Noname manuscript No.

Online Unit Covering and Clustering in Two Dimensions

Mohsen Abbasi ·
Hamid Zarrabi-Zadeh ·
AmirMahdi AhmadiNejad

Abstract In the unit clustering problem, given a set of points on the plane, the goal is to group these points into minimum number of clusters of unit size. In the online version, the points arrive one by one and upon each point's arrival, it must be assigned to some cluster. Another related problem is online unit covering in which moving clusters after opening them is not allowed. In this paper, the Online Unit Clustering and Online Unit Covering problems are studied in two dimensional Euclidean space. An online algorithm with competitive ratio of 5 is presented for the online unit covering problem. In addition, lower bounds of 2.5 and 4.66 are established for these problems.

1 Introduction

Clustering is the problem of dividing a set of points into groups to optimize various objective functions and is used in a wide variety of applications such as information retrieval and data mining. In the Unit Clustering problem, each of these clusters is enclosable by a unit radius circle.
In the online version of this problem, the input is a sequence of points, i.e., every point must be placed inside one cluster without any information about the remaining points to come. On the arrival of every point, we have the option to create a new cluster or assigning some existing cluster to the new point and after the assignment of one cluster to the point, this decision is irreversible. In other words, the points cannot move from one cluster to another. Also we cannot remove, merge or split the clusters.
In this paper we study two related problems:

Problem 1 (Online Unit Covering) A sequence of points arrive on a plane and upon each point's arrival, it must be covered using a circle with unit radius. The goal is to minimize the number of circles used.

Problem 2 (Online Unit Covering) A sequence of points arrive on a plane and upon each point's arrival, it must be placed inside a cluster with unit radius. Radius of a cluster refers to radius of its smallest enclosing ball and the goal is to minimize the number of clusters opened.

In the Online Unit Covering problem, when a unit radius circle is created, its location is fixed but in the Online Clustering problem, the exact location of the cluster is not fixed as long as it covers all the previously assigned points.
The cost of an algorithm for these problems is the number of clusters opened. We measure the performance of an online unit clustering algorithm *Alg* by comparing it to an optimal offline algorithm *Opt* which has all the input sequence at first. This measurement is called competitive ratio and is defined as:

$$sup_{\sigma} \frac{Alg(\sigma)}{Opt(\sigma)}$$

In which, σ shows all the possible input sequences.

Chan and Zarrabi-Zadeh introduced Online Unit Clustering in [1] as a variation on Online Unit Covering which was studied in [2]. Charikar et al. presented an algorithm for the Online Unit Covering problem with competitive ratio $O(2^d d log d)$ in R^d space and provided the lower bound $\Omega(\frac{log\ d}{log\ log\ log\ d})$ for it.
By the use of randomization, a $\frac{15}{8}$-competitive algorithm was presented in [1] for one dimensional space which was improved to a $\frac{11}{6}$-competitive algorithm later in [8]. Epstein et al. designed a $\frac{7}{4}$-competitive deterministic algorithm for this problem in one dimension [7]. The best known upper bound for the Online Unit Clustering in one dimension is $\frac{5}{3}$ which is introduced by Ehmsen et al. in [5]. This results in an upper bound $\frac{10}{3}$ for the problem in two dimensions by the idea of extending the one dimensional solution to upper dimensions in L_∞ norm [1].
Various objective function for the Online Clustering Problem are introduced in [6]. Also Online Clustering with variable-sized clusters is studied in [3] and [4].

Department of Computer Engineering, Sharif University of Technology. m̂abbasi@ce.sharif.edu, zarrabi@sharif.edu, am_ahmadinejad@ce.sharif.edu

Algorithm 1 CENTER2D ALGORITHM

```
Covering ← ∅
while A new point p arrives do
    if p is not covered then
        Create a cicrle c centered at p
        Add c to the Covering
return Covering
```

Our results. In this paper, we obtain some new results on the Unit Covering and Unit Clustering problems, a summary of which is listed below.

– We present an algorithm to solve Online Unit Covering problem with competitive ratio 5, improving upon the current best solution for the problem that offers $O(2^d\ d\ logd)$ in R^d space.
– We present a new 4.66 lower bound for competitive ratio on the Online Unit Covering problem improving upon the current 4 lower bound in 2 dimensions.
– We present a new 2.5 lower bound for competitive ratio on the Online Unit Clustering Problem.

The rest of the paper is organized as follows. In Section 2, we present a new algorithm in order to solve the Online Unit Covering Problem. In sections 3 and 4 we provide new lower bounds for Online Unit Covering and Online Unit Clustering problems respectively.

2 Center2D Algorithm

In this section we present an algorithm to solve the Online Unit Covering problem. In this algorithm, upon each point's arrival, if it is not already covered, a new circle is created centered at this point. We prove that competitive ratio of this algorithm is 5.

Observation 1 *Considering all the circles created by Centered2D algorithm, center point of no circle is covered by another circle.*

Proof We assume that there are two circles A_1 and A_2 with center points p_1 and p_2 respectively, and p_1 is covered by A_2. It's obvious that p_2 is also covered by A_1. If A_1 is created before A_2, then p_2 was already covered by A_1 at its arrival time so the algorithm would have not created A_2. The argument is valid for p_1 if A_2 was created before A_1.

Observation 2 *We consider an optimal circle O centered at o. Any circle with a center in O would cover o.*

Proof It's obvious.

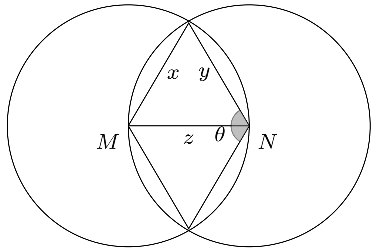

Fig. 1 Covered circumference in each circle is at least $\frac{2\pi}{3}$

Observation 3 *Consider circle O in the optimal covering centered at o. If the center points of a set of circles created by Centered2D, belong to O, in case these circles cover the circumference of O, the area of O is covered completely.*

Proof Suppose that there is some point p in O which is not covered by any of the circles in the set. We define a line crossing p and o and assume that intersection points of this line and O are m and n. Both m and n are covered by some circle which also contains o (Observation 2) so this circle also covers p which contradicts the assumption. Therefore there is no point in O which is not covered by the set of circles.

Lemma 1 *Consider two circles with the same size which contain eachother's center points. Circumference of each circle which is covered by the other is at least $\frac{2\pi}{3}$. And if the two circles don't share a same center, the covered circumference of each one is less than π.*

Proof We assume that each center point is located on the circumference of the other circle. As it is shown in Figure 1, x, y and z have equal lengths and are the three sides of a triangle. Therefore $\angle\theta$ is $\frac{2\pi}{3}$. It's obvious that if the overlapped area is increased, the covered circumference of each circle is increased too. Also, if the covered circumference of two circles equals π, we can conclude that two circle share a same diameter, therefore their center points are the same.

Lemma 2 *We assume that O is a circle in the optimal covering. By creating a new circle A centered at some point in O, if A does not overlap with more than one circle created by the algorithm, at least another $\frac{\pi}{3}$ of the O's circumference is covered.*

Proof If A has no overlap with any previously created circles by the algorithm, according to Lemma 1, it will cover at least another $\frac{2\pi}{3}$ of O's circumference. So we consider the case in which A overlaps with another circle. As we see in Figure 2 if A is centered at p it covers $\frac{\pi}{3}$ of O's circumference (half of the case in which there

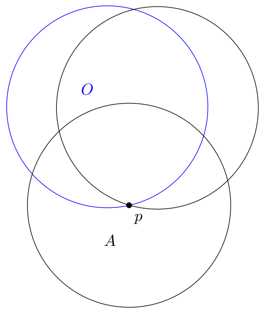

Fig. 2 The covered circumference of O increases at least $\frac{\pi}{6}$ by creating a new circle.

was no other circle). Of course the new point is not contained by the previously created circle. By moving the point (and the center of A) both alongside the circumference of created circle or further from it, the covered circumference of O by A is increased. Therefore at least another $\frac{\pi}{3}$ of O's circumference is covered by A.

Theorem 4 *Competitive ratio of Center2D algorithm is 5. This competitive ratio is tight.*

Proof We consider a hypothetical circle O in the optimal solution and assume that O will be covered by six circles created by the algorithm. By entering the first point in O, algorithm creates a circle centered at this point. We can divide O to six isomorph sectors as illustrated in Figure 3, in a way that the arrived point is on the border of two of the sectors. According to Lemma 1 and Observation 2, the circle which was created to cover the point, at least covers the two sectors incident to the point and at most four sectors are left uncovered. By Lemma 2, we know that by entering a point in each of the remaining sectors and creating a circle centered at this point, the corresponding sector is completely covered. Therefore by the assumption that O is covered by 6 circles centered in O, at least one center point is located in another circle which is in contradiction with Observation 1. Therefore at most five circles will be created to cover any circle in the optimal solution.
As shown in Figure 4 by the Centered2D Algorithm, there is a 5 competitive ratio example therefore this competitive ratio is tight for this algorithm.

3 Lower Bound for Online Unit Covering problem

In this section we present a new lower bound for the competitive ratio of algorithms for the Online Unit Covering problem. First, we provide some observations and

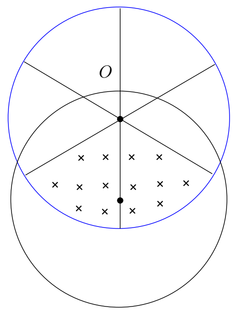

Fig. 3 By entering the first point in O, a circle centered at this point is created which covers at least two sectors

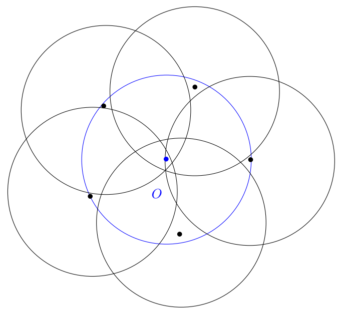

Fig. 4 5 competitive ratio example for Center2D algorithm

lemmas which will help us to prove our claim.

Observation 5 ***Loose Circle*** *We consider a set of points that have been covered by some arbitrary covering algorithm. We also assume that all these points are covered by only one circle O in the optimal solution. In case there are still two uncovered points with a distance more than 2 that each can be contained by O separately, a sequence of points (which are not already covered) in O can arrive that forces any algorithm to create at least two more circles.*

Proof We consider two circles that each one covers one of the points. If the next entry point appears in the overlapped area between these two circles, the algorithm has no way to find out which of the two circles are the optimal circle, so by selecting each of them, the adversary can decide to give the next point in the other circle.

Observation 6 *Consider the case in which there are three circles and each one crosses the other two's center points (as shown in Figure 5). In this state, the covered circumference of each circle is π. By relocating any of these circles, the covered circumference of at least one of them would reduce to less than π.*

Proof line segments oo', oo'' and $o'o''$ have equal sizes therefore, $\angle oo'o''$, $\angle o'o''o$ and $\angle o''oo'$ and their corresponding sectors are $\frac{\pi}{6}$. Also by Lemma 1, we know that overlapped sectors between each two circles is $\frac{2\pi}{3}$,

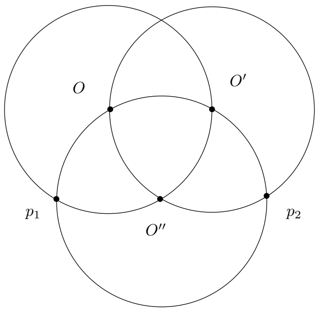

Fig. 5 Every circle crosses the center points of the other two.

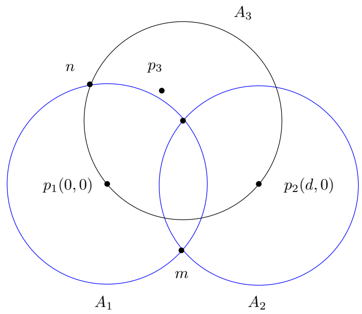

Fig. 6 Independent from the value of d, the circle crossing m and n has radius of unit size.

therefore the total covered circumference of each circle is π.
The covered circumference of each circle is π so its obvious that points p_1 and p_2 are placed on a diameter of O'', therefore the sectors of O and O' which are equally covered by O'' are maximized simultaneously. By relocating O'' in any direction, at least one of the sectors which is exclusively covered by O'' is reduced.

Lemma 3 *Suppose there are two circles A_1 and A_2 with center points p_1 and p_2 respectively and the distance between p_1 and p_2 (d) is less than $\sqrt{3}$. Also there is a circle A_3 crossing p_1 and p_2 that covers some arbitrary point p(as shown in Figure 6). If p_1, p_2 and p_3 are all covered by the same circle in the optimal solution, the points of this circle which are not covered by A_1, A_2 or A_3, are all positioned at the same side of hypothetical line p_1p_2.*

Proof We first show that there is a unit radius circle covering p_1, p_2, p_3, m and n. Considering an xy-plane, without loss of generality we assume that p_1 and p_2 are positioned at $(0,0)$ and $(d,0)$ respectively and $d < \sqrt{3}$ (so A_3 would not cover both intersection points of A_1 and A_2). The intersection points of circles A_1 and A_2 are $(\frac{d}{2}, \pm\frac{\sqrt{4-d^2}}{2})$. In the next step, we calculate the intersection point of circles A_1 and A_3 which is $n = (\frac{d-\sqrt{3}\sqrt{4-d^2}}{4}, \frac{\sqrt{4-d^2}+\sqrt{3}d}{4})$. Now in order to calculate the radius of the circle crossing p_2, m and n, we use this formula:

$$r = \frac{\|p_2 - m\|}{2sin\theta}$$

In which, θ refers to $\angle p_2nm$. By calculating θ we reach this equation based on d:

$$tg\theta = -\frac{8d\sqrt{4-d^2}+8\sqrt{3}}{8\sqrt{3}d\sqrt{4-d^2}+24}$$

which is a constant function. By putting an arbitrary $1 < d < \sqrt{3}$ in the formula we get $r = 1$.
The circle covering all p_1, p_2, m, n has a radius of size 1.

Therefore no circle in the optimal covering which covers p_1, p_2 and arbitrary p_3, can have uncovered points in both sides of the hypothetical line p_1p_2.

Corollary 1 *Observation 6 states that if the algorithm creates three overlapping circles, the only way to make sure that the covered circumference of each one is π, is to position them so they cross eachothers center points. It is important that each circles covered circumference to be at least π so the loose circle case as explained in Observation 5, doesn't happen. But Lemma 3 states that if two of the circles are centered at two points (which are covered by the same circle in the optimal solution), the loose circle won't happen as long as the third circle crosses the center points of the other two.*

Lemma 4 *Consider a circle A centered at p and some arbitrary point p_1 which is covered by it (p and p_1 are not the same point). We assume that p_1 is not in the same circle as p in the optimal covering. There is a sequence for the uncovered points in the circle covering p_1, which enforces the algorithm to create at least 4 more circles to cover it.*

Proof Consider a segment with $1+\epsilon, \epsilon > 0$ length, which starts from p and crosses p_1. As we see in Figure 4 the next point in sequence which is p_2 enters at the other end of this segment (just enough for p_2 to not get covered by A) and some arbitrary circle is created to cover it. In the next step p_3 enters and some arbitrary circle covers this point too. By Observation 5, we know that if the covered circumference of one of the three circles is less than π, the loose circle case happens. By Lemma 3, we know that the only way to prevent this from happening is to open A_1 centered at p_2 and choose the location of third circle so it crosses the other two's center points; if this is not the case then the algorithm creates two more circles and we are done, Therefore we consider the case where circles are positioned in a way that each one crosses the center points of the other two. As shown in Figure 4, the optimal circle which covers

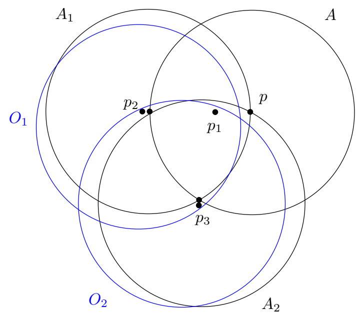

Fig. 7 Each of O_1 or O_2 could be the optimal circle covering p_1, p_2 and p_3.

p_1, p_2 and p_3, could be each of O_1 or O_2 (loose circle). Therefore by Observation 5, we know that there is a way to make algorithm create two more circles.

Corollary 2 *By the same reasoning as lemma 4, we can conclude that if points p_1 and p are the same, there is a sequence which enforces the algorithm to create at least 3 other circles.*

Now we give two sequences of points that in each of them, any covering algorithm has to create at least 9 circles. In both cases at first two points p_1 and p_2 are given to the algorithm with distance $1+\epsilon, \epsilon > 0$, and we assume that algorithm creates two circles A_1 and A_2 centered at these points in order to cover them. In the end we show that this assumption, doesn't affect the outcome.

Sequence 1:
After the arrival of p_1 and p_2 which are covered by A_1 and A_2, we assume that the point p is given to algorithm as shown in Figure 8. Every point in the specified location in Figure 8 has a distance more than 1 from m and n. Therefore if a circle with a center point in the specified location is created to cover p, the loose circle case happens and according to Observation 1, there is a sequence which enforces the algorithm to create two more circles when the optimal solution uses just 1 circle which leads to competitive ratio 5. In order to prevent that from happening, any circle which is created to cover p must be centered in the areas of O which are not hachured. According to lemma 2, if the circle covering p, is not the same circle as the one which covers p_1 and p_2, there is a sequence for the remaining points in this circle which enforces any algorithm to use at least 4 other circles to cover it. Also the loose circle case happens for the optimal circle that covers p_1 and p_2. Therefore, in order to cover all the entered points, the algorithm creates 9 circles although the optimal solution uses 2 circles.

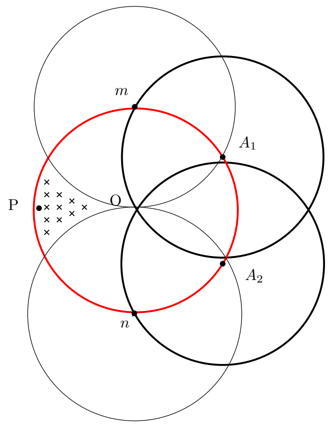

Fig. 8 Every point in the specified area has a distance more than 1 from m and n.

Sequence 2:
As explained before, we assume that the algorithm has already created two circles A_1 and A_2 centered at p_1 and p_2, in order to cover them. Let's say the next point arrives at p_3, As shown in Figure 9. We consider two cases:

1. The algorithm creates a circle centered at O. In this case we can assume that the point p_1 and p_2 are covered by different circles in the optimal solution. By corollary 2, we know that adversary could make algorithm create 3 other circles to cover the optimal circles completely. Therefore the algorithm creates 9 circles while the optimal solution uses 2 circles.
2. The algorithm creates an arbitrary circle centered at a point different from O. In this case the second point arrives at p_4. The algorithm has to cover all the uncovered points in O (otherwise the competitive ratio would be 5), Therefore at least one of the circles covering p_3 or p_4 has to cover hypothetical point m. Without loss of generality, we assume that this is the circle covering p_3. The fifth point arrives at p_5. Its obvious that in order to prevent a loose circle case (as shown in Figure 10) the new circle must be centered at a point within the A_1 (as shown in Figure 11) in which also the loose case observation could be applied. Therefore either ways, the algorithm uses 5 circles to cover the optimal circle covering p_1. By corollary 1, we know that the algorithm may create 3 more circles to cover the other optimal circle, therefore in total, 9 circles are created by the algorithm while the optimal solution uses just 2 circles.

In both sequences we assumed that the algorithm covered p_1 and p_2 by A_1 and A_2 which were centered at them. We show that this doesn't affect the generality of our reasoning. At first we assume that A_1 is centered

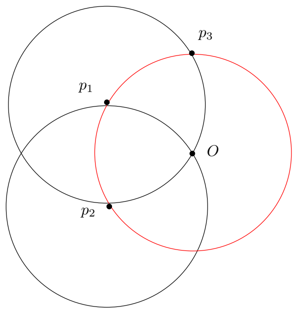

Fig. 9 We consider two cases after the arrival of point p_3.

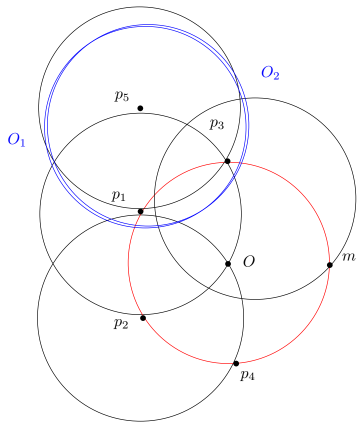

Fig. 10 After covering p_5, loose circle case happens.

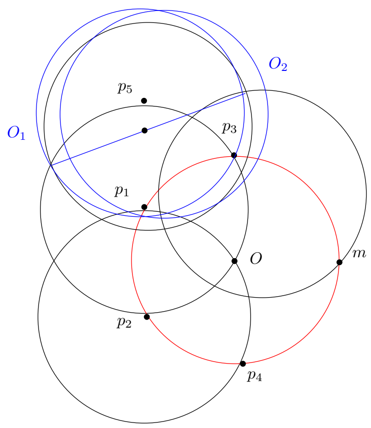

Fig. 11 Even after covering p_5 with a circle centered in A_1, loose circle case happens.

at p_1 and A_2's position is arbitrary and the third point arrives at p_3. If p_1, p_2 and p_3 belong to the same circle in the optimal solution, by Observation 1 and corollary 0, we know that after covering p_3 by creating the third circle, the only way to make sure the loose circle case won't happen is to choose the center of A_2 at p_2. Otherwise the competitive ratio would be 5. The other case is when A_1 is not centered at p_1. Consider a hypothetical segment which starts from A_1s center and crosses p_1 and the A_1 circumference. If p_2 arrives at the other end of this segment, the previous reasoning holds true.

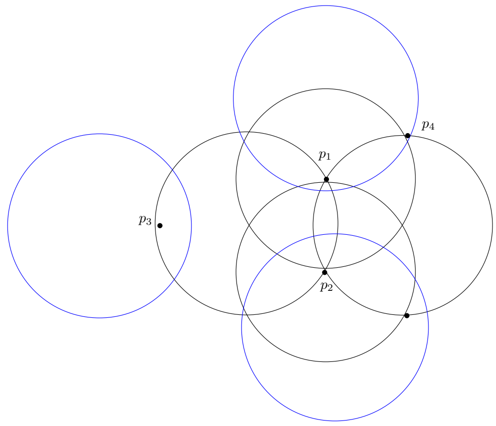

Fig. 12 The algorithm is forced to create 14 circles to cover the points while the optimal solution uses 3.

Theorem 7 *No deterministic algorithm for the Online Unit Covering problem can have a competitive ratio less than* 4.66*.*

Proof First p_1 and p_2 arrive and get covered by A_1 and A_2 which are respectively centered at p_1 and p_2 as we see in Figure 12. The third point arrives at p_3. As same as sequence 1, the algorithm would not create the next circle centered at p_3. A sequence of points arrives in a way that the algorithm has to create 4 more circles to cover the optimal circle covering p_3. The next point arrives at p_4. As same as sequence 2, the algorithm has to create 9 circles to cover the optimal circles for p_1 and p_2. Therefore algorithm uses 14 circles to cover all points while the optimal solution uses 3.

4 Lower bound for the Online Clustering Problem

The best known lower bound on the competitive ratio of Online Unit Clustering algorithms in 2 dimensions, is $\frac{13}{6}$ that is provided for the L_∞ norm [7]. In this section we present a better lower bound for this probblem in Euclidean norm.

Theorem 8 *No deterministic algorithm for the Online Unit Clustering problem in Euclidean plane can have a competitive ratio less than 2.5.*

Proof Suppose two points x_1 and x_2 with distance 2 arrive. Two cases can happpen:

1. Algorithm puts both points in one cluster. We assume that these points are not in the same cluster in optimal solution and points x_3 and x_4 arrive as shown in Figure 13. The distance between x_3 and x_4 is also 2. Consider two cases:

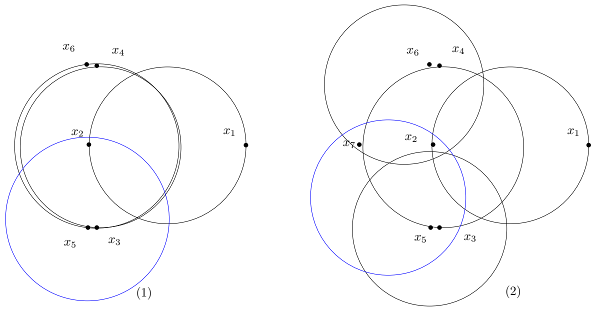

Fig. 13 The algorithm puts x_3 and x_4 in the same circle.

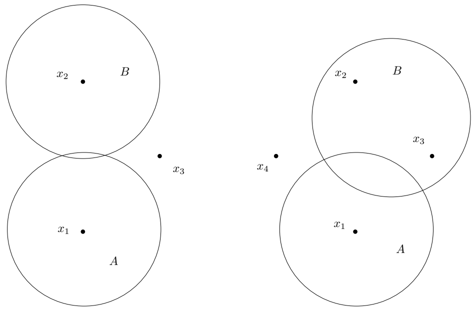

Fig. 14 x_3 and x_4 arrive in a way that all points can be covered by on circle.

- The algorithm puts both points in one cluster. In this case points x_5 and x_6 with distance 2 arrive. If algorithm puts x_5 and x_6 in one cluster first case in Figure 13 happens. Another cluster is needed to cover the optimal cluster shown, also the sequence can repeat for the optimal cluster covering x_4 and x_6 and the optimal cluster covering x_1. This way the algorithm opens 10 clusters to cover the points while the optimal solution uses 4 clusters. In case algorithm put x_5 and x_6 in separate clusters, second case in Figure 13 happens. When x_7 arrives, the algorithm has to use one of the clusters to cover it otherwise the competitive ratio exceeds 2.5. In this case x_7 is put in the same cluster as x_4 and x_6 while in optimal solution it's grouped by x_3 and x_5. Therefore the sequence (x_3, x_4, x_5 and x_6) can repeat for the cluster covering x_4, x_6 and x_7. This way the algorithm opens 10 clusters to cover the points while the optimal solution uses 4 clusters.
- The algorithm puts each of x_3 and x_4 in one cluster. If this case happens to the optimal cluster covering x_1, we are done. If not, arriving a sequence such as previous section, leads to a competitive ratio more than 2.5.

2. Algorithm puts x_1 and x_2 in separate clusters. In this case x_3 and x_4 arrive as shown in Figure 14. The distance between x_3 and x_4 is 2 and they appear in a way that all x_1, x_2, x_3 and x_4 could be covered by one circle. To prevent the competitive ratio from exceeding 2.5 the algorithm has to cover x_3 and x_4 by the existing clusters. Figures 15 and 16 show both ways that the algorithm can cover these points and the clusters in the optimal solution. By repeating a sequence as same as the previous sections, in both cases the algorithm opens 10 clusters while the optimal solution uses 4 clusters.

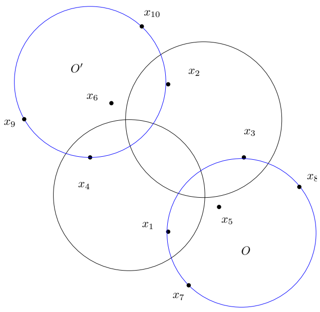

Fig. 15 One cluster covers x_3 and one cluster covers x_4.

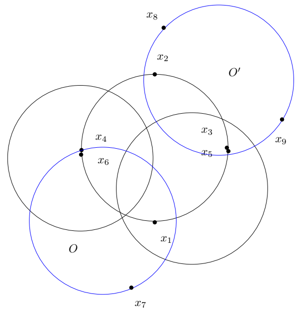

Fig. 16 One cluster covers both points.

References

1. T. M. Chan and H. Zarrabi-Zadeh. A randomized algorithm for online unit clustering. In *Approximation and Online Algorithms*, pages 121–131. Springer, 2007.
2. M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental clustering and dynamic information retrieval. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 626–635. ACM, 1997.
3. J. Csirik, L. Epstein, C. Imreh, and A. Levin. Online clustering with variable sized clusters. In *Mathematical Foundations of Computer Science 2010*, pages 282–293. Springer, 2010.
4. G. Divéki and C. Imreh. An online 2-dimensional clustering problem with variable sized clusters. *Optimization and Engineering*, 14(4):575–593, 2013.
5. M. R. Ehmsen and K. S. Larsen. *Better bounds on online unit clustering*. Springer, 2010.
6. L. Epstein, A. Levin, and R. van Stee. Online unit clustering: variations on a theme. *Theoretical Computer Science*, 407(1):85–96, 2008.

7. L. Epstein and R. Van Stee. On the online unit clustering problem. In *Approximation and Online Algorithms*, pages 193–206. Springer, 2008.
8. H. Zarrabi-Zadeh and T. M. Chan. An improved algorithm for online unit clustering. In *Computing and Combinatorics*, pages 383–393. Springer, 2007.