IMFM
Institute of Mathematics, Physics and Mechanics
Jadranska 19, 1000 Ljubljana, Slovenia

Preprint series
Vol. 49 (2011), 1138
ISSN 2232-2094

COVERING A BICHROMATIC POINT SET WITH TWO DISJOINT MONOCHROMATIC DISKS

S. Cabello J.M. Díaz-Báñez
P. Pérez-Lantero

Ljubljana, February 23, 2011

Covering a Bichromatic Point Set with Two Disjoint Monochromatic Disks

S. Cabello [*] J.M. Díaz-Báñez [†] P. Pérez-Lantero [‡]

February 19, 2011

Abstract

Let P be a set of n points in the plane in general position such that its elements are colored red or blue. We study the problem of finding two disjoint disks D_r and D_b such that D_r covers only red points, D_b covers only blue points, and the number of elements of P contained in $D_r \cup D_b$ is maximized. We prove that this problem can be solved in $O(n^{11/3} \operatorname{polylog} n)$ time. We also present a randomized algorithm that with high probability returns a $(1-\varepsilon)$-approximation to the optimal solution in $O(n^{4/3}\varepsilon^{-6} \operatorname{polylog} n)$ time.

1 Introduction

In data mining and classification problems, a natural method for analyzing data is to select prototypes representing different data classes. A standard technique for achieving this is to perform cluster analysis on the training data [DHS01, HSM01]. The clusters can be obtained by using simple geometric shapes such as disks or boxes. Aronov and Har-Peled [AHP08], Eckstein et al. [EHL$^+$02], and Liu et al. [LN03] considered disks and axis-aligned boxes for the selection problem. Aronov and Har-Peled [AHP08] studied the following problem: *Given a bicolored set of n points in the plane, find a disk that contains the maximum number of red points without containing any blue point.* They propose an algorithm to solve this problem optimally in $O(n^2 \log n)$ time, and also provide a $(1-\varepsilon)$-approximation algorithm that needs near-linear time. This type of classification is *asymmetric* in the way red and blue points are treated. In this paper, we consider a *symmetric two-class* version, where we want to find a witness set for each color. We next formalize the problem.

Let P be a set of n points in the plane such that its elements are colored red or blue. Denote by R (resp. B) the set of red (resp. blue) elements of P. We say that $Y \subset \mathbb{R}^2$ is *red* (resp. *blue*) if Y contains only red (resp. blue) elements of P, and Y is monochromatic if it is either red or blue. In this paper we study the following problem:

[*] Department of Mathematics, FMF, University of Ljubljana, `sergio.cabello@fmf.uni-lj.si`. Partially supported by the Slovenian Research Agency, program P1-0297

[†] Departamento Matemática Aplicada II, Universidad de Sevilla, `dbanez@us.es`. Partially supported by Grant MEC MTM2009-08625

[‡] Departamento Matemática Aplicada II, Universidad de Sevilla, `pplantero@us.es`. Partially supported by Grant MAEC-AECI and MEC MTM2009-08625

> **The Two Disjoint Disks problem (2DD-problem)**: find a red disk D_r and a blue disk D_b such that D_r and D_b are disjoint and $|D_r \cap R| + |D_b \cap B|$ is maximized.

We provide algorithms to solve the 2DD-problem optimally and approximately. It is easy to see that the 2DD-problem can be solved optimally in $O(n^5)$ time. We reduce this running time to $O(n^{11/3} \operatorname{polylog} n)$. This result is described in Section 2. We also provide a randomized approximation scheme that, with probability at least $1 - O(1/n)$, returns a $(1 - \varepsilon)$-approximation to the optimal solution in $O(n^{4/3}\varepsilon^{-6} \operatorname{polylog} n)$ time. Under the assumption that at least a constant fraction of the points is covered in the optimal solution, the $(1 - \varepsilon)$-approximation can be obtained in $O(n\varepsilon^{-13} \operatorname{polylog} n)$ time. This approximation algorithm is described in Section 3. In our algorithms, we did not try to improve the exponents of ε or $\log n$.

We next discuss variants of the 2DD-problem that have been considered previously. If in the 2DD-problem we do not restrict D_r and D_b to be disjoint, then we can solve the problem considering each color separately. First, we find the disk covering the maximum number of red points and no blue point, and after that, the disk that contains the maximum number of blue points and no red point. These two problems can be solved optimally in $O(n^2 \log n)$ time, or a $(1-\varepsilon)$-approximation can be obtained in near-linear time with high probability [AHP08].

Another variant of the 2DD-problem that permits intersection of the disks but penalizes it is the problem of finding disks D_r and D_b that maximize $|(D_r \setminus D_b) \cap R| + |(D_b \setminus D_r) \cap B|$. This criterion is considered for axis-aligned boxes in [CDBPL$^+$09]. An optimal solution for this variant can be found in $O(n^2)$ time using a key observation and known approaches. Namely, suppose that (D_r, D_b) is an optimal solution, and denote by ℓ the radical axis of D_r and D_b. Let π_1 and π_2 be the open half-planes bounded by ℓ such that $((D_r \setminus D_b) \cap R) \subset \pi_1$ and $((D_b \setminus D_r) \cap B) \subset \pi_2$. Note that $(D_r \setminus D_b) \cap R = \pi_1 \cap R$ and $(D_b \setminus D_r) \cap R = \pi_2 \cap B$ because the pair (D_r, D_b) is optimal. Since both half-planes bounded by the line ℓ are disks with infinite radii, the problem reduces to finding a line ℓ such that the number of red points to one side of ℓ plus the number of blue points to the other side is maximized. This latter problem is known as the *Weak Separation Problem* [Cha05, ERvK96, Hou93] and can be solved in $O(n^2)$ time in the worst case [Hou93]. Moreover, it was proven in [BDBL$^+$09] that the Weak Separation Problem is 3SUM-hard [GO95].

Another variant of the 2DD-problem is the problem of finding two unit disks D_r and D_b with disjoint interiors, but not necessarily monochromatic, such that $|D_r \cap R| + |D_b \cap B|$ is maximized. Note that in this variant there are two differences with the 2DD-problem: the disks are unitary and do not need to be monochromatic. This variant was considered in [CDBS$^+$08], where an $O(n^{8/3} \log^2 n)$-time algorithm is described.

Notation. Given two points p and q we denote: by $\overline{pq}$ the straight line segment joining p and q, by $\ell(p, q)$ the straight line containing both p and q, by $\operatorname{bis}(p, q)$ the line perpendicular to $\overline{pq}$ passing through the midpoint of $\overline{pq}$ (i.e. the bisector of p and q), and by $D(p, q)$ the disk centered at p with radius equal to the length of $\overline{pq}$. Given a region $S \subset \mathbb{R}^2$, let δS denote the boundary of S.

General position. We assume general position, that is, there are no four cocircular points in P, neither three collinear points. We relax the definition of our problem by allowing the boundary of the red disk (resp. blue disk) to contain one blue point (resp. one red point). A solution to the relaxed problem induces a solution to the original one by shrinking the disks slightly.

2 An exact algorithm

In this section we provide an exact $O(n^{11/3} \operatorname{polylog} n)$-time algorithm to solve the 2DD-problem. First, we will show that we only need to consider a certain type of solutions, thus obtaining a discretization of the problem. Secondly, we will consider a decision version of the problem, where we want to decide if there exists a solution covering a prescribed number of points of each color. Finally, we will discuss how to find an optimal solution to the 2DD-problem.

2.1 Discretization

It is not hard to see that a simple discretization of our problem in which the boundary of each disk contains three points of P, or two diametrically opposed points, is not possible. In order to obtain an appropriate discretization we will use the following lemmas.

Lemma 2.1 *If the points p, q, o, p', q', and o' are such that $o \in \operatorname{bis}(p,q)$, $o' \in \operatorname{bis}(p',q')$, and $D(o,p) \cap D(o',p') = \emptyset$, then both o and o' can be moved simultaneously along $\operatorname{bis}(p,q)$ and $\operatorname{bis}(p',q')$, respectively, so that at every moment $D(o,p) \cap D(o',p') = \emptyset$, until o or o' reaches infinity.*

Proof. Notice that both p' and q' are in the same half-plane bounded by $\ell(p,q)$, or both p and q are in the same half-plane bounded by $\ell(p',q')$. Assume w.l.o.g. the former case. We prove now that such a movement exists so that o reaches infinity. Denote by π_1 and π_2 the open half-planes bounded by $\ell(p,q)$ and suppose w.l.o.g. that $p', q' \in \pi_1$. Refer to Figure 1 a). Let h_o be the half-line starting at o such that $h_o \subset \operatorname{bis}(p,q)$ and $h_o \cap \pi_2$ is unbounded. Analogously define $h_{o'}$ as the half-line starting at o' such that $h_{o'} \subset \operatorname{bis}(p',q')$ and $h_{o'} \cap \pi_1$ is unbounded. If $D(o',p') \subset \pi_1$ then the result follows by moving only o through h_o in direction to infinity. Otherwise, let $\varepsilon > 0$ be a small enough value and $u(o') \in h_o$ be the furthest point from o such that the disk $D(u(o'),p)$ is exterior tangent to $D(o',p')$ (see Figure 1 b)). We first move o in direction to $u(o')$ until the distance between o and $u(o')$ is equal to ε. After that we simultaneously move o' through $h_{o'}$ in direction to infinity and o through h_o in such a way that o stays at distance ε from $u(o')$. We stop this simultaneous movement when $D(o',p') \subset \pi_1$. At this moment the center o can be moved to infinity. □

Lemma 2.2 *Let D_1 be a disk with center o'. If the points p, q, and o, are such that $o \in \operatorname{bis}(p,q)$ and $D(o,p) \cap D_1 = \emptyset$, then o can be moved along $\operatorname{bis}(p,q)$, in such a way that $D(o,p) \cap D_1 = \emptyset$ at every moment, until one of the next conditions is satisfied: (i) o reaches infinity, or (ii) the segment $\overline{oo'}$ contains p or q.*

If D_1 is a half-plane, then o can be moved to the line perpendicular to δD_1 passing through p or q.

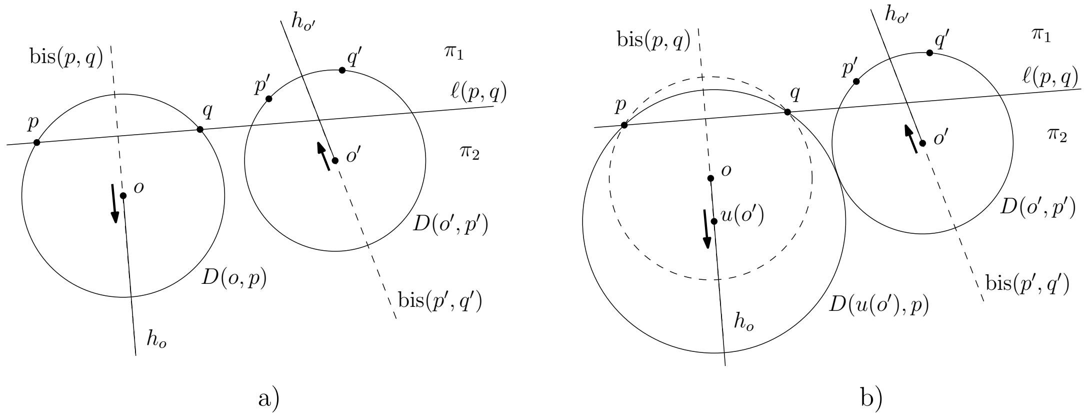

Figure 1: Proof of Lemma 2.1.

Proof. The set of centers $o \in \text{bis}(p, q)$ such that $D(o, p) \cap D_1 = \emptyset$ is a connected interval of $\text{bis}(p, q)$. If this interval is unbounded, then we can move o to infinity, and condition (i) is satisfied. If the interval is bounded, then the line $\ell(p, q)$ intersects D_1. The extremes o_1 and o_2 of the interval correspond to the two points such that the disks $D(o_1, p)$ and $D(o_2, p)$ are exterior tangent to D_1. Consider w.l.o.g. that p is closer to o' than q. See Figure 2 a). Denote by o_3 the intersection point of the line $\ell(o_1, o_2)$ with $\ell(p, o')$. Since $\ell(p, q)$ intersects D_1, the point p lies in the segment $\overline{o'o_3}$, and thus the disk $D(o_3, p)$ does not intersect D_1. This implies that o_3 is on the segment $\overline{o_1o_2}$, and condition (ii) can be satisfied by moving o to o_3.

Consider now the case where D_1 is a half-plane, and assume w.l.o.g. that p is closer to δD_1 than q. See Figure 2 b) The set of centers $o \in \text{bis}(p, q)$ such that $D(o, p) \cap D_1 = \emptyset$ is bounded, and thus the second case of the previous analysis applies. □

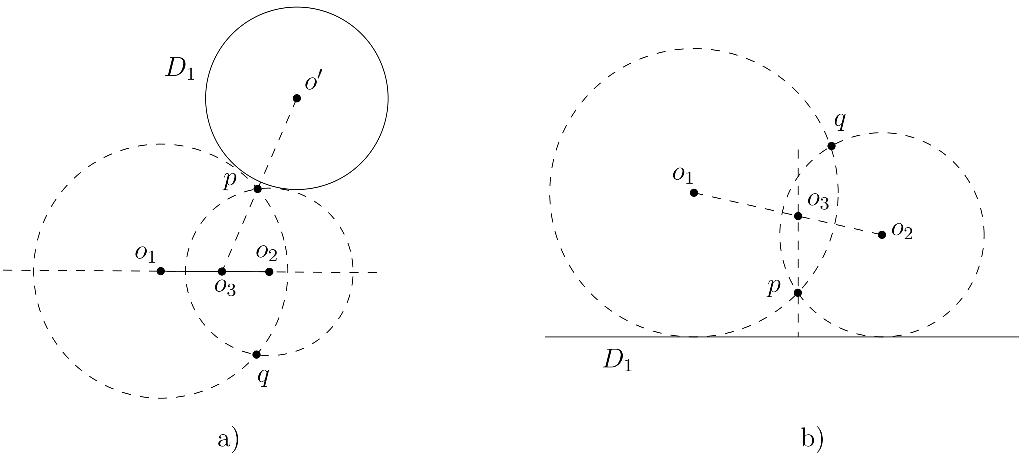

Figure 2: Proof of Lemma 2.2. a) Case when D_1 is a disk. b) Case when D_1 is a half-plane.

Solutions (D_r, D_b) such that $|D_r \cap R| = 1$ can be easily found as follows. We find a blue disk D_b that contains the maximum number of blue points by using [AHP08], and set D_r to be a degenerate disk containing a single red point. Similarly, we can treat the case $|D_b \cap B| = 1$. Thus we assume from now on that $|D_r \cap R| \geq 2$ and $|D_b \cap B| \geq 2$.

We now proceed with the discretization to the 2DD-problem. Let $\mathbb{D}_R$ be the family of the disks D with red interior such that δD contains exactly three red points, or two red points and one blue point. Let $\mathbb{H}_R$ be the family of the closed red half-planes H such that δH contains two red points. Let the families $\mathbb{D}_B$ and $\mathbb{H}_B$ be defined analogously for the blue color.

Lemma 2.3 *For any feasible solution (D'_r, D'_b) to the 2DD-problem, there exists another feasible solution (D_r, D_b) with $D'_r \cap R \subseteq D_r \cap R$ and $D'_b \cap B \subseteq D_b \cap B$ that satisfies one of the next conditions:*

(*a*) *$D_r \in \mathbb{D}_R \cup \mathbb{H}_R$ and $D_b \in \mathbb{D}_B$.*

(*b*) *$D_r \in \mathbb{D}_R$, δD_b contains two blue points, and one blue point in δD_b is on the segment connecting the center of D_r to the the center of D_b.*

(*c*) *$D_r \in \mathbb{H}_R$, δD_b contains two blue points, and one blue point in δD_b is on the line perpendicular to δD_r passing through the center of D_b.*

(*d*) *the cases symmetric to* (*a*)*-*(*c*) *by exchanging colors.*

Proof. We can shrink D'_r into a disk D''_r contained in D'_r that has two red points on its boundary and satisfies $D''_r \cap R = D'_r \cap R$. An analogous transformation can be done to obtain D''_b from D'_b.

Assume that D''_b is the blue disk centered at o with p and q on its boundary, and that D''_r is the red disk centered at o' with p' and q' on its boundary. Keeping the set of points in the interior of $D(o,p)$ and $D(o',p')$ invariant, we move o and o' as in Lemma 2.1 until $D(o,p)$ belongs to $\mathbb{D}_B \cup \mathbb{H}_B$ or $D(o',p')$ belongs to $\mathbb{D}_R \cup \mathbb{H}_R$. Up to symmetry, we may assume the latter case: $D(o',p') \in \mathbb{D}_R \cup \mathbb{H}_R$. We leave $D_1 = D_r = D(o',p')$ unchanged for the rest of the proof. Keeping the set of points in the interior of $D(o,p)$ unchanged, we can now move o as in Lemma 2.2. If at any moment $D(o,p)$ belongs to $\mathbb{D}_B$ because its boundary touches a third point we obtain a feasible solution satisfying case (*a*). If o reaches infinity (condition (*i*) in Lemma 2.2), then $D(o,p)$ belongs to $\mathbb{H}_B$, and we obtain a solution satisfying case (*d*) symmetric to (*a*). If $D(o',p')$ is a disk and o is moved to satisfy condition (*ii*) in Lemma 2.2, then we obtain case (*b*). If $D(o',p')$ is a half-plane and o is moved as described in Lemma 2.2, then we obtain case (*c*). The result follows. □

2.2 Decision and optimization problem

We now consider the following decision version of the problem for integers i, j, $2 \leq i, j \leq n$.

> (i,j)**-2DD-problem:** find a solution (D_r, D_b) to the 2DD-problem subject to the constraints $|D_r \cap R| \geq i$ and $|D_b \cap B| \geq j$; or report that no such solution exists.

Note that, if there exists a solution to the (i^*, j^*)-2DD-problem, then there is a solution to the (i,j)-2DD-problem whenever $i \leq i^*$ and $j \leq j^*$. Also, when searching for a solution to the (i,j)-2DD-problem, it is enough to restrict the search to pairs of disks that contain exactly i red points and j blue points.

Lemma 2.4 *The (i,j)-2DD-problem can be solved in $O(n^{8/3} \operatorname{polylog} n)$ time.*

Proof. Let $\mathbb{D}_R(i)$ be the set of disks of $\mathbb{D}_R$ containing exactly i red points, and let $\mathbb{H}_R(i)$ be the set of half-planes of $\mathbb{H}_R$ containing exactly i red points. Notice that the centers of the disks of $\mathbb{D}_R(i)$ are vertices of the $(i-2)$th-order or the $(i-1)$th-order Voronoi Diagram of $R \cup B$, while the half-planes in $\mathbb{H}_R(i)$ correspond to some of the unbounded edges in the $(i-1)$th-order Voronoi diagram. Since the kth-order Voronoi diagram has combinatorial complexity $O(k(n-k)) = O(n^2)$ in total [Aur91], the sets $\mathbb{D}_R(i)$ and $\mathbb{H}_R(i)$ have at most $O(n^2)$ elements. Analogous sets $\mathbb{D}_B(j)$ and $\mathbb{H}_B(j)$ can be defined as those covering j blue points.

For every blue point q, let $S_B(q,j)$ be the set of straight-line segments s such that for each point o of s the disk $D(o,q)$ is blue, covers j blue points, and its boundary contains other blue point distinct from q. The set $S_B(q,j)$ is a subset of the edges in the $(j-1)$th-order Voronoi diagram of $R \cup B$.

We next discuss how to construct $\mathbb{D}_R(i)$ and $\mathbb{H}_R(i)$ in $O(n^{8/3})$ time. Using [ABMS98, CSY87], the kth-order Voronoi diagram ($k = i-2, i-1$) for n points can be constructed in $O(n^{2+\delta}) = O(n^{8/3})$ time, for any $\delta > 0$. Within the same time bound, we can attach to each vertex (resp. edge) of the diagram a label telling how many of its $k+2$ (resp. $k+1$) closest neighbours are red and how many are blue.

Both the vertices of the $(i-2)$th-order Voronoi diagram of $R \cup B$ whose i closest neighbours are red, and the vertices of the $(i-1)$th-order Voronoi diagram of $R \cup B$ whose $i+1$ closest neighbours are all red except one that is blue, correspond to the centers of the disks $\mathbb{D}_R(i)$. The infinite edges of the $(i-1)$th-order Voronoi Diagram of $R \cup B$ whose i neighbours are all red correspond to the half-planes $\mathbb{H}_R(i)$.

Similarly, using the $(j-2)$th- and the $(j-1)$th-order Voronoi diagrams of $R \cup B$ we can construct the sets $\mathbb{D}_B(j)$ and $\mathbb{H}_B(j)$. Furthermore, the sets $S_B(q,j)$ for all $q \in B$ can also be constructed using the $(j-1)$th-order Voronoi diagram. In this case, we have to attach to each edge the two furthest points among its j closest neighbours.

Let V_j denote the Furthest Disk Voronoi Diagram of $\mathbb{D}_B(j)$. The *Furthest Disk Voronoi Diagram* is the furthest site Voronoi diagram for a set of disks. Given a point p and a disk (i.e. site) D, the distance function used is $\phi(p,D) = d(p,c(D)) - r(D)$, where d is the Euclidean distance, and $c(D)$ and $r(D)$ are the center and the radius of D, respectively. The Furthest Disk Voronoi Diagram for m disks can be constructed in $O(m \log m)$ time [Rap92], and thus we can construct V_j in $O(n^2 \log n)$ time. We further preprocess V_j in $O(n^2 \log n)$ time in order to support $O(\log n)$-time point location queries [Sno97].

For each blue point q and each edge e of $S_B(q,j)$, consider the double wedge obtained by the union of lines that intersect both q and e, and let $\triangle(q,e)$ be wedge that does not contain e. Thus $\triangle(q,e)$ has q as apex and the prolongation of its sides pass through the endpoints of e. (If e is infinite then a side of $\triangle(q,e)$ is parallel to e.) The set $\triangle(q,e)$ has the following property: a point $x \in \mathbb{R}^2$ is in $\triangle(q,e)$ if and only if the ray emanating from x towards q intersects e after q.

We now discuss how to decide if there exists a solution to the (i,j)-2DD-problem. We only need to restrict our search to solutions satisfying the properties listed in Lemma 2.3. Finding if there is a solution satisfying condition (a) can be done as follows. For each disk D in $\mathbb{D}_R(i)$ centered at o, we make a point location query in V_j with o in order to obtain the furthest disk D' to o in $\mathbb{D}_B(j)$. If the disks D and D' are disjoint, then (D,D') is a solution satisfying condition (a). If they intersect, then there is no solution satisfying condition (a) for the disk D. A similar point location procedure can be used to detect if there is a solution involving a half-plane from $\mathbb{H}_R(i)$: we query V_j with the "center" of the half-plane, which we can take to be any symbolic point far enough in the direction perpendicular to the boundary of the half-plane. We spend here $O(n^2 \log n)$ time in total because we have to make $O(|\mathbb{D}_R(i)| + |\mathbb{H}_R(i)|) = O(n^2)$ point locations in V_j.

Finding if there is a solution satisfying condition (b) can be done as follows. For each disk D in $\mathbb{D}_R(i)$ centered at o, and each blue point q not in δD we want to know if the ray emanating from o towards q crosses a segment of $S_B(q, j)$ after passing through q. If such intersection o' exists, then $(D, D(o', q))$ is a solution satisfying condition (b). And vice versa, each solution satisfying condition (b) corresponds to one such intersection. The intersection exists if and only if o belongs to the union $\bigcup_{e \in S_B(q,j)} \triangle(q, e)$. We can test for this intersection over all blue points q and all disks in $\mathbb{D}_R(i)$ together. We are thus interested in deciding

$$\text{does some center of some disk in } \mathbb{D}_R(i) \text{ lie in } \bigcup_{q \in B} \bigcup_{e \in S_B(q,j)} \triangle(q, e)?$$

which is a problem of deciding if there is any incidence between a set of $O(n^2)$ points and $O(n^2)$ triangles in the plane. This problem can be solved in $O((n^2)^{4/3} \operatorname{polylog} n) = O(n^{8/3} \operatorname{polylog} n)$ time using machinery for simplicial range searching [Cha10, Mat93].

Finding if there is a solution satisfying condition (c) can be done similarly to condition (b), as follows. For each half-plane H of $\mathbb{H}_R(i)$, let $v(H)$ be the vector normal to δH that is included in H. We want to decide if the vector $v(H)$ is contained inside any of the wedges $\bigcup_{e \in S_B(q,j)} \triangle(q, e)$. Again, this can be solved using range searching techniques in $O(n^{8/3} \operatorname{polylog} n)$ time [Cha10, Mat93]. (This can actually be done faster because it is a 1-dimensional range query.) The solutions satisfying condition (d) can be obtained with symmetric algorithms. □

Theorem 2.5 *The 2DD-problem can be solved in $O(n^{11/3} \operatorname{polylog} n)$ time.*

Proof. We find values for i and j so that there exists a solution to the (i, j)-2DD-problem and $i + j$ is maximized. In fact, we find all the Pareto optimal pairs (i, j) for which the (i, j)-2DD-problem has a feasible solution. We start with $i = n - 2$ and $j = 2$. If the (i, j)-2DD-problem has a solution, the value of j is incremented in one. Otherwise, the value of i is decremented by one. This process is continued until $i = 1$ or $j = n - 1$. During this process we keep the best feasible solution found to the (i, j)-2DD-problem. The time complexity is $O(n^{11/3} \operatorname{polylog} n)$ because the decision problem of Lemma 2.4 is invoked $O(n)$ times. □

3 An approximation algorithm

In this section we provide a $(1 - \varepsilon)$-approximation algorithm to the 2DD-problem whose running time is roughly $O(n^{4/3})$, for any constant ε. The algorithm is randomized and successful with high probability. We assume henceforth that $\varepsilon < 1/4$. We first consider an approximate version of the (i, j)-2DD-problem, in a precise sense that will be described below. The main idea for this step is using sampling to approximately count the number of points in any disk. However, care has to be taken to restrict the search to feasible solutions in the original scenario. Thus, we have to work with the original sets of points and the sample sets together. Then, we run a search for the optimal pair (i, j), trying values that are exponentially increasing. We did not attempt to reduce the number of logarithmic factors or the dependency on ε.

3.1 Random samples

Given a finite set A and a parameter $\rho \in [0, 1]$, a ρ-sample of A is a subset of A obtained by taking each element from A with probability ρ, independently. We will use notation like $\widetilde{A}$ to denote such random samples. It is natural to use $\rho^{-1}|D \cap \widetilde{A}|$ as an estimator for $|D \cap A|$. We first describe the properties of this estimator that we will use.

Lemma 3.1 *Let Q be set of n points in the plane and a be a given parameter. Let $\widetilde{Q}$ be a ρ_a-sample of Q, where $\rho_a = \min\{1, ca^{-1}\varepsilon^{-2}\log n\}$ and $c > 0$ is an appropriate constant. With probability at least $1 - 1/n^3$ we have*

(*i*) *for all disks D that contain at least $a/4$ points of Q*

$$(1 - \varepsilon/2)|D \cap Q| \;\le\; \rho_a^{-1} \cdot |D \cap \widetilde{Q}| \;\le\; (1 + \varepsilon/2)|D \cap Q|\,.$$

(*ii*) *for all disks D containing at most $a/4$ points of Q*

$$\rho_a^{-1} \cdot |D \cap \widetilde{Q}| \;\le\; a/2.$$

Proof. This is a standard application of Chernoff bounds [MR96] using the fact that there are at most $O(n^3)$ different sets in the family $\{D \cap Q \mid D \text{ is a disk}\}$. Similar arguments are used in [AHP08, dBCHP09].

If $\rho_a = 1$ then $\widetilde{Q} = Q$ and all inequalities hold. Thus, we consider the case $\rho_a = ca^{-1}\varepsilon^{-2}\log n$. For each point $p \in Q$, let X_p be the indicator random variable, that takes value 1 if the point p is in the sample $\widetilde{Q}$ and value 0 otherwise. Thus, for a disk D we have

$$|D \cap \widetilde{Q}| = \sum_{p \in D \cap Q} X_p$$

whose expected value is

$$\mu_D := \mathbb{E}\left[|D \cap \widetilde{Q}|\right] = \sum_{p \in D \cap Q} \mathbb{E}\left[X_p\right] = |D \cap Q| \cdot \rho_a.$$

Since the variables X_p are independent, we can use Chernoff bounds for $|D \cap \widetilde{Q}|$. We distinguish the two cases of the statement:

- For a fixed disk D with $|D \cap Q| \ge a/4$ we have

$$\begin{aligned}
\Pr\left[\left|\rho_a^{-1} \cdot |D \cap \widetilde{Q}| - |D \cap Q|\right| > (\varepsilon/2) \cdot |D \cap Q|\right] &= \Pr\left[\left||D \cap \widetilde{Q}| - \mu_D\right| > (\varepsilon/2) \cdot \mu_D\right] \\
&\le e^{-\Omega(\mu_D \varepsilon^2)} \\
&= e^{-\Omega(|D \cap Q| \cdot \rho_a \cdot \varepsilon^2)} \\
&= e^{-\Omega(|D \cap Q| \cdot ca^{-1} \log n)} \\
&= n^{-\Omega(c)}\,.
\end{aligned}$$

- For a fixed disk D with $|D \cap Q| \leq a/4$ we consider a disk D' containing D and exactly $a/4$ points of Q. We can then use the previous case for D' and obtain, with probability at least $1 - n^{-\Omega(c)}$,

$$\rho_a^{-1} \cdot |D \cap \widetilde{Q}| \;\leq\; \rho_a^{-1} \cdot |D' \cap \widetilde{Q}| \;\leq\; (1 + \varepsilon/2)|D' \cap Q| \;=\; (1 + \varepsilon/2)a/4 \;\leq\; a/2\,.$$

Therefore, for a fixed disk, the inequalities of the lemma are true with probability at least $1 - n^{-\Omega(c)}$. We choose the constant c so that $n^{-\Omega(c)} \leq n^{-6}$. Since there are at most n^3 distinct sets $D \cap Q$ over all disks D, the result for *all* disks follows from the union bound. □

3.2 Decision problem

We now consider the approach to approximately solve the (i, j)-2DD-problem for given values i, j. Throughout this section, we assume that i and j are fixed.

We construct a ρ_i-sample $\widetilde{R}$ of R and a ρ_j-sample $\widetilde{B}$ of B, where $\rho_a = \min\{1, ca^{-1}\varepsilon^{-2}\log n\}$ as defined in Lemma 3.1. The samples will be used to approximately count the points contained in a pair of disks. However, we cannot just discard the sets R and B, since they are important to restrict the search to feasible solutions in the original scenario. Thus, we have to work simultaneously with the sets R, B, $\widetilde{R}$, and $\widetilde{B}$.

Note that $i \cdot \rho_i = j \cdot \rho_j = c\varepsilon^{-2} \log n$ is the average "target" number of points in each disk, where c is the constant in Lemma 3.1. However, there can be some error in the counting because of using the random sample. We thus set

$$k \;:=\; (1 - \varepsilon/2) \cdot c\varepsilon^{-2} \log n \;=\; (1 - \varepsilon/2) \cdot i \cdot \rho_i \;=\; (1 - \varepsilon/2) \cdot j \cdot \rho_j$$

as the relevant threshold for approximately counting. Consider the following problem:

> **ApproxDecision-2DD-problem:** find two disjoint disks D_r and D_b subject to $D_r \cap B = D_b \cap R = \emptyset$, $|D_r \cap \widetilde{R}| \geq k$, and $|D_b \cap \widetilde{B}| \geq k$; or report that no such solution exists.

Like before, to solve the ApproxDecision-2DD-problem it is enough to restrict the search to pairs of disks (D_r, D_b) with $|D_r \cap \widetilde{R}| = |D_b \cap \widetilde{B}| = k$. We will use the following relation between the (i, j)-2DD-problem and the ApproxDecision-2DD-problem.

Lemma 3.2 *Assume that there is a solution to the (i, j)-2DD-problem. With probability at least $1 - O(1/n^3)$:*

(i) there is a solution to the ApproxDecision-2DD-problem;

(ii) any solution to the ApproxDecision-2DD-problem is a solution to the $((1 - \varepsilon)i, (1 - \varepsilon)j)$-2DD-problem.

Proof. Assume that there is a solution (D_r^*, D_b^*) to the (i, j)-2DD-problem. Because of Lemma 3.1 (i) we have, with probability at least $1 - O(1/n^3)$,

$$|D_r^* \cap \widetilde{R}| \;\geq\; \rho_i \cdot (1 - \varepsilon/2) \cdot |D_r^* \cap R| \;\geq\; \rho_i \cdot (1 - \varepsilon/2) \cdot i \;=\; k\,.$$

A similar argument shows that $|D_b^* \cap \widetilde{B}| \geq k$, and we conclude that (D_r^*, D_b^*) is a solution to the ApproxDecision-2DD-problem. This finishes the proof of part (i).

Consider now any feasible pair of disks (D_r, D_b) that is not a solution to the $((1-\varepsilon)i, (1-\varepsilon)j)$-2DD-problem. We will then show that, with probability at least $1 - O(1/n^3)$, the pair (D_r, D_b) is not a solution to the ApproxDecision-2DD-problem. It follows that, with high probability, only solutions to the $((1-\varepsilon)i, (1-\varepsilon)j)$-2DD-problem can be solutions to the ApproxDecision-2DD-problem.

If (D_r, D_b) is not a solution to the $((1-\varepsilon)i, (1-\varepsilon)j)$-2DD-problem, then $|D_r \cap R| < (1-\varepsilon)i$ or $|D_b \cap B| < (1-\varepsilon)j$. Let us assume the former case; the other case is similar. If $|D_r \cap R| > i/4$, then Lemma 3.1 (i) tells that, with probability at least $1 - O(1/n^3)$,

$$\begin{aligned} |D_r \cap \widetilde{R}| &\leq \rho_i \cdot (1 + \varepsilon/2) \cdot |D_r \cap R| \\ &< \rho_i \cdot (1 + \varepsilon/2) \cdot (1 - \varepsilon)i \\ &\leq \rho_i \cdot (1 - \varepsilon/2)i \\ &= k. \end{aligned}$$

If $|D_r \cap R| \leq i/4$, then Lemma 3.1 (ii) tells that, with probability at least $1 - O(1/n^3)$,

$$|D_r \cap \widetilde{R}| \leq \rho_i \cdot i/2 < k.$$

In either case, $|D_r \cap \widetilde{R}| < k$ with high probability, and (D_r, D_b) cannot be a solution to the ApproxDecision-2DD-problem. This finishes the proof of part (ii). □

To solve the ApproxDecision-2DD-problem we will use a discretization, analogous to Lemma 2.3. Let $\widetilde{\mathbb{D}}_R$ be the family of the disks D with red interior such that δD contains exactly three red points of $\widetilde{R}$, or two red points of $\widetilde{R}$ and one blue point of B; and let $\widetilde{\mathbb{H}}_R$ be the family of the closed red half-planes H such that δH contains two red points of $\widetilde{R}$. Let the families $\widetilde{\mathbb{D}}_B$ and $\widetilde{\mathbb{H}}_B$ be defined analogously with respect to the points $\widetilde{B}$.

Lemma 3.3 *If there exists a solution (D'_r, D'_b) to the ApproxDecision-2DD-problem, then there exists another solution (D_r, D_b) with $D'_r \cap \widetilde{R} \subseteq D_r \cap \widetilde{R}$ and $D'_b \cap \widetilde{B} \subseteq D_b \cap \widetilde{B}$, satisfying one of the next conditions:*

- *(a)* $D_r \in \widetilde{\mathbb{D}}_R \cup \widetilde{\mathbb{H}}_R$ *and* $D_b \in \widetilde{\mathbb{D}}_B$.
- *(b)* $D_r \in \widetilde{\mathbb{D}}_R$, *$\delta D_b$ contains two blue points of $\widetilde{B}$, and one blue point in δD_b is on the segment connecting the center of D_r to the the center of D_b.*
- *(c)* $D_r \in \widetilde{\mathbb{H}}_R$, *$\delta D_b$ contains two blue points of $\widetilde{B}$, and one blue point in δD_b is on the line perpendicular to δD_r passing through the center of D_b.*
- *(d) the cases symmetric to (a)-(c) by exchanging colors.*

Proof. The proof is similar to the proof of Lemma 2.3, but during the transformation we enforce that the red disk D_r keeps both $D_r \cap B$ and the elements of $\widetilde{R}$ contained in the interior of D_r invariant, while the blue disk D_b keeps both $D_b \cap R$ and the elements of $\widetilde{B}$ contained in the interior of D_b invariant. □

Lemma 3.4 *The ApproxDecision-2DD-problem can be solved in $O((k^2n+(kn)^{4/3}) \operatorname{polylog} n)$ time.*

Proof. The approach is very similar to the algorithm in Lemma 2.4, and we will thus skim over some of the details. Let $\widetilde{\mathbb{D}}_R(k)$ be the set of disks of $\widetilde{\mathbb{D}}_R$ containing exactly k red points of $\widetilde{R}$, and let $\widetilde{\mathbb{H}}_R(k)$ be the set of half-planes of $\widetilde{\mathbb{H}}_R$ containing exactly k red points of $\widetilde{R}$. The centers of the disks of $\widetilde{\mathbb{D}}_R(k)$ are vertices of $(k-2)$th-order or the $(k-1)$th-order Voronoi Diagram of $\widetilde{R} \cup B$, while the half-planes in $\widetilde{\mathbb{H}}_R(k)$ correspond to unbounded edges in the $(k-1)$th-order Voronoi diagram. Since the kth-order Voronoi diagram has combinatorial complexity $O(k(n-k)) = O(kn)$ in total [Aur91], the sets $\widetilde{\mathbb{D}}_R(k)$ and $\widetilde{\mathbb{H}}_R(k)$ have at most $O(kn)$ elements. Analogous sets $\widetilde{\mathbb{D}}_B(k)$ and $\widetilde{\mathbb{H}}_B(k)$ can be defined with respect to the blue color, and they are related to the higher-order Voronoi diagrams of $R \cup \widetilde{B}$.

For every blue point $q \in \widetilde{B}$, let $\widetilde{S}_B(q,k)$ be the set of straight-line segments s such that for each point o of s the disk $D(o,q)$ is blue, covers k blue points of $\widetilde{B}$, and its boundary contains other blue point of $\widetilde{B}$ distinct from q. The set of segments $\widetilde{S}_B(q,k)$ is a subset of the edges in the $(k-1)$th-order Voronoi diagram of $R \cup \widetilde{B}$.

The sets $\widetilde{\mathbb{D}}_R(k)$ and $\widetilde{\mathbb{H}}_R(k)$ can be obtained from the $(k-2)$th- and the $(k-1)$th-order Voronoi Diagrams of $\widetilde{R} \cup B$. Since computing the kth-order Voronoi diagram of n points can be done in $O(k^2 n \log n)$ time [Lee82], the sets $\widetilde{\mathbb{D}}_R(k)$ and $\widetilde{\mathbb{H}}_R(k)$ can be obtained in $O(k^2 n \log n)$ time. Similarly, using the $(k-2)$th- and the $(k-1)$th-order Voronoi diagrams of $R \cup \widetilde{B}$ we can construct the sets $\widetilde{\mathbb{D}}_B(k)$ and $\widetilde{S}_B(q,k)$ for all $q \in B$ in $O(k^2 n \log n)$ time.

Let $\widetilde{V}_k$ denote the Furthest Disk Voronoi Diagram of $\widetilde{\mathbb{D}}_B(k)$. We can compute $\widetilde{V}_k$ and preprocess it for point location queries in $O((kn)\log(kn)) = O(kn \log n)$ time [Rap92, Sno97], so that any point location in $\widetilde{V}_k$ can be answered in $O(\log n)$ time. For each blue point q and each edge e of $\widetilde{S}_B(q,k)$, consider the double wedge obtained by the union of lines that intersect q and e, and let $\triangle(q,e)$ be wedge that does not contain e.

We now discuss how to decide if there exists a solution with the properties listed in Lemma 2.3 that covers k points of $\widetilde{R}$ and k points of $\widetilde{B}$. Finding if there is a solution satisfying condition (a) reduces to point location queries in $\widetilde{V}_k$ with the centers of disks from $\widetilde{\mathbb{D}}_R(k)$ and far enough points representing the "centers" of half-planes from $\widetilde{\mathbb{H}}_R(k)$. We spend here $O(kn \log n)$ time in total because we have to make $O(|\widetilde{\mathbb{D}}_R(k)| + |\widetilde{\mathbb{H}}_R(k)|) = O(kn)$ point locations in $\widetilde{V}_k$.

Finding if there is a solution satisfying condition (b) reduces to deciding

$$\text{does some center of some disk in } \widetilde{\mathbb{D}}_R(k) \text{ lie in } \bigcup_{q \in B} \bigcup_{e \in \widetilde{S}_B(q,k)} \triangle(q,e)?$$

This is a problem of deciding if there is any incidence between a set of $O(kn)$ points and a set of $O(kn)$ triangles in the plane, which can be solved in $O((kn)^{4/3} \operatorname{polylog}(nk)) = O((kn)^{4/3} \operatorname{polylog} n)$ time using machinery for simplicial range searching [Cha10, Mat93]. Finding if there is a solution satisfying condition (c) can be done similarly to condition (b). The solutions satisfying condition (d) can be obtained with symmetric algorithms.

The algorithm spends $O(k^2 n \log n) + O((kn)^{4/3} \operatorname{polylog} n) = O((k^2 n + (kn)^{4/3}) \operatorname{polylog} n)$ time in total. □

Lemma 3.5 *Let c_0 be a constant, and assume that $i, j > \varepsilon n / c_0$. Then the ApproxDecision-2DD-problem can be solved in $O(k^4 \varepsilon^{-3} n \operatorname{polylog} n)$ time with high probability.*

Proof. We reuse the notation and approach in the proof of Lemma 3.4. In this case, with high probability $\widetilde{R}$ and $\widetilde{B}$ have $O((n/i)\varepsilon^{-2} \log n) = O(k/\varepsilon)$ points. Since each disk of

$\widetilde{\mathbb{D}}_R(k)$ can contain only red points in its interior, the set $\widetilde{\mathbb{D}}_R(k)$ contains $O(|\widetilde{R}|^3)$ disks, which is bounded by $O((k/\varepsilon)^3)$ with high probability. Similarly, $\widetilde{\mathbb{D}}_B(k)$, $\widetilde{\mathbb{H}}_R(k)$, and $\widetilde{\mathbb{H}}_B(k)$ contain $O((k/\varepsilon)^3)$ disks or half-planes with high probability.

All the steps in the proof of Lemma 3.4 take $O(kn \log n)$ time but for the step where we have to decide

$$\text{does some center of some disk in } \widetilde{\mathbb{D}}_R(k) \text{ lie in } \bigcup_{q\in B} \bigcup_{e\in \widetilde{S}_B(q,k)} \triangle(q,e)?$$

In this scenario, the problem is that of deciding if there is an incidence between a set of $O((k/\varepsilon)^3)$ points and $O(kn)$ triangles. This problem can be solved trivially in $O(k^4\varepsilon^{-3}n)$ time by checking each point against each triangle. The result follows. □

We can summarize the results of this section with the following Lemma.

Lemma 3.6 *There is a randomized algorithm* RandAlg *that in time* $O(n^{4/3}\varepsilon^{-4} \operatorname{polylog} n)$ *returns a feasible solution and has the following property: if there is a solution to the* (i,j)*-2DD-problem, with probability at least* $1 - O(1/n^3)$ *it returns a solution to the* $((1-\varepsilon)i, (1-\varepsilon)j)$*-2DD-problem.*

If c_0 *is a constant and* $i, j > \varepsilon n/c_0$ *then the algorithm* RandAlg *takes* $O(\varepsilon^{-11} n \operatorname{polylog} n)$ *time with high probability*

Proof. We construct the ρ_i-sample $\widetilde{R}$ or R and the ρ_j-sample $\widetilde{B}$ of B, and set $k = (1 - \varepsilon/2)c\varepsilon^{-2}\log n$. We then solve the ApproxDecision-2DD-problem using Lemma 3.4, which takes $O((k^2 n + (kn)^{4/3}) \operatorname{polylog} n) = O(((\varepsilon^{-2}\log n)^2 n + n^{4/3}(\varepsilon^{-2}\log n)^{4/3}) \operatorname{polylog} n) = O(n^{4/3}\varepsilon^{-4} \operatorname{polylog} n)$ time. If we find a solution (D_r, D_b) to the ApproxDecision-2DD-problem, then we return that solution. Otherwise, we return an arbitrary pair of feasible disks. The properties of the algorithm follow from Lemma 3.2. When $i, j > \varepsilon n/c_0$, we just use the algorithm from Lemma 3.5 instead of Lemma 3.4, which takes $O(k^4\varepsilon^{-3} n \operatorname{polylog} n) = O(\varepsilon^{-11} n \operatorname{polylog} n)$ time because $k = O(\varepsilon^{-2}\log n)$. □

3.3 Algorithm to approximate the 2DD-problem.

Theorem 3.7 *There is a randomized algorithm that computes a* $(1-\varepsilon)$*-approximation to the 2DD-problem in* $O(n^{4/3}\varepsilon^{-6} \operatorname{polylog} n)$ *time. The probability of error is at most* $O(1/n)$*.*

Proof. Consider the set

$$T = \left\{ \lceil (1+\varepsilon)^s \rceil \mid s = 0, 1, 2, 3, \ldots, \log_{1+\varepsilon} n \right\}.$$

Note that T has $O(\log_{1+\varepsilon} n) = O(\varepsilon^{-1}\log n)$ elements because

$$\lim_{\varepsilon\to 0} \frac{\varepsilon^{-1}\ln n}{\log_{1+\varepsilon} n} = \lim_{\varepsilon\to 0} \frac{\varepsilon^{-1}\ln n}{\ln n \,/\, \ln(1+\varepsilon)} = \lim_{\varepsilon\to 0} \frac{\ln(1+\varepsilon)}{\varepsilon} = \lim_{\varepsilon\to 0} \frac{1/(1+\varepsilon)}{1} = 1.$$

For any integer i, define $\psi(i) = \left\lceil (1+\varepsilon)^{\lfloor \log_{1+\varepsilon} i \rfloor} \right\rceil$, and note that $\psi(i) \in T$. We have that

$$\psi(i) = \left\lceil (1+\varepsilon)^{\lfloor \log_{1+\varepsilon} i \rfloor} \right\rceil \le \left\lceil (1+\varepsilon)^{\log_{1+\varepsilon} i} \right\rceil = \lceil i \rceil = i$$

and

$$\psi(i) \;=\; \left\lceil (1+\varepsilon)^{\lfloor \log_{1+\varepsilon} i \rfloor} \right\rceil \;\geq\; \left\lceil (1+\varepsilon)^{(\log_{1+\varepsilon} i)-1} \right\rceil \;=\; \left\lceil \frac{i}{1+\varepsilon} \right\rceil \;\geq\; \frac{i}{1+\varepsilon} \geq (1-\varepsilon)i\,.$$

We thus conclude that

$$(1-\varepsilon)i \;\leq\; \psi(i) \;\leq\; i.$$

We can now see that among the pairs $(i,j) \in T^2$ for which there exists a solution to the (i,j)-2DD-problem, one maximizing $i+j$ is a $(1-\varepsilon)$-approximation to the 2DD-problem. Indeed, if (i^*, j^*) is an optimal solution, then there exists a solution to the $(\psi(i^*), \psi(j^*))$-2DD-problem because $\psi(i^*) \leq i$ and $\psi(j^*) \leq j$. Furthermore, a solution to the $(\psi(i^*), \psi(j^*))$-2DD-problem covers $\psi(i^*) + \psi(j^*) \geq (1-\varepsilon)(i^* + j^*)$ points, and is thus a $(1-\varepsilon)$-approximation. Finally, note that $(\psi(i^*), \psi(j^*)) \in T^2$.

For each pair $(i,j) \in T^2$ we use the algorithm RandAlg of Lemma 3.6 to find a feasible solution. Among all the returned solutions, we select the best one by counting the points covered. Let us assume first that there are no errors in all our calls to RandAlg in Lemma 3.6. When we call RandAlg for the value $(\psi(i^*), \psi(j^*))$, we obtain a solution to the $((1-\varepsilon)\psi(i^*), (1-\varepsilon)\psi(j^*))$-2DD-problem, and thus a solution to the $((1-\varepsilon)^2 i^*, (1-\varepsilon)^2 j^*)$-2DD-problem. Thus, the best solution found among the $O(|T|^2)$ calls to RandAlg is a $(1-\varepsilon)^2$-approximation to the optimal solution. Since $(1-\varepsilon)^2 \geq 1-2\varepsilon$, we can achieve a $(1-\varepsilon')$-approximation by taking $\varepsilon = \varepsilon'/2$. [1]

To analyze the running time of the algorithm, note that it makes $|T|^2 = O(\varepsilon^{-2}\log^2 n)$ calls to the algorithm RandAlg, and thus takes $O(\varepsilon^{-2}\log^2 n \cdot (n^{4/3}\varepsilon^{-4}\operatorname{polylog} n)) = O(n^{4/3}\varepsilon^{-6}\operatorname{polylog} n)$ time. Selecting the best solution among the $|T|^2$ feasible solutions can be done in $O(|T|^2 n) = O(n\varepsilon^{-2}\operatorname{polylog} n)$ time. Since each call to RandAlg (Lemma 3.6) has a probability of error bounded by $1/n^3$ and $|T^2| = O(n^2)$ it follows from the union bound that all outputs of RandAlg are correct[2] with probability at least $1 - O(1/n)$. Thus, the algorithm finds a $(1-\varepsilon)$-approximation with probability at least $1 - O(1/n)$. □

For the special case where the optimal solution covers several points, we can obtain a $(1-\varepsilon)$-approximation in near-linear time.

Corollary 3.8 *Let c_0 be a constant. If the optimal solution to the 2DD-problem covers at least n/c_0 points, then we can compute a $(1-\varepsilon)$-approximation to the 2DD-problem in $O(n\varepsilon^{-13}\operatorname{polylog} n)$ time. The probability of error is at most $O(1/n)$ and the running time of the algorithm is with high probability.*

Proof. We use the same algorithm as in Theorem 3.7, but only restrict attention to values $(i,j) \in T^2$ with $i,j \geq \varepsilon n/c_0$. Since we can use the second case in Lemma 3.6, each call to RandAlg takes $O(\varepsilon^{-11} n \operatorname{polylog} n)$ time with high probability. We make $|T|^2 = O(\varepsilon^{-2}\log^2 n)$ calls to the algorithm RandAlg, and thus spend a total of $O(\varepsilon^{-2}\log^2 n \cdot \varepsilon^{-11} n \operatorname{polylog} n) = O(\varepsilon^{-13} n \operatorname{polylog} n)$ time. □

[1] We could run a more careful search in T^2 as it was done in Theorem 2.5. However, this would only decrease the running time by a factor of $|T| = O(\varepsilon^{-1}\log n)$. Also, we could reduce the number of calls to RandAlg by computing first a simple 2-approximation algorithm: choose the best among the disks covering only blue points and the disks covering only red points. If the returned solution covers A points, only pairs $(i,j) \in |T|^2$ with $A \leq i+j \leq 2A$ are relevant. However, with the approach that we describe, we obtain $(1-\varepsilon)$-approximations for both i^* and j^*. This is convenient in several scenarios, for example, when maximizing $\min\{|D_b \cap B|, |D_r \cap R|\}$.

[2] In fact, the algorithm returns a $(1-\varepsilon)$-approximation unless the call to RandAlg with the value $(\psi(i^*), \psi(j^*))$ fails. However, with this approach we $(1-\varepsilon)$-approximate both i^* and j^*.

4 Conclusions

Given a bicolored point set $P = R \cup B$, in this paper we have studied the problem of finding two disjoint disks D_r and D_b such that D_r (resp. D_b) covers only red (resp. blue) points and the number of elements of P contained in $D_r \cup D_b$ is maximized. We gave an exact algorithm running in $O(n^{11/3} \operatorname{polylog} n)$ time based on solving instances of the (i, j)-2DD-problem, which asks for the same disks D_r and D_b subject to $|D_r \cap R| \geq i$ and $|D_b \cap B| \geq j$. The (i, j)-2DD-problem was solved in $O(n^{8/3} \operatorname{polylog} n)$ time for any given values of i and j.

Using random samples of R and B to estimate $|D \cap R|$ and $|D \cap B|$, where D is any disk, we presented a randomized $(1 - \varepsilon)$-approximation algorithm running in $O(n^{4/3}\varepsilon^{-6} \operatorname{polylog} n)$ time, where the probability of error is $O(1/n)$. A better, near-linear time algorithm was obtained when the solution covers a constant fraction of the input points.

It is worth noticing that if we want to solve the max-min version of the problem, that is, to maximize the minimum between $|D_r \cap R|$ and $|D_b \cap B|$, then we can consider $|D_r \cap R| = |D_b \cap B|$. Therefore, the problem reduces to finding a maximum value of i such that the (i, i)-2DD-problem has solution. This can be done in $O(\log n \cdot n^{8/3} \operatorname{polylog} n) = O(n^{8/3} \operatorname{polylog} n)$ time using a binary search. Additionally, a $(1 - \varepsilon)$-approximation algorithm can also be obtained. Indeed, we can run RandAlg of Lemma 3.6 only for the pairs $(i, i) \in T$, where T is the set in the proof of Theorem 3.7. Since $|T| = O(\varepsilon^{-1} \log n)$, a randomized $(1 - \varepsilon)$-approximation algorithm running in $O(\varepsilon^{-1} \log n \cdot n^{4/3}\varepsilon^{-4} \operatorname{polylog} n) = O(n^{4/3}\varepsilon^{-5} \operatorname{polylog} n)$ time is obtained.

References

[ABMS98] P. K. Agarwal, M. de Berg, J. Matousek, and O. Schwarzkopf. Constructing levels in arrangements and higher order voronoi diagrams. *SIAM J. Comput.*, 27:654–667, 1998.

[AHP08] B. Aronov and S. Har-Peled. On approximating the depth and related problems. *SIAM J. Comput.*, 38:899–921, 2008.

[Aur91] F. Aurenhammer. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[BDBL$^+$09] S. Bereg, J. M. Díaz-Báñez, D. Lara, P. Pérez-Lantero, C. Seara, and J. Urrutia. Bichromatic discrepancy via convex partitions. In *XIII Encuentros de Geometría Computacional*, pages 259–265, Zaragoza, Spain, 2009.

[CDBPL$^+$09] C. Cortés, J. M. Díaz-Báñez, P. Pérez-Lantero, C. Seara, J. Urrutia, and I. Ventura. Bichromatic separability with two boxes: A general approach. *J. Algorithms*, 64:79–88, 2009.

[CDBS$^+$08] S. Cabello, J. M. Díaz-Báñez, C. Seara, J. A. Sellarès, J. Urrutia, and I. Ventura. Covering point sets with two disjoint disks or squares. *Comput. Geom. Theory Appl.*, 40:195–206, 2008.

[Cha05] T. M. Chan. Low-dimensional linear programming with violations. *SIAM J. Comput.*, 34:879–893, 2005.

[Cha10] T. M. Chan. Optimal partition trees. In *Proc. of the 26th Annual Symposium on Computational Geometry*, SoCG'10, pages 1–10, 2010.

[CSY87] R. Cole, M. Sharir, and C. K. Yap. On k-hulls and related problems. *SIAM J. Comput.*, 16:61–77, 1987.

[dBCHP09] M. de Berg, S. Cabello, and S. Har-Peled. Covering many or few points with unit disks. *Theory Comput. Syst.*, 45(3):446–469, 2009.

[DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition, November 2001.

[EHL$^+$02] J. Eckstein, P. L. Hammer, Y. Liu, M. Nediak, and B. Simeone. The maximum box problem and its applications to data analysis. *Comput. Optim. Appl.*, 23:285–298, 2002.

[ERvK96] H. Everett, J-M. Robert, and M. van Kreveld. An optimal algorithm for the ($\leq k$)-levels, with applications to separation and transversal problems. *Int. J. Comput. Geometry Appl.*, 6(3):247–261, 1996.

[GO95] A. Gajentaan and M. H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom.*, 5:165–185, 1995.

[Hou93] M. E. Houle. Algorithms for weak and wide separation of sets. *Discrete Applied Mathematics*, 45(2):139–159, 1993.

[HSM01] D. J. Hand, P. Smyth, and H. Mannila. *Principles of data mining*. MIT Press, Cambridge, MA, USA, 2001.

[Lee82] D-T. Lee. On k-nearest neighbor voronoi diagrams in the plane. *IEEE Trans. Computers*, 31(6):478–487, 1982.

[LN03] Y. Liu and M. Nediak. Planar case of the maximum box and related problems. In *Proc. of the Canadian Conference on Computational Geometry*, pages 11–13, August 2003.

[Mat93] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete and Computational Geometry*, 10:157–182, 1993.

[MR96] R. Motwani and P. Raghavan. Randomized algorithms. *ACM Comput. Surv.*, 28:33–37, 1996.

[Rap92] D. Rappaport. A convex hull algorithm for discs, and applications. *Comput. Geom. Theory Appl.*, 1:171–187, 1992.

[Sno97] J. Snoeyink. Point location. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of discrete and computational geometry*, pages 559–574. CRC Press, Inc., Boca Raton, FL, USA, 1997.