$ the facility opening cost of the solution achieved on the input I. Fix an optimal offline algorithm and let $C_{OPT}(I)$ denote the total cost, $S_{OPT}(I)$ the service cost and $F_{OPT}(I)$ the facility opening cost of the

optimal offline solution obtained on the input sequence I. Note that the number of the opened facilities for algorithm A on input I is $F_A(I)/f$.

We propose the algorithm OFW (Optfollow) for the solution of the online problem. The basic idea is to mimic the behavior of the optimal offline algorithm. OFW uses the following rules to build the online solution after the arrival of s_i.

Algorithm 6 Algorithm OFW

- Step 1. Determine an optimal offline solution for the input I_i.
- Step 2/a. If $F_{OFW}(I_{i-1}) \leq F_{OPT}(I_i)$ then open $(F_{OPT}(I_i) - F_{OFW}(I_{i-1}))/f$ new facilities, and move the $F_{OPT}(I_i)/f$ facilities into the optimal positions.
- Step 2/b. If $F_{OFW}(I_{i-1}) > F_{OPT}(I_i)$ then do not open new facilities, solve the offline $F_{OFW}(I_{i-1})/f$-median problem on I_i and move the facilities into the resulting positions.

Remark: Note that in the case of general metric spaces OFW solves an NP-hard problem in Step 1, also in Step 2/b. These NP-hard problems are well studied and several exact solution algorithms are developed for them (see [14], [27], [37] for details). On the other hand, if the metric space is the line then both problems can be solved in polynomial time. Most of these polynomial algorithms are based on dynamic programming, the fastest algorithm has running time $O(nk)$ for the k-median problem (see [29]), and has running time $O(n^2)$ for the facility location problem (see [41]).

The following polynomial version of OFW called $POFW$ can be defined using approximation algorithms as the solutions of the NP-hard problems.

Remark: Several approximation algorithms have been developed for the facility location and k-median problems, see [52] and [54] for surveys

Algorithm 7 Algorithm *POFW*

- Step 1. Use a polynomial time approximation offline algorithm $FAPPR$ on the input I_i in the facility location problem.
- Step 2/a. If $F_{POFW}(I_{i-1}) \leq F_{APPR}(I_i)$ then open $(F_{FAPPR}(I_i) - F_{POFW}(I_{i-1}))/f$ new facilities, and move the $F_{FAPPR}(I_i)/f$ facilities into the positions given by $FAPPR$.
- Step 2/b. If $F_{POFW}(I_{i-1}) > F_{FAPPR}(I_i)$ then do not open new facilities, use a polynomial time approximation offline algorithm $MAPPR$ for the $F_{POFW}(I_{i-1})/f$-median problem on I_i and move the facilities into the received positions.

on these areas.

4.2 Competitive analysis

In this section algorithms OFW and $POFW$ are analyzed. For general metric spaces the following result is obtained.

Theorem 15 *Algorithm OFW is 2-competitive.*

Proof. Consider an arbitrary input sequence I_n. Investigate the ratio $C_{OFW}(I_n)/C_{OPT}(I_n)$. If OFW uses Step 2/a after the arrival of s_n, then this ratio is 1, and the statement of the theorem holds. Suppose that OFW uses Step 2/b after the arrival of s_n. Then $C_{OFW}(I_n) = S_{OFW}(I_n) + F_{OFW}(I_n)$ and from $F_{OFW}(I_n) = F_{OFW}(I_{n-1}) > F_{OPT}(I_n)$ we obtain that $S_{OFW}(I_n) \leq S_{OPT}(I_n) \leq C_{OPT}(I_n)$ (OFW can use more facilities to serve the requests).

On the other hand, let $r < n$ denote the request point where OFW opened its last facility. Then $F_{OFW}(I_n) = F_{OFW}(I_r) \leq C_{OPT}(I_r)$. Moreover, the optimal cost cannot decrease as new requests appear thus $C_{OPT}(I_r) \leq C_{OPT}(I_n)$ and this yields $F_{OFW}(I_n) \leq C_{OPT}(I_n)$. Hence we conclude $C_{OFW}(I_n) = S_{OFW}(I_n) + F_{OFW}(I_n) \leq 2 \cdot C_{OPT}(I_n)$. □

Theorem 16 *If $FAPPR$ and $MAPPR$ are c_1 and c_2 approximation algorithms respectively, then the algorithm $POFW$ is $c_1(1+c_2)$-competitive.*

Proof. Consider an arbitrary input sequence I_n. Analyze the ratio $C_{POFW}(I_n)/C_{OPT}(I_n)$. If OFW uses Step 2/a after the arrival of s_n, then this ratio is c_1, and the statement of the theorem holds. Suppose that $POFW$ uses Step 2/b after the arrival of s_n. Then $C_{POFW}(I_n) = S_{POFW}(I_n) + F_{POFW}(I_n)$ and from

$$F_{POFW}(I_n) = F_{POFW}(I_{n-1}) > F_{FAPPR}(I_n)$$

we achieve:

$$S_{POFW}(I_n) \leq c_2 \cdot S_{FAPPR}(I_n) \leq c_2 \cdot C_{FAPPR}(I_n) \leq c_2 \cdot c_1 \cdot C_{OPT}(I_n)$$

($POFW$ can use more facilities to serve the requests and it uses the c_2-approximation k-median algorithm MAPPR).

On the other hand, let $r < n$ denote the request where $POFW$ opened its last facility. Then

$$F_{POFW}(I_n) = F_{POFW}(I_r) \leq C_{FAPPR}(I_r) \leq c_1 \cdot C_{OPT}(I_r).$$

As new requests appear the optimal cost cannot decrease, thus $C_{OPT}(I_r) \leq C_{OPT}(I_n)$ and this results: $F_{POFW}(I_n) \leq c_1 \cdot C_{OPT}(I_n)$. The result follows: $C_{POFW}(I_n) = S_{POFW}(I_n) + F_{POFW}(I_n) \leq c_1(1 + c_2) \cdot C_{OPT}(I_n)$. □

A stronger result can be stated for a more special metric space. Suppose that the metric space is the line where the points are real numbers and $d(x, y) = |x - y|$. To analyze OFW on the line the following property of the optimal solutions is needed.

Lemma 2 *If the metric space is the line then for an optimal solution which uses the least number of facilities and for any input I_n and indices $1 < i < j \leq n$ the following inequality is valid: $F_{OPT}(I_i) \leq 2 \cdot F_{OPT}(I_j)$.*

Proof. Consider an arbitrary input sequence I_n, and fix an index $i < n$. Suppose that $F_{OPT}(I_i) = p$, consider an optimal solution OPT_i of I_i which uses the least number of facilities and let $a_1 < a_2 < \cdots < a_p$ denote the points where the facilities are placed and let A_i denote the set of the requests which are assigned to a_i $(i = 1, \ldots, p)$. Consider a $j > i$ and an optimal solution OPT_j of I_j which also uses the least number of facilities and let $b_1 < b_2 < \cdots < b_q$ denote the points where the facilities are placed. We prove by induction that $b_k < a_{2k}$ holds for each k.

Let $k = 1$ and assume by contradiction that $b_1 \geq a_2$. Consider OPT'_j, where the facilities are placed to the points $a_1, b_1, \ldots, b_q$. The elements of A_1 are assigned to a_1 in OPT_i therefore their positions are at least as close to a_1 as to a_2 and thus less than or equal to a_2. This yields that these elements are assigned to b_1 in OPT_j, furthermore they are assigned to a_1 in OPT'_j. Therefore, the service cost of the elements of A_1 is $\sum_{r \in A_1} d(r, b_1)$ in OPT_j and it is $\sum_{r \in A_1} d(r, a_1)$ in OPT'_j. On the other hand, these elements are as close to a_2 as to b_1 thus $\sum_{r \in A_1} d(r, b_1) \geq \sum_{r \in A_1} d(r, a_2)$. We obtained that the difference in the service cost of OPT_j and OPT'_j is at least $\sum_{r \in A_1} d(r, a_2) - \sum_{r \in A_1} d(r, a_1)$ (OPT_j has smaller service cost for the elements of A_1, and it cannot have larger service cost for the other elements), but OPT'_j uses one extra facility. This yields that

$$C(OPT_j) - C(OPT'_j) > \sum_{r \in A_1} d(r, a_2) - \sum_{r \in A_1} d(r, a_1) - f.$$

On the other hand, if we consider the solution OPT'_i for the input I_i which assigns facilities to the points $a_2, \ldots, a_p$, then $C(OPT'_i) = C(OPT_i) + \sum_{r \in A_1} d(r, a_2) - \sum_{r \in A_1} d(r, a_1) - f$. Moreover OPT'_i contains less facilities than OPT_i and it is supposed that OPT_i is the optimal solu-

tion which has the smallest number of facilities, therefore OPT_i' cannot be an optimal solution. This yields that $\sum_{r\in A_1} d(r,a_2) - \sum_{r\in A_1} d(r,a_1) - f > 0$, which yields the contradiction $C(OPT_j') < C(OPT_j)$.

Now let $1 \le k < q$ arbitrary and suppose that $b_k < a_{2k}$. Assume by contradiction that $b_{k+1} \ge a_{2k+2}$. We can accomplish a contradiction in a similar way as in the case of $k = 1$. Let OPT_j' be the solution, where the facilities are placed at the points $b_1, \ldots, b_k, a_{2k+1}, b_{k+1}, \ldots, b_q$. The service cost of the elements of A_{2k+1} is

$$\sum_{r\in A_{2k+1}} \min\{d(r,b_k), d(r,b_{k+1})\} \ge \sum_{r\in A_{2k+1}} \min\{d(r,a_{2k}), d(r,a_{2k+2})\}$$

in OPT_j and it is $\sum_{r\in A_{2k+1}} d(r,a_{2k+1})$ in OPT_j'. Therefore

$$C(OPT_j) - C(OPT_j') \ge$$

$$\ge \sum_{r\in A_{2k+1}} \min\{d(r,a_{2k}), d(r,a_{2k+2})\} - \sum_{r\in A_{2k+1}} d(r,a_{2k+1}) - f.$$

On the other hand, if we consider the solution OPT_i' for the input I_i which assigns facilities to the points $a_1, \ldots, a_{2k}, a_{2k+2}, \ldots, a_p$, then

$$C(OPT_i') = C(OPT_i) + \sum_{r\in A_{2k+1}} \min\{d(r,a_{2k}), d(r,a_{2k+2})\} -$$

$$- \sum_{r\in A_{2k+1}} d(r,a_{2k+1}) - f.$$

Hence, by the optimality of OPT_i we obtain that

$$\sum_{r\in A_{2k+1}} \min\{d(r,a_{2k}), d(r,a_{2k+2})\} - \sum_{r\in A_{2k+1}} d(r,a_{2k+1}) - f \ge 0,$$

which leads to contradiction $C(OPT_j') < C(OPT_j)$.

Up to now we proved $b_q < a_{2q}$. Now assume $p > 2q$. Consider OPT_j', where the facilities are placed to the points $b_1, \ldots, b_q, a_{2q+1}$. The service

cost of the elements of A_{2q+1} is $\sum_{r\in A_{2q+1}} d(r, b_q) > \sum_{r\in A_{2q+1}} d(r, a_{2q})$ in OPT_j and it is $\sum_{r\in A_{2q+1}} d(r, a_{2q+1})$ in OPT'_j. Therefore

$$C(OPT_j) - C(OPT'_j) > \sum_{r\in A_{2q+1}} d(r, a_{2q}) - \sum_{r\in A_{2q+1}} d(r, a_{2q+1}) - f.$$

On the other hand, if we consider the solution OPT'_i for the input I_i which assigns facilities to the points $a_1, a_2, \ldots, a_{2q}$, then $C(OPT'_i) \leq C(OPT_i) + \sum_{r\in A_{2q+1}} d(r, a_{2q}) - \sum_{r\in A_{2q+1}} d(r, a_{2q+1}) - f$. Then OPT'_i contains less facilities than OPT_i and it is supposed that OPT_i is the optimal solution which has the smallest number of facilities, therefore OPT'_i cannot be an optimal solution. This yields that

$$\sum_{r\in A_{2q+1}} d(r, a_{2q}) - \sum_{r\in A_{2q+1}} d(r, a_{2q+1}) - f > 0,$$

which yields the contradiction $C(OPT'_j) < C(OPT_j)$.

We achieved that $p \leq 2q$, and this completes the proof of the lemma.

□

It is worth noting that for general metric spaces a similar statement does not hold, as the following example shows. Let $f = 1$ and consider the metric space which contains n points $P_1, \ldots, P_n$ with the distance function $d(P_1, P_i) = 1 - 1/(2n)$, for $i \neq 1$ and $d(P_i, P_j) = 3/2$ if $i \neq 1, j \neq 1, i \neq j$. Consider the input sequence $P_2, \ldots, P_n, P_1$. It is easy to see that the optimal solution for the prefix $P_2, \ldots, P_n$ opens a facility at each point (then its cost is $n - 1$, which is less than $1 + (n - 1)(1 - 1/(2n))$). On the other hand, the optimal solution for the sequence $P_2, \ldots, P_n, P_1$ opens only one facility at point P_1. The result follows.

With the assistance of Lemma 2 we state the following result if the metric space is the line.

Theorem 17 *The algorithm OFW where we use the optimal solution*

which uses the smallest number of facilities is $\frac{3}{2}$-competitive on the line with the Euclidean distance.

Proof. Consider an arbitrary input sequence I_n. Investigate the ratio $C_{OFW}(I_n)/C_{OPT}(I_n)$. If OFW uses Step 2/a after the arrival of s_n, then this ratio is 1, and the theorem holds. Suppose that OFW uses Step 2/b after the arrival of s_n. Then $C_{OFW}(I_n) = S_{OFW}(I_n) + F_{OFW}(I_n)$ and from $F_{OFW}(I_n) = F_{OFW}(I_{n-1}) > F_{OPT}(I_n)$ we conclude that $S_{OFW}(I_n) \leq S_{OPT}(I_n)$ (OFW can use more facilities to serve the requests).

On the other hand, let $r < n$ denote the request where OFW opened its last facility. Then $F_{OFW}(I_n) = F_{OFW}(I_r) \leq C_{OPT}(I_r)$. Moreover, the optimal service cost cannot decrease as new requests appear, thus $C_{OPT}(I_r) \leq C_{OPT}(I_n)$ and this yields $F_{OFW}(I_n) \leq C_{OPT}(I_n) = S_{OPT}(I_n) + F_{OPT}(I_n)$. Thus we obtain that

$$C_{OFW}(I_n) \leq 2 \cdot S_{OPT}(I_n) + F_{OPT}(I_n).$$

On the other hand, it follows from Lemma 2 that

$$F_{OFW}(I_n) = F_{OFW}(I_r) \leq 2 \cdot F_{OPT}(I_n),$$

and this yields

$$C_{OFW}(I_n) \leq S_{OPT}(I_n) + 2 \cdot F_{OPT}(I_n).$$

These inequalities results in

$$C_{OFW}(I_n) \leq \frac{3}{2}(S_{OPT}(I_n) + F_{OPT}(I_n)) = \frac{3}{2}C_{OFW}(I_n).$$

This completes the proof. □

The following lower bound is proved for the line but of course it is also valid for general case.

Theorem 18 *No online algorithm is C-competitive on the line for any $C < (\sqrt{13}+1)/4 \approx 1.15$.*

Proof. Consider an arbitrary online algorithm A, assume by contradictions that it is C-competitive for a $C < (\sqrt{13}+1)/4$. Investigate the following input sequence: let $s_1 = s_2 = 0$, $s_3 = s_4 = r = (\sqrt{13}-1)/4$ (note that $1/2 < r < 1$). If A opens only one facility, then its total cost is at least $2r+1$, the optimal cost is 2. In this case the sequence ends and $C_A(I_4)/C_{OPT}(I_4) = (2r+1)/2 = (\sqrt{13}+1)/4 > C$ which is a contradiction.

Therefore, we may suppose that A opens two facilities. Then the sequence ends with $s_5 = s_6 = s_7 = r/2$. The optimal solution uses only one facility at the point $r/2$, thus $C_{OPT}(I_7) = 2r+1$. On the other hand, A has already two facilities, thus its cost is at least $2+r$. Therefore, $C_A(I_7)/C_{OPT}(I_7) = (2+r)/(2r+1) = (\sqrt{13}+1)/4 > C$ which is again a contradiction.

□

4.3 Experimental analysis

The competitive analysis gives a worst case bound on the performance of the algorithms, we conjecture that OFW gives much better results in average (it always gives an optimal solution when it uses Step 2/a after the arrival of a new request). An empirical analysis is used to investigate the average behavior of algorithm OFW. The tests measure how close is the solution given by OFW to the optimal one. Moreover, in order to investigate the effect of allowing the facility movements we compared OFW to an online algorithm without server movements. We used the *Partition* algorithm without server movements which is presented in [5].

Since solving the offline facility location problem for general metric spaces is an NP-hard problem, we investigated a special metric space.

We used the $[0, 1]$ interval (for the line the facility location problem can be solved in polynomial time – see [29]). The uniform and Gaussian distributions are used to generate the request sequence, in the same way as in [5]. Two cases are considered depending on the cost of a facility (cost 0.1 and cost 1) for each distribution. We generated input sequences of size 50, 100 and 200 for each distribution and facility cost, then executed algorithms *Partition* and OFW, then determined the optimal offline solution. 10 tests are performed for each size, the average results are summarized in Table 3 and Table 4. (In the summarized data $cost(i)$ denotes the average cost on the input sequence of length i, $fac(i)$ denotes average number of facilities on the input sequence of length i.)

Table 3: Uniform distribution with facility costs 0.1 and 1

cost 0.1	cost(50)	fac(50)	cost(100)	fac(100)	cost(200)	fac(200)
Partition	4.088755	7.3	6.096005	11.6	8.488129	15.7
OFW	1.794464	10.6	2.709996	15	3.973767	21.6
OPT	1.794464	10.6	2.709798	14.9	3.973428	21.5
cost 1	cost(50)	fac(50)	cost(100)	fac(100)	cost(200)	fac(200)
Partition	10.795809	3	15.634886	3.4	24.657901	6.9
OFW	6.57363	3.7	9.374849	5	13.411049	7.1
OPT	6.57363	3.7	9.374849	5	13.411049	7.1

Table 4: Gaussian distribution with facility costs 0.1 and 1

cost 0.1	cost(50)	fac(50)	cost(100)	fac(100)	cost(200)	fac(200)
Partition	3.134154	9.1	4,463008	12.2	6.328076	18
OFW	1.538526	9.2	2.312912	13.1	3.443913	19.3
OPT	1.538526	9.2	2.312912	13.1	3.443567	19.2
cost 1	cost(50)	fac(50)	cost(100)	fac(100)	cost(200)	fac(200)
Partition	8.638324	2.9	11.798673	3	17.945507	5
OFW	5.437508	2.7	7.844956	3.9	11,349347	5.8
OPT	5.437508	2.7	7.844956	3,.	11.349105	5.7

The tests show that the algorithm OFW gives very good results in the average case, furthermore in many cases OFW gave the same solution as

the optimal offline algorithm. Note that *Partition* also gives good results in average case, its average cost is no more than 2.29 times the optimal cost.

The very good performance of the algorithm OFW arises from the fact that it gives an optimal solution in every step when it uses Step 2/a after the arrival of a request. We also investigated how often does OFW use Step 2/b. The same test cases are considered as above and the number of requests is analyzed when Step 2/b is performed. In the case of uniform distribution and facility cost 0.1 Step 2/b was used after the 5 percent of the requests, while in the case of facility cost 1 this ratio decreased to 2 percent. For Gaussian distribution with facility cost 0.1 Step 2/b was performed after 7.5 percent of the requests, in the case of facility cost 1 this ratio was 4 percent. We conclude that for a larger facility cost Step 2/b is less frequent, the reason of this may be the smaller average number of the opened facilities. The maximum of the ratio of the cost of OFW and the optimal solution was also investigated. We obtained that during the tests the ratio was never greater than 1.094.

4.4 Further problems

Considering the model investigated in the thesis several further questions appear, here we list some of them. There is a gap between the proved competitive ratio and the lower bound on the possible competitive ratio, it would be interesting to decrease this difference. The lower bound proved in this paper is valid for the line, which is a very simple metric space, so a further study may give a higher lower bound for general metric spaces.

There are some extensions of the model which have not been investigated yet. One can consider the problem with nonuniform facility cost. In this case changing the position of a facility results in changes in the

opening cost. Furthermore one can consider models where changing the position of a facility is not free, it has some cost which is smaller than opening a new facility. These costs can be constant or they can depend on the distance between the positions.

Összegzés

E disszertáció az online algoritmusok tág témakörének egy részével, a klaszterezési problémákkal foglalkozik. Bemutatjuk az utóbbi néhány évben ezen a területen végzett kutatásaink eredményeit.

Az online algoritmusok feladata az, hogy az egyenként érkező bemenet alapján hozzanak döntést a további kérési pontok ismerete nélkül. A klaszterezésnél a pontokat csoportosítani kell, klaszterhez vagy kiszolgálóhoz rendelni őket. Több célfüggvény határozható meg; ebben a munkában két fő csoportjukkal foglalkozunk: azokkal, amelyek a klaszter átmérőjétől függenek és azokkal, amelyben a költség a kiszolgálótól való távolságtól függ. Új, eddig még nem vizsgált modelleket tanulmányozunk, mindegyik modellnél megadunk egy megoldó algoritmust, és megállapítjuk a versenyképességét. Továbba alsó korlátokat határozunk meg a modellt megoldó bármely online algoritmus versenyképességére.

Az első fejezet az online algoritmusok témakörének alapfogalmait és a versenyképességi elemzés alapjait mutatja be. Itt ismertetjük a klaszterezésben eddig elért eredményeket és áttekintjük a későbbi fejezetekben tárgyalt modelleket.

A második fejezet az 1-dimenziós térben, tehát az egyenesen történő klaszterezési, átmérő négyzetétől függő problémákkal foglalkozik. Először az offline problémát vizsgáljuk, ahol a teljes bemenetet előre ismeri az algoritmus, és bemutatunk egy dinamikus programozáson alapuló optimális megoldó algoritmust.

A következő szakasz a szigorú modellt mutatja be, amelyben a klaszter létrehozásakor visszavonhatatlanul el kell dönteni a méretét és a helyét. A teljesen online és a félig online változatot is vizsgáljuk, ahol a félig online ez esetben azt jelenti, hogy a kérési pontok koordinátáit nagyság szerint

sorbarakjuk. Több algoritmus alapötlete a $GRID$ algoritmus, amelynél a d-dimenziós teret d-dimenziós egységkockákkal fedjük be. Amikor érkezik egy kérési pont, ami nincs már létező klaszterben, akkor az algoritmus új klasztert nyit, ami a zárt kocka. Az online változatra bemutatjuk a $GRID_a$ algoritmust (az a paraméter az intervallum hossza), amely elérheti a 3-versenyképességet. A félig online változatot megoldó $SOSM_a$ algoritmusra bizonyítjuk, hogy 2-versenyképes az a paramétertől függően.

Vizsgáljuk a flexibilis modellt, amelynél a már létrehozott klaszter nyújtható és eltolható, de csak annyira, hogy az eddig hozzárendelt pontokat továbbra is tartalmazza. Az online változatra a $GRID_a$ nyújtható változatát, az $FGRID_a$ algoritmust használjuk, amelyre bebizonyítjuk, hogy megfelelő paraméterezéssel 2-versenyképes.

A harmadik fejezetben az előbbi modellek 2-dimenziós változatait tárgyaljuk. A szigorú modellnél $GRID_a$ további, kifinomultabb változatát, a SHIFT$(1/3)$GRID$_a$ algoritmust vizsgáljuk részletesen, amelynél a négyzethálóban a következő sorokat az előzőkhöz képest $a/3$-mal eltoljuk. Bizonyítjuk, hogy 7-versenyképes a legjobb paraméterválasztást követően. A flexibilis modellnél az előző algoritmus nyújtható kiterjesztésére, a SHIFT$(1/3)$FGRID$_a$ algoritmusra bizonyítjuk az 5.22-versenyképességet az a paraméter megfelelő kiválasztásától függően. Tesztekkel elemezzük mindkét algoritmus hatékonyságát az a paraméter függvényében a kérési pontok egyenletes eloszlása mellett.

A fejezet utolsó szakaszában kitérünk a d-dimenziós euklideszi terekre kiterjesztett $GRID_a$ algoritmusok vizsgálatára a szigorú modellben, illetve tanulmányozzuk a célfüggvény relaxációját is: már nem négyzetes költséggel számolunk, hanem p-edik hatvánnyal. Bizonyítjuk, hogy a $GRID_a$ algoritmus nem konstans versenyképes, ha $p < d$, és 3^d-versenyképes, ha $p \geq d$. 2-dimenzióban a SHIFT$(1/3)$FGRID$_a$ algoritmusra bizonyítjuk, hogy a p hatványú költségre is 7-versenyképes.

A negyedik fejezet az online kiszolgáló-elhelyezéssel foglalkozik, ami definiálható olyan klaszterezési problémaként, amelyben a célfüggvény a kérési pontok kiszolgálótól való távolságainak az összege. E disszertációban azt az új modellt vizsgáljuk, amelyben engedélyezett a már megvett és elhelyezett kiszolgálók áthelyezése, ami nem jár plusz költséggel, de nem adhatjuk el / zárhatjuk be őket. Ismertetjük az OFW algoritmust az online probléma megoldására, amelynek az alapötlete az, hogy utánozza az optimális megoldó algoritmust. Segédalgoritmusokként valamely optimális megoldó algoritmust és valamely k-medián feladatot megoldó algoritmust használunk, amelyek általános metrikus térben NP-nehezek, ám az egyenesen léteznek polinomiális idejű algoritmusok. Bemutatásra kerül a $POFW$ algoritmus is, amely az OFW polinomiális idejű változata, és approximációs segédalgoritmusokat használ. Az OFW algoritmusra bizonyítjuk, hogy 2-versenyképes, a $POFW$-re pedig, hogy $c_1(1 + c_2)$-versenyképes, ahol c_1 és c_2 konstansok a segédalgoritmusok versenyképességi hányadosai. Az egyenesre igazoljuk, hogy az OFW algoritmus 3/2-versenyképes, és bemutatjuk, hogy nincs olyan online algoritmus, amely jobb, mint $\frac{\sqrt{13}+1}{4}$-versenyképes. Végül ismertetjük az OFW, az optimális megoldás és egy kiszolgálómozgatást meg nem engedő algoritmus összehasonlítását tartalmazó teszteredményeket.

Summary

This thesis deals with a part of the wide field of online algorithms, the clustering problems. We summarize the results of our research that has been done over the past few years.

The goal of the online algorithms is to make decisions based on the input which is arriving one-by-one without any knowledge of the further request points. In clustering problems, the points need to be grouped, assigned to a cluster or a facility. Many objective functions can be defined; in this work two main groups are considered: those which depend on the size of the cluster and those which cost depends on the distance of the demand points from the facility. New models are studied which have not been examined yet. In this thesis we present a solving algorithm for every model and determine its competitive ratio. Furthermore, we provide lower bounds for the competitive ratio of any online algorithm which solves the given model.

The first chapter introduces the basic concepts of the online algorithms and the competitive analysis. We present the related results in the field of the clustering problems and summarize the models which are considered in later chapters.

The second chapter deals with the clustering problems depending on the square of the diameter in 1-dimensional space, the line. First we examine the offline problem, where the algorithm knows the whole input in advance and offer an optimal algorithm based on dynamic programming.

The next section shows the strict model where the size and the location of the cluster has to be irrevocably decided when it is opened. We study the pure online and semi online versions, where "semi online" in our case means that the coordinates of the request points are ordered. The basic idea behind many of our algorithms is the algorithm $GRID$ which covers

the d-dimensional space with d-dimensional unit cubes. When a demand point arrives which does not belong to an existing cluster, the algorithm opens a new cluster: the closed cube of the grid. For the online version we present the algorithm $GRID_a$ (where the parameter a is the length of the interval), which can be 3-competitive. For the algorithm $SOSM_a$ which solves the semi online model we prove that it is 2-competitive depending on the parameter a.

The flexible model is also studied where the existing cluster can be expanded and shifted as long as it still contains the points already assigned to it. For the online variant we use the flexible version of the algorithm $GRID_a$ called $FGRID_a$ which is proved to be 2-competitive with the adequate parameters.

We analyze the 2-dimensional versions of the previous models in the third chapter. In the strict model we study in detail a sophisticated version of the algorithm $GRID_a$ called $\textsc{Shift}(1/3)\text{GRID}_a$ where a row is shifted right by $a/3$ compared to the previous one. Having the best choice of the parameter a this algorithm is 7-competitive. In the flexible model we prove that the algorithm $\textsc{Shift}(1/3)\text{FGRID}_a$ is 5.22-competitive depending on the adequate choice of the parameter a. Behaviors of both of the algorithms are tested on various values of the parameter a with uniformly distributed demand points.

In the last section of the chapter we study the extensions of the algorithms $GRID_a$ in d-dimensions in the strict model. Furthermore, the relaxation of the objective function is examined: now the cost is the p-th power instead of the square of the size of the cluster. We prove that the algorithm $GRID_a$ is not constant competitive if $p < d$ and it is 3^d-competitive if $p \geq d$. In 2-dimensions we prove that the algorithm $\textsc{Shift}(1/3)\text{FGRID}_a$ is 7-competitive for the cost p-th power.

The fourth chapter deals with the online facility location problem

which can be defined as a clustering problem where the objective function depends on the sum of the distances of the request points from the facility which "serves" them. In this thesis a new model is investigated where it is allowed to move the already bought and placed facilities. The move has no additional cost but the facilities cannot be sold or closed. We present the algorithm OFW for the solution of the online problem where the basic idea is to imitate the optimal algorithm. Some optimal solving algorithm and some algorithm solving the k-median problem are used as building blocks. These algorithms are NP-hard in the general metric space but on the line there exist algorithms with polynomial time. The algorithm $POFW$ is also presented which is a polynomial time variant of the algorithm OFW; it uses approximation algorithms as building blocks. We prove that the algorithm OFW is 2-competitive and the algorithm $POFW$ is $c_1(1 + c_2)$-competitive where the constants c_1 and c_2 are the competitive ratios of the auxiliary algorithms. It is shown that the algorithm OFW is 3/2-competitive on the line and that no algorithm exists which has a better competitive ratio than $\frac{\sqrt{13}+1}{4}$. Finally we present the test results which compare OFW, the optimal solution and an algorithm which does not allow facility movements.

List of Figures

List of Tables

References

[1] Ahmadian, S., Friggstad, Z., and Swamy, C. Local-search based approximation algorithms for mobile facility location problems. In *Proc. 24th Symp. on Discrete Algorithms (SODA)*, pp 1607–1621, 2013.

[2] Albers, S. Competitive online algorithms. *Optima*, 54:1-8, 1997. Feature article in the newsletter of the Mathematical Programming Society.

[3] Albers, S. Online algorithms: A survey. *Mathematical Programming*, **97**, pp. 3–26, 2003.

[4] Albers, S., and Schröder, B. An Experimental Study of Online Scheduling Algorithms. *ACM Journal of Experimental Algorithmics 7: 3*, 2002.

[5] 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)**, pp. 175–202, 2004.

[6] Arya, V., Garg, N., Khandekar, R., Meyerson,A., Munagala, K., and Pandit, V. Local search heuristics for k-median and facility location problems. *SIAM J. on Computing*, **33(3)**, pp. 544–562, 2004.

[7] 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.

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

[9] Byrka, J., and Aardal, K. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. *SIAM J. on Computing*, **39(6)**, pp. 2212–2231, 2010.

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

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

[12] Charikar, M., and Panigrahy, R. Clustering to minimize the sum of cluster diameters. *J. Comput. Syst. Sci.*, **68(2)**, pp. 417–441, 2004.

[13] Chin, F. Y. L., Ting, H. F., and Zhang, Y. Variable-size rectangle covering. In: *Proc. of the 3rd International Conference on Combinatorial Optimization and Applications (COCOA2009)*, pp. 145–154, 2009.

[14] Christofides, N., and Beasley, J.E. A tree search algorithm for the p-median problem. *European Journal of Operational Research*, **10(2)**, pp. 196–204, 1982.

[15] Csirik, J., Epstein, L., Imreh, Cs., and Levin, A. Online Clustering with Variable Sized Clusters. *Algorithmica*, **65(2)**, pp. 251–274, 2013.

[16] Demaine, E. D., Hajiaghayi, M., Mahini, H., Sayedi-Roshkhar, A. S., Oveisgharan, S., and Zadimoghaddam, M. Minimizing movement. *ACM Trans. on Alg. (TALG)*, **5(3)**, 30, 2009.

[17] Divéki, G. Online Clustering on the Line with Square Cost Variable Sized Clusters. *Acta Cybernetica*, **21(1)**, pp. 75–88, 2013.

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

[19] Divéki, G. and Imreh, Cs. An Online 2-dimensional Clustering Problem with Variable Sized Clusters. *Optimization and Engineering*, **14**, pp. 575–593, 2013.

[20] Divéki, G. and Imreh, Cs. Grid based online algorithms for clustering problems. *CINTI 2014*, accepted for publication, 2014.

[21] Doddi, S., Marathe, M., Ravi, S. S., Taylor, D. S., and Widmayer, P. Approximation algorithms for clustering to minimize the sum of diameters. *Nord. J. Comput.*, **7(3)**, pp. 185–203, 2000.

[22] Drezner, Z., and Hamacher, H. W. Facility location: applications and theory. *Springer*, 2004.

[23] Ehmsen, M. R. and Larsen, K. S. Better bounds on online unit clustering. *Theoretical Computer Science, 500*, pp. 1–24, 2013.

[24] Eisenstat, D., Mathieu, C., and Schabanel, N. Facility location in evolving metrics. *arxiv*, http://arxiv.org/abs/1403.6758

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

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

[27] Erlenkotter, D. A Dual-Based Procedure for Uncapacitated Facility Location. *Operations Research* **26(6)**, pp. 992–1009, 1978.

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

[29] Fleischer, R., Golin, M.J., and Yan, Z. Online maintenance of k-medians and k-covers on a line *In Proceedings of SWAT 2004 Springer LNCS 3111*, pp. 102–113, 2004.

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

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

[32] Fotakis, D. Memoryless Facility Location in One Pass. *ACM Transactions on Algorithms, 7(4) Article 49*, 2011.

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

[34] Fotakis, D. Online and Incremental Algorithms for Facility Location. *ACM SIGACT News Volume 42 Issue 1*, pp. 97–131, 2011.

[35] Fotakis, D., and Koutris, P. Online Sum-Radii Clustering. *Proceedings of Mathematical Foundations of Computer Science - MFCS '12, LNCS 7464*, pp. 395–406, 2012.

[36] Friggstad, Z., and Salavatipour, M. R. Minimizing movement in mobile facility location problems. *ACM Trans. on Alg. (TALG)*, **7(3)**, 28, 2011.

[37] Garfinkel, R.S., Neebe, A.W., and Rao, M.R. An Algorithm for the M-Median Plant Location Problem. *Transportation Science*, **8**, pp. 217–231, 1974.

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

[39] Ghodselahi, A., and Kuhn, F. Serving Online Demands with Movable Centers. *arxiv*, http://arxiv.org/abs/1404.5510

[40] Guha, S., and Khuller, S. Greedy strikes back: Improved facility location algorithms. In *Proc. 9th Symp. on Discrete Algorithms (SODA)*, pp. 649–657, 1998.

[41] Hassin, R., and Tamir, A. Improved complexity bounds for location problems on the real line. *Operations Research Letters* **10**, pp. 395–402, 1991.

[42] Hoffman, A. J., Kolen, A., and Sakarovitch, M. Totally balanced and greedy matrices. *SIAM J. Algebr. Discrete Methods*, **6(4)**, pp. 721–730, 1985.

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

[44] Imreh, Cs., and Németh, T. Parameter learning algorithm for the online data acknowledgment problem. *Optim. Methods Softw.*, **26(3)**, pp. 397–404, 2011.

[45] Kolen, A., and Tamir, A. Covering problems. In: *Mirchandani, P.B., Francis, R.L. (eds.) Discrete Location Theory*, pp. 263–304. Wiley, New York, 1990., Chap. 6

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

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

[48] Németh, T., Gyekiczki, B., and Imreh, Cs. Parameter learning in lookahead online algorithms for data acknowledgment. *Proceedings*

of 3rd IEEE International Symposium on Logistics and Industrial Informatics (LINDI 2011), pp. 195–198, 2011.

[49] Németh, T., Nagy, S., and Imreh, Cs. Online data clustering algorithms in an RTLS system. *Acta Universitatis Sapientiae, Informatica*, **5(1)**, pp. 5–15, 2013.

[50] San Felice, M. F., Williamson D. P., and Lee, O. The Online Connected Facility Location Problem *LATIN 2014: Theoretical Informatics Lecture Notes in Computer Science Volume 8392*, pp 574–585, 2014.

[51] 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.

[52] Shmoys, D. Approximation algorithms for facility location problems. *Proceedings of 3rd International Workshop of Approximation Algorithms for Combinatorial Optimization, Springer-Verlag LNCS vol. 1913 (2000)*, pp. 27–33, 2000.

[53] Sleator, D. D., and Tarjan, R. E. Amortized efficiency of list update and paging rules. *Communication of the ACM*, **28**, pp. 202–208, 1985.

[54] Solis-Oba, R. Approximation Algorithms for the k-Median Problem. *In Efficient Approximation and Online Algorithms, LNCS 3484*, pp. 292–320, 2006.

[55] Tamir, A. Maximum coverage with balls of different radii. *Manuscript (2 pages)*, 2003.

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