Covering Point Sets with Two Convex Objects

J. Miguel Díaz-Báñez* Carlos Seara† J. Antoni Sellarès‡ Jorge Urrutia§ Inma Ventura¶

Abstract

Let P_{2n} be a point set in the plane with n red and n blue points. Let C_R and C_B (S_R and S_B) respectively be red and blue colored and disjoint disks (axis-parallel squares). In this paper we prove the following results. Finding the positions for C_R and C_B that maximizes the number of red points covered by C_R plus the number of blue points covered by C_B can be done in $O(n^3 \log n)$ time. Finding two axis-parallel unit-squares with disjoint interiors that maximizes the sum of the red points covered by S_R plus the number of blue points covered by S_B can be done in $O(n^2)$ time.

1 Introduction

Consider a point set P_{2n} with n *red* and n *blue* points, and a red and a blue coin (not necessarily of the same size) denoted by C_B and C_R respectively. In this paper we study the following problem. Place C_R and C_B on the plane such that the number of red points covered by C_R plus the number of blue points covered by C_B is maximized. We allow C_B and C_R to cover some red (resp. blue) points, but require them to have disjoint interiors. We also consider the problem of finding two isothetic squares S_B and S_R of given sizes with disjoint interiors such that the number of red points covered by S_R plus the number of blue points covered by S_B is maximized. We consider also a similar problem of finding two isothetic squares S_B and S_R of given sizes with disjoint interiors such that the number of red points covered by S_R plus the number of blue points covered by S_B is maximized. In what follows, and to avoid repetitions, a bi-colored point set is a point set such that all its elements are colored red or blue.

In [10] the following problem is considered: given a bi-colored point set P_n on the plane, find an axis-parallel box that does not contain blue points and maximizes the number of red points it covers. An $O(n^2 \log n)$ time algorithm is presented for solving this problem.

In [1] a similar problem is studied: given a bi-colored point set P_n in $\mathbb{R}^d$, find a ball that maximizes the number of red points it contains without containing any blue point in its interior. For $d = 2$, this problem is solved in $O(n^2 \log n)$ time. Monochromatic variants of these problems were studied in [5, 8].

In Pattern Recognition and Classification problems, a natural method to select prototypes to represent a class is to performer cluster analysis on the training data [7]. The clustering can be obtained by using simple geometric shapes such as circles or boxes. Recent papers deal with the *maximum bi-chromatic covering problem.* The problem is the following: given a bi-chromatic point set, the goal is to maximize the number of points of a given color, say red, covered by a given object while avoiding points of the second color. In [1] and [13] circles and boxes respectively are considered for the classification.

In some cases, requiring that the red (resp. blue) covering ball avoid blue (resp. red) points may is some cases lead to invalid classifications and make the results obtained useless. This scenario can arise when facilities interfere with each other, but their possible users are scattered randomly on the plane. A possible solution to this problem is to allow blue points in a red ball and red points in a blue ball. In this paper we introduce this criterion with fixed sizes for the circles or boxes.

A central problem in facility location is to find the best location for a facility to serve a set of users. Maximal covering disk problems are often the main criteria used in facility location where in a natural way a point is served by a facility if it is within a given *distance* from it. In some cases the metric used is the Euclidean distance, and in others the l_∞ (box) metric. Many of these problems also arise in operations research [11]. The problem for locating a maximum covering circle of a given size in a monochromatic set was studied in [6].

*Dpto. Matemática Aplicada II, Universidad de Sevilla, Spain, dbanez@us.es. Partially supported by grant BFM2003-04062.

†Dpt. Matemàtica Aplicada II, Universitat Politècnica de Catalunya, Spain, carlos.seara@upc.edu. Partially supported by projects DGES-SEUID-PB98-0933, MCYT-FEDER-BFM2002-0557 and Gen-Cat-2001SGR00224.

‡Institut d'Informàtica i Aplicacions, Universitat de Girona, Spain, sellares@ima.udg.es. Partially supported by project TIN2004-08065-C02-02.

§Instituto de Matemáticas, Universidad Nacional Autónoma de México, urrutia@matem.unam.mx. Supported by CONACYT of México, Proyecto 37540-A.

¶Dpto. de Matemáticas, Universidad de Huelva, Spain, iventura@us.es. Partially supported by grant BFM2003-04062.

2 Two disjoint disks

Let P_{2n} be a planar bi-colored point set with n blue and n red points. In this section we study the problem of placing two disks, one red disk C_R and one blue disk C_B in such a way that their interiors are disjoint, and the number of red points covered by the red disk plus the number of blue points covered by the blue disk is maximized. In this section we prove:

Theorem 1 *Finding the positions for C_R and C_B such that the number of red points covered by C_R plus the number of blue points covered by C_B is maximized can be done in $O(n^3 \log n)$ time.*

To facilitate our presentation we will assume that the disks are the same size, i.e., equal radii, say r. Assume first that C_R and C_B are placed on the plane such that C_R (resp. C_B) covers a subset R_1 (resp. B_1) of red (resp. blue) points of P_{2n}.

In what follows, when we say that a disk C_R or C_B has some points on its boundary, we shall assume that those points have the same color as the disk. With this in mind, we state the following result (see Figure 1):

Lemma 2 *The disks C_R and C_B can be moved to a new position in the plane such that either: i) one of the disks has at least two points on its boundary and the other one has at least one point on its boundary, or ii) each disk has only one point on its boundary and these points are collinear with the centers of the disks.*

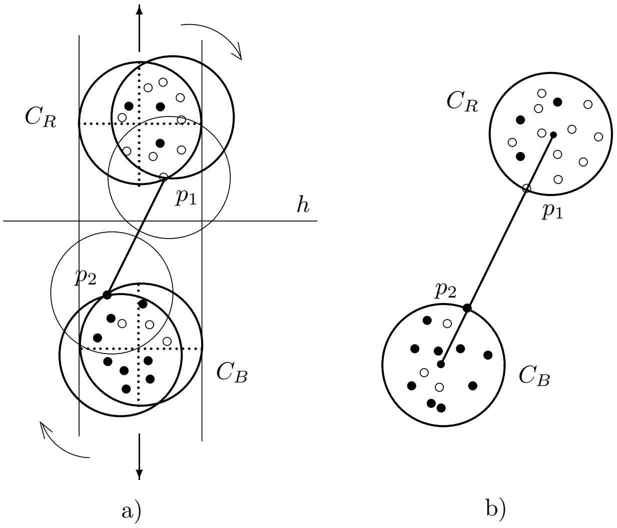

Figure 1: Points in the boundaries of the disks.

Constructing an arrangement of red and blue circles

Let P_n be a bi-colored point set on the plane. For each red point r_i (resp. blue point b_j) we consider the red (resp. blue) circle with center in r_i (resp. b_j) and radius r. Let $\mathcal{A}$ be the arrangement of the $2n$ red and blue circles. As the circles are closed Jordan curves such that any of them intersect in at most two points, the following results can be applied to the arrangement $\mathcal{A}$.

Theorem 3 [12, 4, 5] *The following statements hold for $\mathcal{A}$: 1) The combinatorial complexity of a single face in $\mathcal{A}$ is at most $\lambda_2(n) = O(n)$. 2) The combinatorial complexity of the zone of a circle in $\mathcal{A}$ is $O(n\alpha(n))$. 3) $\mathcal{A}$ can be computed in $O(n^2)$ time or using the sweep-line algorithm of Bentley and Ottmann [3] in $O(n^2 \log n)$ time.*

Notice that a circle can contribute more than one arc to a cell of $\mathcal{A}$. Associate in $\mathcal{A}$ the pair of numbers (nr_j, nb_j) to each cell j such that nr_j (resp. nb_j) is the number of red (resp. blue) *convex arcs* which belong to the boundary of j. Clearly a circle of radius r with center at any point x in a cell j covers exactly nr_j red points and nb_j blue points (Figure 2). Using a line-sweep algorithm, we can compute the pair (nr_j, nb_j) associated to each cell j in $O(n^2 \log n)$ time.

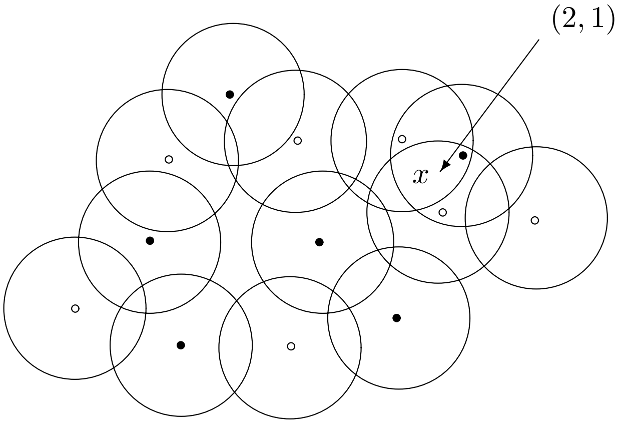

Figure 2: Arrangement $\mathcal{A}$.

Pre-processing a circle-arrangement

For each point $p_k \in P_{2n}$, the locus a_k of centers of circles passing through p_k with radius r defines a circle with center at p_k and radius r.

By Theorem 3, each a_k is split into at most $O(n)$ arcs, each associated to a cell j of $\mathcal{A}$. We label each of these sub-arcs with the pair (nr_j, nb_j) of its associated cell. We store this information for a_k in a *segment tree* [2] so that we can obtain for any sub-arc SC of a_k the sub-arc of $\mathcal{A}$ contained in a_k that intersects SC and maximizes the number nr_i or nb_j, which is the maximum number of red or blue points covered by a circle of radius r with center in a_k, in $O(\log n)$ time. For each circle a_k, the segment tree uses $O(n \log n)$ space and can be built in $O(n \log n)$

time. Doing this at a pre-process stage and storing the information for all the circles of $\mathcal{A}$ takes $O(n^2 \log n)$ time and $O(n^2 \log n)$ space.

The algorithm

We now show how to compute the centers of the two disks C_B and C_R such that the sum of the number of red points and blue points covered respectively by C_B and C_R is maximized.

TWO-DISKS-EQUAL-RADII-ALGORITHM

1. Compute $\mathcal{A}$. For each cell j of $\mathcal{A}$ compute the pair (nr_j, nb_j) and the segment trees for the circles a_k above.

2. By Lemma 2 we can find C_B and C_R as follows:

   (a) For any red point p_i and any blue point p_j compute the circles $C_{R(i)}$ and $C_{B(j)}$ (having p_i and p_j respectively on their boundaries) of radius r with centers on the line joining p_i to p_j, and at distance $|p_i - p_j| + 2r$. In $O(n)$ time, compute the number or red and blue points contained in $C_{R(i)}$ and $C_{B(j)}$ respectively. Keep the pair that maximizes the number of red points in C_{R_i} plus the number of blue points in $C_{B(j)}$.

   (b) For any circle $C_{R(i,j)}$ of radius r and center $c(i,j)$ that passes through a pair of red points p_i, p_j in P_{2n}, do the following: for any blue point p_k at distance greater than or equal to r from $c(i,j)$, compute in constant time, the sub-arc a'_k of the circle a_k consisting of all the points of a_k at distance greater than or equal to r from $c(i,j)$. Using the information stored in the *segment tree* of a_k in $O(\log n)$ time, get the sub-arc of a_k in $\mathcal{A}$ that intersects a'_k with the largest nb_i, which corresponds to a disk C_B with radius r and center on a'_k that maximizes the number of blue points it contains. Compute the numbers of red and blue points covered by $C_{R(i,j)}$ and C_B respectively, keeping the best solution, see Figure 3. Do the same for circles of radius r that pass through two blue points, and keep the best overall solution.

Clearly the best of the solutions obtained in 2(a) and 2(b) solves our problem.

Analysis of the algorithm. Step 1 of the algorithm can be done in $O(n^2 \log n)$ time, step 2(a) can be done in $O(n^3)$ time. Finally, step 2(b) takes $O(n \log n)$ time per pair of points, i.e., $O(n^3 \log n)$ total time, assuming a pre-process of $O(n^2 \log n)$ time for computing the *segment trees*. Thus the algorithm runs in $O(n^3 \log n)$ time. It is easy to see that with some slight modifications we can drop the condition that the two circles have the same size. This concludes the proof of Theorem 1.

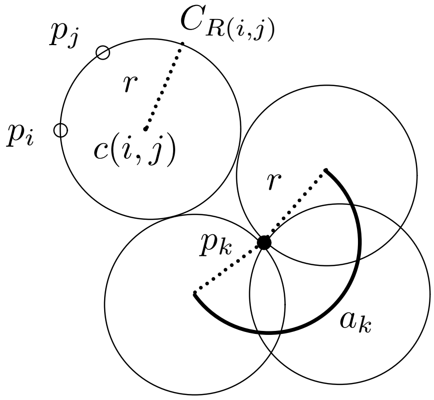

Figure 3: Locus a_k of centers of circles C_2.

3 Two disjoint axis-parallel squares

In this section we study a similar problem in which instead of finding two circles C_R and C_B, we now want two isothetic squares S_B and S_R of given sizes with disjoint interiors such that the number of red points covered by S_R plus the number of blue points covered by S_B is maximized. As in the previous section, we restrict ourselves to squares of the same size, i.e. unit squares. This condition can be lifted easily, leaving the results unchanged. It is easy to see that a similar result is valid for two disjoint quadrilaterals with pairwise parallel sides with given directions. Assume that the red points of P_{2n} are labeled $r_1, \ldots, r_n$ such that for $i < j$ the x-coordinate x_i of r_i is smaller than the x-coordinate of r_j. Assume a similar labeling for the blue points $b_1, \ldots, b_n$ of P_{2n}.

3.1 Two disjoint axis-parallel unit squares

Let S_R and S_B be the elements of an optimal solution. Observe first that since S_R and S_B are isothetic, there is a horizontal or vertical line l that separates them. Assume that l is vertical, and that S_R is to the left and S_B to the right of l. Slide S_R to the left until its right side meets a red point. Observe that since S_R is in an optimal solution, the number of red points it covers does not change. Thus we can assume that S_R contains a red point on its right side. Similarly we can assume that S_B contains a blue point on its left edge. Using these observations, we now outline a process to find an optimal pair S_R and S_B in $O(n^2)$ time. First order the red and the blue points of P_{2n} according to its y-coordinate.

For each red point $r_i \in P_{2n}$, find the unit square containing r_i on its right side which contains the maximum number of red points in P_{2n}. This can be done in linear time as follows.

First find the set R_i of red points to the left of r_i within the vertical strip ST of unit width bounded to its right by l. Find the unit square S contained in ST such that r_i is on its top side, and count the number of red points it contains. Now slide S from bottom to top (keeping it within ST) until it reaches a position in which r_i lies on its bottom edge. While sliding S, keep track of the number of red points it contains as they enter or leave S. This can be done easily in linear time using the order of the red points of P_{2n} according to its y-coordinate. At the end of this process, we have identified the square containing r_i on its right side that contains the maximum number m_{r_i} of red points.

In a similar way we can process the blue points of P_{2n} such that for each blue point b_j we find a square containing b_j on its left edge, and containing the maximum number m_{b_j} of blue points.

For each r_i let $mr(i) = \max\{m_{r_k}; k \leq i\}$. The set $\{mr(i) : i = 1, \dots, n\}$ can be found in linear time with a single scan of the set of red points from left to right. In a similar way, we can find the set of values $\{mb(j) : j = 1, \dots, n\}$ such that $mb(j) = \max\{m_{b_k} : k \geq j\}$.

Using the lists $mr(1), \dots, mr(n)$, $mb(1), \dots, mb(n)$ and traversing all the points (blue and red) of P_{2n} from left to right we can find an optimal pair of squares S_R and S_B in linear time. For brevity, the details are left to the reader. Full details appear in a longer version of this paper. Proceeding in a similar way when S_B is to the left of l and S_R to its right, or when a horizontal line separates S_R and S_B we obtain the following theorem.

Theorem 4 *Finding two axis-parallel unit-squares with disjoint interiors such that the sum of the red points covered by S_R plus the number of blue points covered by S_B is maximized can be done in $O(n^2)$ time.*

Theorem 5 *The two axis-parallel unit-squares with disjoint interiors such that the sum of the the red points covered by S_R plus the number of blue points covered by S_B is maximized requires $\Omega(n \log n)$ time under the algebraic computation tree model.*

The lower bound can be proved by a reduction to the uniform gap problem [9].

To close we would like to mention that using similar techniques as those in this section, we can obtain the following result:

Theorem 6 *Let P_n be a set of n points in the plane. The problem of finding three axis-parallel rectangles (not necessarily of the same size) with disjoint interiors such that the number of points of P_n covered by them is maximized can be solved in $O(n^3)$ time.*

Acknowledgments

These problems were posed and partially solved during the *Second Spanish Workshop on Geometric Optimization*, July 5–10, 2004, El Rocio, Huelva, Spain.

References

[1] B. Aronov, S. Har-Peled. On approximating the depth and related problems. *Proceedings 16th Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA 2005), 2005.

[2] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf. *Computational Geometry, Algorithms and Applications.* Springer, 1997.

[3] J. L. Bentley, T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.* C-28, 1979, pages 643–647.

[4] A. Calatayud. Problemas geométricos de localización. *PhD Thesis, Universidad Politécnica de Madrid*, 2004.

[5] B. Chazelle, D. T. Lee. On a circle placement problem. *Computing*, Vol. 36, 1986, pages 1–16.

[6] Z. Drezner. On a modified one-center model. *Management Science*, 27, 1981, pages 848–851.

[7] R. Duda, P. Hart, D. Stork. *Pattern Classification. John Wiley and Sons, Inc.*, New York, 2001.

[8] M. J. Katz, K. Kedem, M. Segal. Improved algorithms for placing undesirable facilities. *Computers & Operations Research*, 29, 2002, pages 1859–1872.

[9] D. T. Lee, Y. F. Wu. Geometric complexity of some location problems. *Algorithmica*, 1, 1986, pages 193–211.

[10] Y. Liu, M. Nadiak. Planar case of the maximum box and related problems. In *Proceeding of the Canadian Conference on Computational Geometry*, CCCG'03, Halifax, Nova Scotia, 2003, pages 11–13.

[11] R. F. Love, J. G. Morris, G. O. Wesolowsky. *Facilities Location. North-Holland*, Amsterdam, 1988, chapter 3.3, pages 51–6.

[12] M. Sharir, P. K. Agarwal. *Davenport-Schinzel sequences and their geometric applications. Cambridge University Press*, 1995.

[13] M. Segal. Planar maximum box problem. *Journal of Mathematical Modelling and Algorithms*, 3, 2004, pages 31–38.