

**M. Tech. (Computer Science) Dissertation Series**

# **Computation of Minimum Width Rectangular Annulus**

**a dissertation submitted in partial fulfilment of the  
requirements for the M. Tech. (Computer Science)  
degree of the Indian Statistical Institute**

*By*

Joydeep Mukherjee

CS0801

**under the supervision of**

Dr. Sandip Das

ACMU



**INDIAN STATISTICAL INSTITUTE**

203, Barrackpore Trunk Road

Calcutta-700 035

# Computation of Minimum Width Rectangular Annulus

A disseratation submitted in partial fulfillment of the requirement of  
M.Tech(Computer Science)Degree of the Indian Statistical Institute,Kolkata  
by

**Joydeep Mukherjee**

Under the supervision of  
**Dr.Sandip Das**

Advanced Computing and Microelectronics Unit,  
Indian Statistical Institute ,Kolkata

## Acknowledgments

With great pleasure and sense of obligation I express my heartfelt gratitude to my guide Dr.Sandip Das (ACMU) . I am highly indebted to him for his invaluable guidance and his readiness for anytime help. I also convey my thanks to Prof.S.C.Nandy and Dr.Arijit Bishnu but for whose persistent efforts in their teaching I could not have thought of attempting to solve the problem.

I thank Sumitda (Research Scholar) who helped me to learn Latex(apart from many other helps). In this regard I should also thank the research scholars at ACMU Priyada, Arindamda, Sruti, Aritrada who became instrumental in drafting my theses.

Without wasting this valuable chance, I want to thank my friends Sumanta, Saurabh, Minati, Bahadur, Anisur, Soumitra, Soumyattam without whose support I could not have made this report a reality.

### Abstract

In this work we consider the problem of finding two parallel rectangles in arbitrary orientation one rectangle contained in another such that given a point set the entire point set is contained within the annulus formed by the two rectangles and the maximum euclidean distance between any two sides of the possible pair of rectangles is minimized. We propose an algorithm of time complexity  $O(n^2 \log n)$  with a preprocessing time of  $O(n^3)$  and space complexity  $O(n^2)$ .

# 1 Introduction

Two rectangles are said to be parallel when each side of a rectangle is parallel to a side of the other rectangle. In this work we consider the following problem. A point set  $P$  is given. We try to find a pair of parallel rectangles so that one rectangle is fully contained in another and  $P$  is contained within space enclosed by the two rectangles such that the maximum euclidean distance between the parallel sides is minimized among all such pair of rectangles.

Some work has been done based on the same notion. The most studied annulus problem is that of circular annuli as done by Aggarwal and Sharir, Goodrich. But the characterization used there does not easily extend to in case of rectangles when the distance is euclidean. Gluchshenko studied a similar problem in the isothetic direction but the distance metric was rectilinear. He in his paper gives an optimal algorithm of  $O(n \log n)$ . Barequet, Goodrich deals with a similar problem where the external convex polygon is given and an annulus of minimum size is found out such that all points are contained in the annulus. They also treat another variant of the problem where given the size of the annulus they try to maximize the point containment. Some other works related to arbitrary orientation of rectangles also require a mention here. Two very apparently similar works are as follows. One by Chaudhuri, Nandy and Das where they compute the maximum area empty rectangle in a given point set. The other is by Saha and Das where they find out two parallel rectangles such that for covering a given set of points such that the area of the larger rectangle is minimized.

## 2 Preliminaries

**Observation 1.** *In a MWMRA, the center of the outer rectangle must lie inside the inner rectangle.*

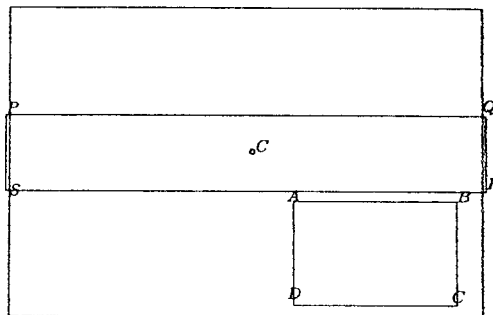


Figure 1:

*Proof.* Suppose the center point  $C$  of the external rectangle is outside the internal rectangle  $ABCD$  as shown in the fig. Let  $ABCD$  be the MWMRA. We claim that we can always find a better rectangle than  $ABCD$ . Take the nearest point above  $C$  and below  $C$ . Draw two parallel horizontal lines through those two points. It's easy to see that the rectangle has its vertical side coincided with the external rectangle. As shown in the fig this rectangle is  $PQRS$ . It's easy to verify that  $\maxgap(ABCD) > \maxgap(PQRS)$ . Thus  $ABCD$  is not MWMRA. Hence the proof.  $\square$

**Observation 2.** *Assume the center of the outer rectangle is contained in the inner rectangle and the centers of these two rectangles are not coincident. They can be made to coincide with suitable translation of the sides of the outer rectangle. By these translations, the rectangular annulus of the minimum width found can be made a MWMRA.*

## 3 Algorithm for finding out MWMRA in the isothetic direction

The following algorithm will compute axis parallel MWMRA.

1. Find the center  $C^*$  of the outer rectangle  $MER$ .
2. Choose the points to the right of the point  $C^*$  and store in an array  $R$  in ascending order.

3. Take the points to the left of the point  $C^*$  and keep these points in an array  $L$  in descending order.
4. Similarly, compute an array  $U$  to store the points above  $C^*$  in descending order.
5. Compute an array  $B$  to store the points below  $C^*$  in ascending order.
6. Compute a axis parallel rectangle that contains the points  $L[1], R[1], U[1], B[1]$  on its left, right, top and bottom boundaries respectively.
7. Initialize  $i = 1$  and  $choice = 0$   
 $MAXGAP1 = findMAXGAP()$ .
8. **while**( $i! = n$ )
  - 8.1  $i = i + 1$
  - 8.2 **If**( $y(R[i - 1]) > y(C^*)$  and  $y(L[i - 1]) > y(C^*)$ ) and ( $y(R[i - 1]) < Y_u$  or  $y(R[i - 1]) < Y_u$ )
    - 8.2.1  $Y_u = \min(y(R[i - 1]), y(L[i - 1]))$
  - 8.3 **Else**
    - 8.3.1 **If** ( $y(R[i - 1]) > y(C^*)$  and  $y(R[i - 1]) < Y_u$ )
      - 8.3.1.1  $Y_u = R[i - 1]$
    - 8.3.2 **if** ( $y(L[i - 1]) > y(C^*)$  and  $y(L[i - 1]) < Y_u$ )
      - 8.3.2.1  $Y_u = L[i - 1]$
  - 8.4 **If**( $y(R[i - 1]) < y(C^*)$  and  $y(L[i - 1]) < y(C^*)$ ) and ( $y(R[i - 1]) > Y_b$  or  $y(R[i - 1]) > Y_b$ )
    - 8.4.1  $Y_b = \max(y(R[i - 1]), y(L[i - 1]))$
  - 8.5 **Else**
    - 8.5.1 **If** ( $y(R[i - 1]) < y(C^*)$  and  $y(R[i - 1]) > Y_b$ )
      - 8.5.1.1  $Y_u = R[i - 1]$
    - 8.5.2 **if** ( $y(L[i - 1]) < y(C^*)$  and  $y(L[i - 1]) > Y_b$ )
      - 8.5.2.1  $Y_b = L[i - 1]$
  - 8.6  $MAXGAP2 = findMAXGAP()$ .
  - 8.7 **If** ( $MAXGAP1 \leq MAXGAP2$ )
    - 8.7.1  $Choice+ = GetArrayId()$
    - 8.7.2 **switch** (Choice)

Case-1:  $i = i - 1$ ; disable L; break  
 Case-2:  $i = i - 1$ ; disable R; break  
 default:  $flag = 1$   
 8.7.3 if( $flag == 1$ )  
     8.7.3.1 break  
 8.8 Else  
     8.8.1  $MAXGAP1 = MAXGAP2$

In this algorithm, the function  $findMAXGAP()$  calculates the maximum gap among four possible widths and the function  $GetArrayId()$  returns the id of the array which need not be considered to traverse. Here we use the the id values of the arrays  $L$  and  $R$  as 1 and 2 respectively. Here  $Y_u$  and  $Y_b$  denote the  $y$ -coordinates of the point on the top and top boundaries of the inner rectangle respectively.

## 4 Definitions related to the Algorithm for arbitrary orientation

**Definition 1.** *Rotational Replacement:* While rotating the internal rectangle if two points come on one of its sides, a choice is made between the two points such that when the rectangle is further rotated about the chosen point the rectangle remains empty.

**Definition 2.** *Orthogonal Distance Pair:* Suppose a point  $P$  forms the vertex of the internal rectangle. The orthogonal distance pair corresponding to  $P$  is the pair of distance, where each distance is the distance between the side of the internal rectangle passing through  $P$  and corresponding side of the external rectangle.

**Definition 3.** *Continuous Domain:* The angular interval in which the points on the side of the external rectangle does not change.

**Definition 4.** *Point Pairing:* The Point Pairing in an angular interval, is the set of two points such that the corresponding sides of the internal and the external rectangle passes through these two points in that angular interval.

**Definition 5.**  $L_{L\psi}(ABCD)$ : Its the distance between the left side of the internal rectangle  $ABCD$  and the external rectangle when  $ABCD$  is drawn at



an orientation  $\psi$ .

Similarly  $L_{R\psi}(ABCD), L_{B\psi}(ABCD), L_{T\psi}(ABCD)$  are for right, bottom and top sides respectively.

**Definition 6.**  $L_L(ABCD)$ : Its the distance between the left side of the internal rectangle  $ABCD$  and the external rectangle.

Similarly  $L_R(ABCD), L_B(ABCD), L_T(ABCD)$  are for right, bottom and top sides respectively.

**Definition 7.**  $L_{L\psi}(X)$ : Suppose the left side of the inner rectangle contains a point  $X$ .  $L_{L\psi}(X)$  denotes the distance of the left side of the inner rectangle and the left side of the outer rectangle when the inner rectangle is drawn at an orientation  $\psi$  and the left side of the inner rectangle contains the point  $X$ .

Similar things are denoted by  $L_{R\psi}(X), L_{B\psi}(X), L_{T\psi}(X)$  except in place of left side they are right, bottom and top sides respectively.

**Definition 8.**  $maxgap(ABCD)$ : This denotes the maximum among the four distances between the corresponding sides of the internal rectangle  $ABCD$  and the external rectangle.

**Definition 9.** Exchange Operation: In a quadruple, if a point is replaced by another point s.t when the replacement was performed they were not on the same side of the rectangle, then this replacement is said to be an exchange operation, provided this replacement gives a better rectangle in terms of  $maxgap$ .

## 5 Algorithm

**Algorithm in the continuous domain:**

**Aim of the algorithm:** The algorithm starts with the best rectangle which is found in the isothetic direction by using the above stated algorithm. The following algorithm then takes this rectangle and computes the best rectangle in the continuous domain.

**Step 1:** Draw an external rectangle in the isothetic direction;

**Step 2:** Draw the internal rectangle in the isothetic direction such that it gives the best rectangle pair in terms of  $maxgap$ . The internal rectangle is computed using the Algorithm for finding out MWMRA in the isothetic direction.

**Step 3:** Rotate the internal rectangle in CCW.

**Step 4:** If two points come on one side apply *Rotational Replacement*. Then go to step3.

**Step 5:** If *orthogonaldistancepair* becomes equal and if one of the distances corresponds to the maxgap, then an *exchangeoperation* is done such that the distance in the direction of maxgap is reduced, provided the other distance not in the direction of maxgap reduces in CCW direction. Then go to step3.

**Step 6:** *shiftmaxgapevent* While rotating in CCW if the maxgap becomes equal to some other side distance, then *exchangeoperation* is done such that the distance in that other side is reduced, provided the *exchangeoperation* is feasible. Then go to step3.

**Step 7:** Do this until the external and internal rectangle pair does not leave the continuous domain.

The same algorithm is repeated with orientation being in the clockwise direction.

## 6 Correctness of the Algorithm

**Theorem 1.** *For a point pairing, in a continuous domain, the distance between the parallel lines through the pair of points never decrease and then increase.*

*Proof.* To prove this we make a short claim. The distance between two parallel lines passing through two fixed points is maximum when the parallel lines passing through them are perpendicular to line segment joining the two points. Refer to fig. Suppose S is rotated CW as shown. But since AB becomes the hypotenuse of  $\triangle APB$  length of  $AB >$  length of  $BP$ . But the length of BP is the distance between V and U. Its also easy to see from similar arguments that length is decreasing as S is moved CW or CCW until the parallel lines coincide. But this will never happen, rather the orientation at which this

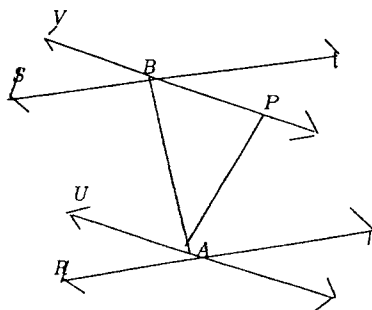


Figure 2:

is supposed to happen will destroy the point pairing since one of the points is a hull point. Hence the proof.  $\square$

If we can prove that the algorithm gives the best rectangle at any orientation within the continuous domain and considering the rectangles at the event points suffice then we are done. In order to prove that it gives the best rectangle we consider the following two points.

1. When the rect. is rotated between two event points it always gives the best rectangle.
2. The steps taken by the algorithm at a certain orientation when an event point occurs yields the best rect. in that orientation.

We prove (1) and (2) by induction. The proof of the base case is similar. Suppose upto  $k^{th}$  event point we get the best rectangle. We take an arbitrary orientation  $\phi$  in between  $k^{th}$  and  $k+1^{th}$  event points. Now we consider the following cases.

CASE1.1 : Suppose an RR (Rotational Replacement) occurs at the  $k^{th}$  event point.  $ABCD$  goes to  $A'B'C'D'$  with RR as shown in fig(8). RR replaces point  $P$  by point  $Q$ . Now our claim is that we can not get a better rect. than  $A'B'C'D'$  at an orientation  $\phi$ . Following our induction hypothesis we got the best rect  $ABCD$  at the  $k^{th}$  event point. In order to prove that  $A'B'C'D'$  is the best rect. at orientation  $\phi$  we consider the following two possibilities.

CASE1.1.1 :  $AD$  has the maxgap. Since no other event has occurred while the rectangle is rotated from the  $k^{th}$  instant orientation to  $\phi$  we can say  $A'D'$  has also the maxgap. In order to get a better rect. than  $A'B'C'D'$ ,  $A'D'$  must be given a translational shift such that gap corresponding to  $A'D'$  decreases in the same orientation. If it occurs then point  $Q$  enters into  $A'B'C'D'$  as  $T$  is fixed. Thus in order to make it happen point  $Q$  should come on the bot-

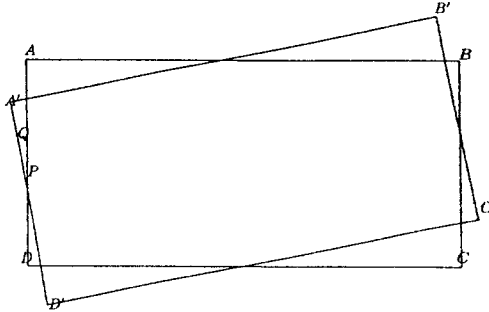


Figure 3:

tom side. But that indicates that an exchange operation should take place. Or said explicitly  $L_{B\psi}(Q) = L_{L\psi}(Q)$  where  $\psi$  is an orientation between the orientation at the  $k^{th}$  instant and  $\phi$ . We say  $\psi$  to be such an orientation because this can happen only after the same equality holds for  $P$ . But this again is an event point. We are considering the interval where no event point occurs. Hence a contradiction.

CASE1.1.1: RR occurs at a non-maxgap side. In these cases also the argument follows the similar lines.

CASE1.2: Suppose an exchange operation occurs at the  $k^{th}$  instant which

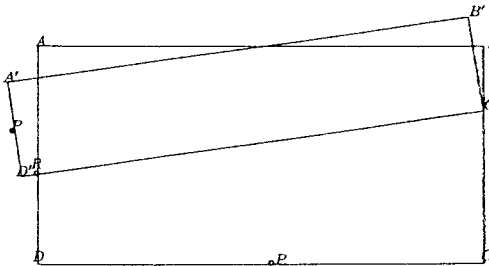


Figure 4:

gives the best possible rect. As shown in the fig the exchange operation replaces point  $Q$  by point  $P$ .  $A'B'C'D'$  at the  $k^{th}$  instant after a CCW rotation such that no event point occurs in between. W.l.g we assume  $A'D'$  has the maxgap and since no event point occurs within the specified interval we have  $A'D'$  to be the maxgap side. The only way to construct a better rect. at  $\phi$  is to make a translation so as to reduce the maxgap. For this  $P$  has to be on the bottom side. Now following the same line of argument as in CASE1.1.1 we can conclude that  $A'B'C'D'$  so obtained is the best possible rect.

CASE1.3 : Suppose at the  $k^{th}$  event point the gap of one side becomes equal to the maxgap. The side whose gap become equal to the maxgap we say it to be the *shiftside*. In this case we consider the following two possibilities as faced by the algorithm.

CASE1.3.1:an exchange operation is done.

CASE1.3.2:an exchange operation is not done.

CASE1.3.1 : When a *shiftside* due to rotation of the internal rectangle its evident that the maxgap would occur on the *shiftside* if any further rotation is done. When an exchange operation is done we can argue as in that of CASE1.2 to prove that we get the best rectangle as in the interval considered.

CASE1.3.2 : Suppose maxgap was on the left side just before the  $k^{th}$  event

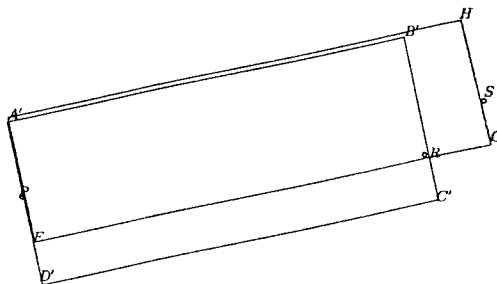


Figure 5:

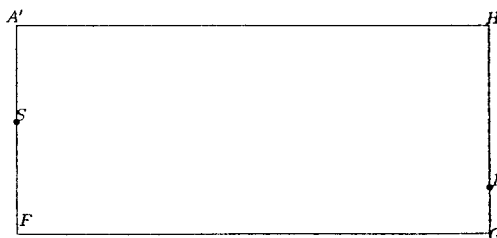


Figure 6:

point and at this event point it becomes equal to the right gap. So any further rotation in CCW would imply that the right side gap would be dominating. This justifies the fact that the right side of the rect. we started with gets shifted as shown in the fig. In this case as the rectangle was rotated to  $\phi$  we get  $A'B'C'D'$ . Suppose we get a better rect  $A'FGH$ . Now if we consider the rect.  $A'FGH$  rotated  $\phi$  in backwards direction, then we obtain a rect. as shown in the fig. But this contradicts our assumption that  $ABCD$  is the best rect. at the  $k^{th}$  event point as it implies an exchange operation. But we are

considering the case where no exchange operation has been done. Now suppose an exchange operation is being done within the interval but the event goes unnotified. In order for this to happen somewhere the orthogonal distance pair should be equal otherwise we are in the same problem of infeasibility of the exchange operation. But the equality of the orthogonal distance pair is an event point that the algorithm takes care of. Thus in this case also we get the best rectangle in the interval considered.

Now we want to prove that the  $k^{th}$  event point rectangle implies the best rectangle at  $k + 1^{th}$  event point. By our induction hypothesis  $k^{th}$  event point rectangle is the best rectangle at that orientation. Thus if we can prove that  $k^{th}$  event point rectangle implies the best rectangle at  $k + 1^{th}$  event point and the rect. so obtained is the best in that direction then it's done.

Suppose at the  $k + 1^{th}$  event point an exchange operation takes place due to equality of orthogonal distance pair.

We make a short claim: In a particular direction if an exchange operation preserves the maxgap, then the pair of rectangles so obtained are the optimal in that particular direction.

Following this claim we can say that we obtain the best rectangle in the orientation where the  $k + 1^{th}$  event point occurs. But the rectangle from which this was obtained was also the best rect. in this orientation which in turn was obtained by the rect. at the  $k^{th}$  event point. Thus in this case  $k^{th}$  event point rectangle implies the best rectangle at  $k + 1^{th}$  event point.

Suppose at the  $k + 1^{th}$  event point RR takes place. Consider the fig be-

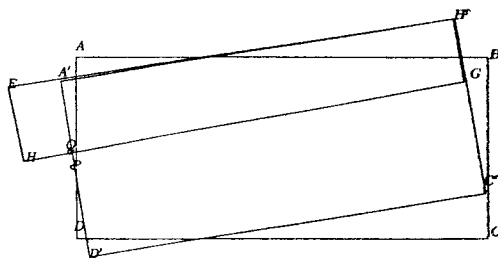


Figure 7:

low. Let  $ABCD$  be the  $k^{th}$  event point rectangle.  $A'B'C'D'$  will be the best rect when two points  $P$  and  $Q$  come on the side  $A'D'$ . This follows from the proof of the fact that we get the best rect. in all the orientations occurring

between two consecutive event points. Let  $EB'GH$  be the best rect. in that orientation as shown in the fig. (assuming  $A'D'$  to be the maxgap side). But this means we could have started with the rect. with  $Q$  at the bottom side at the  $k^{th}$  event point. This defies the fact that  $ABCD$  is the best rectangle at  $k^{th}$  event point because  $AD$  side or  $A'D'$  side is the maxgap and if we had started with the rect with  $Q$  on the bottom side at the  $k^{th}$  event point we would have got a better rectangle. Thus  $A'B'C'D'$  is the best rect. at the  $k+1^{th}$  event point since  $ABCD$  is the best rect. at  $k^{th}$  event point. In a similar way it can be proved that if  $RR$  occurs in the non-maxgap side then also it gives the best rect. at  $k+1^{th}$  event point.

Next we prove for the case when the  $k+1^{th}$  event point is shift maxgap.

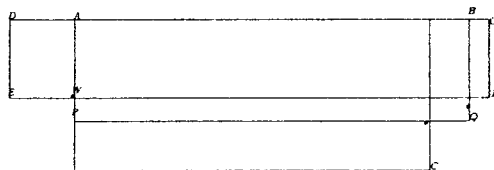


Figure 8:

Let  $APQB$  be the rect. as shown in the fig obtained after the shift maxgap event. The maxgap was on the left side and at  $k+1^{th}$  event point this has become equal to the right side gap. Lets consider a better rect  $DEFG$ . If we consider  $DEFG$  to be the desired rect then shifting  $DC$  to  $Ef$  s.t  $W$  comes on the bottom side should not change the value of the maxgap. By our assumption left side is the maxgap side. Since no event specifically shift maxgap event occurs between the  $k^{th}$  and  $k+1^{th}$  event we can say left side was the maxgap side at the  $k^{th}$  instant. We rotate back  $DEFG$  to the orientation of the  $k^{th}$  rect. We can also say that the gap on the bottom side of  $DEFG$  did not exceed the maxgap as no other events occurred in between. This implies that we could have taken  $DEFG$  'rotated backward' as the desired rect instead of  $ABCD$  'rotated backward'. But this contradicts our induction hypothesis that we get the best rect. at the  $k^{th}$  instant. Thus  $DEFG$  cannot be the best rect at the  $k+1^{th}$  event which in turn proves that since  $ABCD$  'rotated backward' is the best rect in the  $k^{th}$  instant hence  $APQB$  is the best rect at  $k+1^{th}$  instant.

Now we prove that seeing only the rect at the event points suffice. For this if we can prove that the maxgap is monotonic between any two event points or don't have a minima we are done. Suppose while the execution of the algorithm the maxgap at first decrease and then increases in between two event points. Since in the continuous domain the distance between two parallel lines passing through two fixed points can't decrease and then increase, then the

situation stated above can only happen in the following two cases(ref theorem1):

(i)if the point pairing changes.

(ii)if the quadruple changes.

In the continuous domain the point pairing changes iff there is a change in the maxgap. But if there is a hift in the maxgap its marked by an event. The quadruple changes only at the event points of the algorithm. The curve between any two event points thus has no minima. Now since the algorithm computes the best rect. at every orientation in the continuous domain while the rect is rotated in CCW the event points suffice.



## 7 Complexity of the Algorithm

**Theorem 2.** *The orthogonal distance pair event occurs  $O(n)$  times in the continuous domain.*

*Proof.* Since the point pairing does not change in the continuous domain hence the event occurs at most  $2n$  times.  $\square$

**Theorem 3.** *If the maxgap side (i.e the side contains the same point) does not change then its sufficient to consider only the last shift maxgap event before the point on the maxgap side leaves the rectangle.*

*Proof.* Let there be  $k$ -successive shift maxgap events. If we can prove that the maxgap side remains same for the  $k$ -successive event points then we are done. This is because of the following fact. The distance between two parallel lines through two fixed points can't decrease and then increase. Suppose in some intermediate stage the algorithm changes the maxgap side. This can only happen if the point on the left side is changed. This can be proved by considering the shift maxgap event. But the point on the maxgap side can change only if orthogonal distance pair event occurs through the point on the left side. This proves that if the side changes then the successive maxgap events are not actually successive which is exactly the contrapositive of what we were trying to prove.  $\square$

In the algorithm for arbitrary orientation, the step which should be taken for the above theorem is not mentioned (in Step 6). But while analysing the algorithm we assume that the corresponding change has been included. Now to analyse the complexity we first observe that orthogonal distance pair event and RR event occurs  $n^2$  times. The shift maxgap event occurs in between the two above events. The problem crops up if we need to consider successive shift maxgap event. But with help of theorem 3 we can avoid this problem. Using proper data structure the cost of each event is  $\log n$ . So for a the computation takes  $O(n^2 \log n)$ . But in order to implement the property as in Theorem 6 we need an  $O(n^2)$  preprocessing time in the continuous domain. There are  $n$  continuous domain in the worst case. Hence the worst case time complexity is  $O(n^3)$  since its dominated by the preprocessing time. The space used for angular sweep is  $O(n^2)$ . Calculation for the events other than RR can be made as the algorithm progresses. Hence the overall space complexity is  $O(n^2)$ .

## 8 Bibliography

- [1]D. T. Lee, Y. T. Ching: The Power of Geometric Duality Revisited. *Inf. Process. Lett.* 21(3): 117-122 (1985)
- [2]Gill Barequet, Amy J. Briggs, Matthew Dickerson, Michael T. Goodrich: Offset-polygon annulus placement problems. *Comput. Geom.* 11(3-4): 125-141 (1998)
- [3]Mark de Berg, Prosenjit Bose, David Bremner, Suneeta Ramaswami, Gordon T. Wilfong: Computing constrained minimum-width annuli of point sets. *Computer-Aided Design* 30(4): 267-275 (1998)
- [4]Prosenjit Bose, Luc Devroye: Intersections with random geometric objects. *Comput. Geom.* 10(3): 139-154 (1998)
- [5]Pankaj K. Agarwal, Boris Aronov, Sariel Har-Peled, Micha Sharir: Approximation Algorithms for Minimum-Width Annuli and Shells. *Discrete and Computational Geometry* 24(4): 687-705 (2000)
- [6]Olga Gluchshenko, Horst W. Hamacher, Arie Tamir: An optimal  $O(n \log n)$  algorithm for finding an enclosing planar rectilinear annulus of minimum width. *Oper. Res. Lett.* 37(3): 168-170 (2009)
- [7]Chandan Saha, Sandip Das: Covering a set of points in a plane using two parallel rectangles. *Inf. Process. Lett.* 109(16): 907-912 (2009)
- [8]Jeet Chaudhuri, Subhas C. Nandy, Sandip Das: Largest empty rectangle among a point set. *J. Algorithms* 46(1): 54-78 (2003)