COVERING MANY POINTS WITH A SMALL-AREA BOX*

Mark de Berg,† *Sergio Cabello,*‡ *Otfried Cheong,*§ *David Eppstein,*¶ *and Christian Knauer*‖

ABSTRACT. Let P be a set of n points in the plane. We show how to find, for a given integer $k > 0$, the smallest-area axis-parallel rectangle that covers k points of P in $O(nk^2 \log n + n \log^2 n)$ time. We also consider the problem of, given a value $\alpha > 0$, covering as many points of P as possible with an axis-parallel rectangle of area at most α. For this problem we give a probabilistic $(1-\varepsilon)$-approximation that works in near-linear time: In $O((n/\varepsilon^4) \log^3 n \log(1/\varepsilon))$ time we find an axis-parallel rectangle of area at most α that, with high probability, covers at least $(1-\varepsilon)\kappa^*$ points, where κ^* is the maximum possible number of points that could be covered.

1 Introduction

In this paper we consider two closely related shape-fitting problems for a given point set P in the plane. In both problems we are searching for an axis-parallel rectangle, or a *box* as we will call it, and we are interested in the trade-off between the box area and the number of points covered by the box. More precisely, we are interested in the following two optimization problems.

- Given a set P of points and an integer $k \geqslant 2$, find

$$\text{area}^*(P,k) \;=\; \min\big\{\text{area}(R) \;\big|\; R \text{ is a box with } |R \cap P| \geqslant k\big\}.$$

  That is, we are interested in covering at least k points of P with a box of minimum area.

- Given a set P of points and a real value $\alpha > 0$, find

$$\kappa^*(P,\alpha) \;=\; \max\big\{|R \cap P| \;\big|\; R \text{ is a box with } \text{area}(R) \leqslant \alpha\big\}.$$

  That is, we are interested in covering the maximum number of points of P with a box of area at most α.

*Mark de Berg was supported by the Netherlands Organization for Scientific Research (NWO) under project no. 024.002.003. Sergio Cabello was supported by the Slovenian Research Agency, program P1-0297 and projects J1-8130 and J1-8155. Otfried Cheong was supported by ICT R&D program of MSIP/IITP [R0126-15-1108]. David Eppstein was supported in part by NSF grants CCF-1228639, CCF-1618301, and CCF-1616248. Christian Knauer was supported in part by DFG grant Kn 591/3-3.

†*Department of Computer Science, TU Eindhoven, the Netherlands.*

‡*Department of Mathematics, IMFM, and Department of Mathematics, FMF, University of Ljubljana, Slovenia.*

§*School of Computing, KAIST, Korea.*

¶*Computer Science Department, University of California, Irvine, USA.*

‖*Computer Science Department, University of Bayreuth, Germany.*

The two problems are closely related because for all finite point sets P, and all $k \in \mathbb{N}$ and $\alpha \in \mathbb{R}_{>0}$ we have

$$\text{area}^*(P,k) \leqslant \alpha \iff \kappa^*(P,\alpha) \geqslant k.$$

So the second problem can be solved using binary search on k and a solution to the first problem.

When minimizing the area of the box covering k points, the set of optimal solutions is invariant under scaling of either of the axes. This means that, if we consider any map $(x,y) \mapsto \varphi(x,y) = (\alpha_1 x + \beta_1, \alpha_2 y + \beta_2)$ with $\alpha_1, \alpha_2 \neq 0$, then a box R is an optimal solution for $\text{area}^*(P,k)$ if and only if $\varphi(R)$ is a solution for $\text{area}^*(\varphi(P),k)$. Thus, minimizing the area is especially useful when the units of each axis have incomparable meanings. In contrast, in such a case it is meaningless to minimize the perimeter.

The problem of covering k points with a minimum-area (or minimum-perimeter) box was previously considered by Segal and Kedem [20], who provided an algorithm suitable for values of k close to n, with running time $O(n + k(n-k)^2)$. In contrast, we study the case when k is small, so that it is preferable to decrease the dependence on n at the expense of increasing the dependence on k. For the case of small k, several papers [3, 7, 20] erroneously claim that previous algorithms of Aggarwal et al. [2] and Eppstein and Erickson [12] solve the problem in running time $O(k^2 n \log n)$ or $O(n \log n + k^2 n)$, respectively. However, these previous algorithms apply only to the minimum-perimeter version of the problem. They do not work for the minimum-area version, because they are based on the fact that for the minimum-perimeter version, the optimal subset of k points can be found among the $O(k)$ nearest neighbors to one of the points—something which is not true for the minimum-area version. The same obstacle appears when trying to extend the algorithms of Datta et al. [8] from the minimum-perimeter to the minimum-area problem. For the minimum-area problem that we study here, we cannot restrict our attention to sets of nearest neighbors, and must use alternative methods to obtain our time bounds. The results in those papers do not depend on the mistaken claim, only the attribution of previous work is incorrect.

After our preprint was made public [9], Kaplan et al. [16] showed that the problem of covering k points with a minimum-area box can be solved in $O(n^{5/2} \log^2 n)$ time. This is the first subcubic algorithm and it is more efficient than previous results for a large range of k. For the minimum-perimeter problem, they provide an algorithm running in $O(nk^{3/2} \log k \log n)$ time. However, as they note, one of the steps used in their algorithm does not work for the minimum-area problem. The difficulty is essentially as we mentioned above: For the minimum-area problem, we do not know how to transform the problem into $O(n/k)$ instances of $O(k)$ points each.

There have been several works on minimizing the size of the smallest *disk* that contains k points. Here it does not matter whether we minimize the area or the perimeter. Har-Peled and Mazumdar [13] give a randomized algorithm to find a disk that contains k points in $O(nk)$ expected time, improving the works by Efrat, Eppstein, Erickson, Matoušek, and Sharir [11, 12, 18]. In follow-up work, Har-Peled and Raichel [14] aim for fast $(1+\varepsilon)$-approximations. Das et al. [7] consider covering k points with rectangles of arbitrary orientations.

The problems that we are interested in, where we want to find an optimal box of

arbitrary aspect ratio, are relatively easy if we make certain assumptions about the input. For instance, if we were given the aspect ratio of an optimal box, we could rescale one axis to reduce the problem to finding an optimal square, a problem that is very similar to the problem for disks. Similarly, if we had, say, a 2-approximation to the aspect ratio of the optimal box, then we could rescale one axis and reduce the search to fat boxes. In this scenario, finding a smallest square box gives a constant-factor approximation to the optimum fat box, and using a grid approach, like in Har-Peled and Mazumdar [13], we only need to solve $O(n/k)$ instances of size $O(k)$, which can be done in roughly $O(nk^2)$ time. Thus, we can search for the optimal box with constant fatness in roughly $O(nk^2)$ time. Also, if we assume that the coordinates are integers between 0 and a bound U, this approach allows us to compute $\text{area}^*(P,k)$ in roughly $O(nk^2 \log U)$ time, by trying $O(\log U)$ different aspect ratios in geometric progression. The main goal of our paper is to avoid any such assumptions, and to still get similar running times.

Our results. Here is a summary of our main results and an overview of the approach. Let P be a given set of n points in the plane.

(a) We show how to find, for a given integer $k > 0$, the value $\text{area}^*(P,k)$ in $O(nk^2 \log n + n \log^2 n)$ time. Within the same time bound we can also construct an optimal solution, that is, a box that contains k points of P and whose area is $\text{area}^*(P,k)$. This is the only known algorithm with a near-linear dependency on n; see the discussion above.

To achieve this result, we use a divide-and-conquer method that resembles the one by Aronov et al. [4]. More precisely, we use a horizontal line ℓ that splits the points into two sets of roughly the same cardinality, compute the best rectangle intersected by ℓ, and recursively solve the problems above and below the line. To find the best rectangle intersected by ℓ, we generate $O(n)$ subproblems, each with $O(k)$ points, where we only have to consider boxes that contain a fixed point on the boundary. These subproblems are generated using an idea based on proximity.

In fact, we solve a slightly more general problem that enables some improvements in the running time of the problem discussed below, in item (b). We show how to generate in $O(nk \log n + n \log^2 n)$ time all subproblems that arise in the recursive process for a given k, and observe that these subproblems can also be used for $k' < k$. This allows us to find in $O(nk^2 \log n)$ time the minimum area-area box that contains k' points of P, for any given $k' \leqslant k$.

These results are presented in Section 2.

(b) We give a randomized algorithm that, for a given value $\alpha > 0$ and a parameter $\varepsilon \leqslant 1/2$, with high probability runs in $O((n/\varepsilon^4) \log^3 n \log(1/\varepsilon))$ time and returns a box that has area α and covers at least $(1-\varepsilon)\kappa^*(P,\alpha)$ points of P. Note that the running time is $O(n \log^3 n)$ when ε is fixed.

An overview of the approach is as follows; a similar high-level approach is used for example in [1,5,6,10]. First, we find a simple 4-approximation to the value $\kappa^*(P,\alpha)$. Then we use a random sample S of P such that, for each box R with $\Theta(\kappa^*(P,\alpha))$ points of P inside, the value $|R \cap S| \cdot \frac{|P|}{|S|}$ is a $(1 \pm \varepsilon)$-approximation to the value $|R \cap P|$ (with high probability).

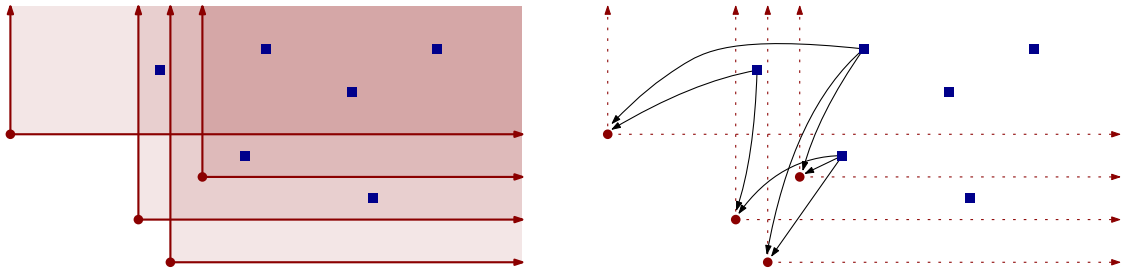

Figure 1: Left: Example of the scenario considered in Lemma 1. Right: Points to be reported for each rectangle when k=2. An arrow indicates that the tail is reported for the head.

This is a common technique in approximate range counting. Choosing the size of S appropriately we can guarantee that, with high probability, $\kappa^*(S,\alpha) = \Theta((1/\varepsilon^2)\log n)$ and an optimal solution for $\kappa^*(S,\alpha)$ contains $(1-\varepsilon)\kappa^*(P,\alpha)$ points of P. Thus, finding an optimal box for the sample S we get a $(1-\varepsilon)$-approximation for P. The problem for S can be solved using the algorithm of (a) for all values $k' \in \{1,\ldots,\kappa^*(S,\alpha)\}$. Since $\kappa^*(S,\alpha) = \Theta((1/\varepsilon^2)\log n)$, each value of k' is $O((1/\varepsilon^2)\log n)$, each use of the the algorithm of (a) takes near-linear time.

Some additional observations are used to slightly improve the final running time. Firstly, instead of a linear search, we can use a binary search to find $\kappa^*(S,\alpha)$. At each step we have to decide whether $\text{area}^*(S,k') \leqslant \alpha$ for some given $k' = O((1/\varepsilon^2)\log n)$. Secondly, we do not really need to compute $\kappa^*(S,\alpha)$ exactly for the random sample S, as we can afford to use a $(1-\varepsilon)$-approximation instead. Finally, we are reusing always the same set S, but the test values k' are different. Thus, the generalized problem mentioned in item (a) becomes useful.

These results are described in Section 3.

Notation and conventions. As noted, a *box* is an axis-parallel rectangle. For a box R, let $\text{top}(R)$ and $\text{bot}(R)$ be its top and bottom edge. For a point $p \in \mathbb{R}^2$, we use p_x and p_y for its x- and y-coordinate, respectively.

We assume that the point set is in *general position*, meaning that no two points have the same x-coordinate or the same y-coordinate. This can be enforced by a symbolic perturbation of the points. For example, we can index the points as $p_1,\ldots,p_n$ and replace each point p_i with the point $p_i + (i\cdot\varepsilon, i\cdot\varepsilon)$ for an infinitesimal value $\varepsilon > 0$. When minimizing the area of the box covering k points, we drop in the resulting area any terms that depend on ε. When maximizing the number of points to be covered, we allow boxes of area $\alpha + n\varepsilon\rho$, where ρ is the perimeter of the bounding box of P.

2 Minimizing area for a given number of points

We will use the following result for batched reporting in 2-sided rectangles. See Figure 1 for an example.

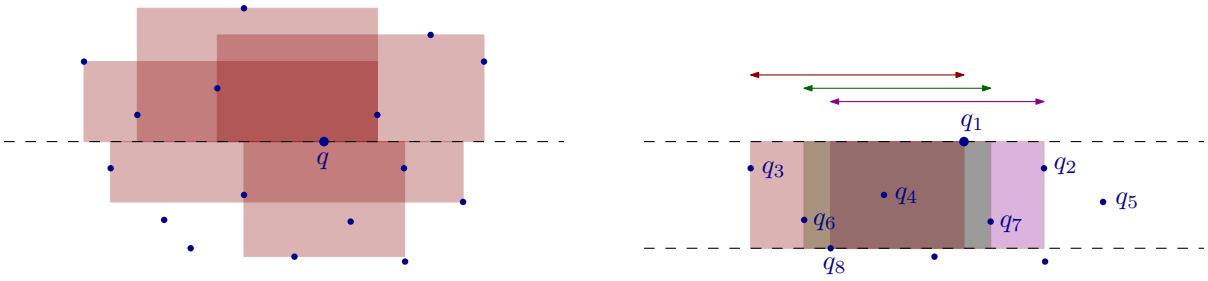

Figure 2: Left: Example of the scenario considered in the definition of $\Phi(Q, q, k)$. Some of the feasible boxes when $k = 5$ are shown. Right: The boxes considered when $k = 5$, $q \in \text{top}(R)$ and we consider $Q_8 = \{q_1, \ldots, q_8\}$. Thus $q_8 \in \text{bot}(R)$. The span of the relevant boxes is shown with the arrows above.

Lemma 1. *Let A and B be sets of at most n points in $\mathbb{R}^2$. For each point $b \in B$, let R_b be the 2-sided rectangle $[b_x, \infty) \times [b_y, \infty)$. In time $O(kn + n \log n)$ we can find, for all $b \in B$, the k points in $A \cap R_b$ with smallest x-coordinate.*

Proof. The result can be obtained in a standard manner, using a sweep-line algorithm that sweeps the plane with a vertical line ℓ from left to right. For completeness we give the details.

Let A_ℓ and B_ℓ be the points to the left of ℓ of A and B, respectively. Consider the family of rectangles $\mathcal{R}_\ell = \{R_b \mid b \in B_\ell\}$. At each moment, we maintain the subset $\mathcal{R}'_\ell \subseteq \mathcal{R}_\ell$ of rectangles that do not contain k points of A_ℓ. The rectangles $R_b \in \mathcal{R}'_\ell$ are stored in a dynamic balanced binary search tree T sorted by the value b_y. Moreover, for each rectangle $R_b \in \mathcal{R}'_\ell$ we also store a list $\mathcal{L}_b$ of the points of A_ℓ that it contains and the length of the list $\mathcal{L}_b$, that is, $|R_b \cap A_\ell|$.

When the line ℓ arrives at a point $a \in A$, we find the m rectangles of $\mathcal{R}'_\ell$ that contain a using a traversal of the tree T, which takes $O(m + \log n)$ time. For each of the m rectangles $R_b \in \mathcal{R}'_\ell$ that contain a, we add a to the list $\mathcal{L}_b$. Moreover, if $\mathcal{L}_b$ now contains k points, then R_b does not belong to $\mathcal{R}'_\ell$ anymore and we remove the record from the tree T.

When the line ℓ reaches a point $b \in B$, then R_b becomes an element of $\mathcal{R}_\ell$ and we insert R_b into T. If there is a point a that belongs to A and B, then we first consider it as a point of B and then as a point of A. In this way a becomes an element of R_a.

Each insertion or deletion in T takes $O(\log n)$. We make $|B|$ insertions and at most $|B|$ deletions in T, for a total of $O(n \log n)$ time. For each point $a \in A$ we spend $O(\log n)$ plus $O(1)$ time for each rectangle R_b for which we report a. Thus, the running time is $O(kn + n \log n)$. □

For a set Q of points, a point $q \in Q$, and a parameter k define

$$\Phi(Q, q, k) := \min \{\text{area}(R) \mid R \text{ is a box with } q \in \text{top}(R) \text{ or } q \in \text{bot}(R) \text{ and } R \text{ contains at least } k \text{ points of } Q.\ \}$$

An example is shown in Figure 2. We will reduce our problem to many instances of

the problem of computing $\Phi(Q, q, k)$ with $|Q| = O(k)$. We first discuss how to solve such instances.

Lemma 2. *Given Q, q and k, we can compute $\Phi(Q, q, k)$ in $O(|Q|^2)$ time.*

Proof. Let us discuss the case where $q \in \text{top}(R)$, the other case being symmetric. Let $q_1, q_2, \ldots, q_m$ be the points of Q whose y-coordinate is not larger than q_y, in decreasing order of y-coordinate, and let $Q_i = \{q_1, \ldots, q_i\}$.

Once we have a sorted list with the elements of Q_i in increasing x-coordinate, then we can find in $O(|Q_i|) = O(i)$ time the minimum-area box R that contains k points with $q \in \text{top}(R)$ and $q_i \in \text{bot}(R)$, using a linear scan of the list with two pointers that are offset by k elements. See Figure 2 for an example.

We can therefore proceed as follows: We first compute the set Q_m and sort it by x-coordinate, in time $O(|Q| \log |Q|)$. We then repeatedly compute the best box for the current set Q_i (initially $i = m$) in time $O(i)$, then delete from the list the element with the smallest y-coordinate to obtain Q_{i-1}, again in time $O(i)$. The total running time is $O(|Q|^2)$. □

For a set P of points, a horizontal line ℓ, and a parameter k define

$$\Psi(P, \ell, k) := \min \{\text{area}(R) \mid R \text{ is a box intersecting } \ell \text{ such that } R \text{ contains at least } k \text{ points of } P\}$$

Recall that $\text{area}^*(P, k)$ is the area of the optimal solution for the original, global problem. Thus, it is obvious that $\text{area}^*(P, k) \leqslant \Psi(P, \ell, k)$ for all P, ℓ and k. The following lemma explains that when an optimal, global solution is intersected by the line ℓ, then we can reduce the search to a few small problems of size $O(k)$.

Lemma 3. *Given P, ℓ, and k, we can compute in $O(kn + n \log n)$ time sets $Q_p \subseteq P$, indexed by $p \in P$, with the following properties:*

- *Q_p has $O(k)$ points for each $p \in P$.*
- *For each $k' \leqslant k$, if $\text{area}^*(P, k') = \Psi(P, \ell, k')$, then $\text{area}^*(P, k') = \Phi(Q_p, p, k')$ for some $p \in P$.*

Proof. For each $p \in P$ let $\overline{p}$ be the point symmetric to p with respect to the line ℓ. For each point q of the plane, $q \notin \ell$, we define the following objects. See Figure 3.

- Let $\text{slab}(q)$ be the horizontal slab defined by ℓ and the line parallel to ℓ through q.
- Let $R_q^{\rightarrow}$ be the 3-sided rectangle $\text{slab}(q) \cap \{(x, y) \in \mathbb{R}^2 \mid x \geqslant q_x\}$ and let $P_q^{\rightarrow}$ be the k points of P with smallest x-coordinate inside $R_q^{\rightarrow}$.
- Let $R_q^{\leftarrow}$ be the 3-sided rectangle $\text{slab}(q) \cap \{(x, y) \in \mathbb{R}^2 \mid x \leqslant q_x\}$ and let $P_q^{\leftarrow}$ be the k points of P with largest x-coordinate inside $R_q^{\leftarrow}$.

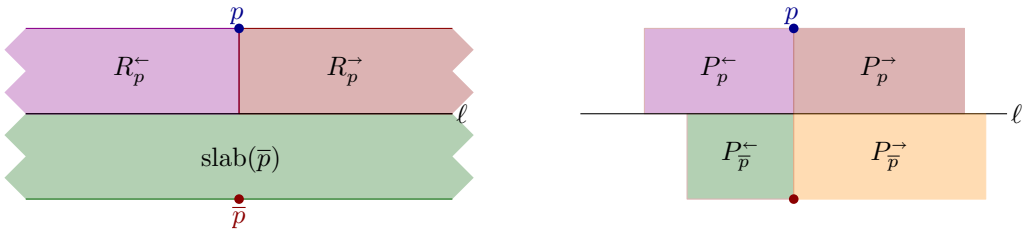

Figure 3: Notation used in Lemma 3. On the right side we show the portions of the 3-sided rectangles that contain Q_p.

For each $p \in P$, we define Q_p as the union of $P_p^{\rightarrow}$, $P_p^{\leftarrow}$, $P_{\overline{p}}^{\rightarrow}$, and $P_{\overline{p}}^{\leftarrow}$. It is clear that each set Q_p has at most $4k$ points, so the first property of the lemma holds.

To show the second property, consider any fixed $k' \leqslant k$ and assume that $\text{area}^*(P, k') = \Psi(P, \ell, k')$. Then there exists an optimal box R^* with $\text{area}(R^*) = \text{area}^*(P, k')$ such that R^* intersects ℓ. Let t^* and b^* be points of P on $\text{top}(R^*)$ and $\text{bot}(R^*)$, respectively. Assume without loss of generality that the distance from t^* to ℓ is at least the distance from b^* to ℓ. This means that $R^* \cap P$ is contained in $\text{slab}(t^*) \cup \text{slab}(\overline{t^*})$.

We next show that $R^* \cap P$ is contained in Q_{t^*}. Assume, for the sake of reaching a contradiction, that $R^* \cap P$ contains a point a that is not in Q_{t^*}. See Figure 4. Therefore, a is contained in one of the 3-sided rectangles used to define Q_{t^*}, namely $R_{t^*}^{\rightarrow}$, $R_{t^*}^{\leftarrow}$, $R_{\overline{t^*}}^{\rightarrow}$, $R_{\overline{t^*}}^{\leftarrow}$. Let $\tilde{R} \in \{R_{t^*}^{\rightarrow}, R_{t^*}^{\leftarrow}, R_{\overline{t^*}}^{\rightarrow}, R_{\overline{t^*}}^{\leftarrow}\}$ be the 3-sided rectangle that contains a, let $\tilde{P} \in \{P_{t^*}^{\rightarrow}, P_{t^*}^{\leftarrow}, P_{\overline{t^*}}^{\rightarrow}, P_{\overline{t^*}}^{\leftarrow}\}$ be the set contained in $\tilde{R}$ and let $\tilde{q}$ be the point of $\tilde{P}$ furthest from the vertical line through t^*. Note that $\tilde{P}$ contains k points, as otherwise there cannot be any point of P in $\tilde{R} \setminus \tilde{P}$ and a cannot exist. By the way we selected the points of $\tilde{P}$ inside $\tilde{R}$ we have

$$|t_x^* - \tilde{q}_x| \;<\; |t_x^* - a_x|.$$

Here we are using general position to rule out the possibility of equality. Note that the bounding box $\text{bb}(\tilde{P})$ of $\tilde{P}$ contains $k \geqslant k'$ points and has area at most

$$|t_x^* - \tilde{q}_x| \cdot \text{dist}(t^*, \ell),$$

where $\text{dist}(t^*, \ell)$ denotes the vertical distance from t^* to the line ℓ. On the other hand, since R^* intersects ℓ and has a and t^* in its boundary, we have

$$\text{area}(R^*) \;\geqslant\; |t_x^* - a_x| \cdot \text{dist}(t^*, \ell) \;>\; |t_x^* - \tilde{q}_x| \cdot \text{dist}(t^*, \ell) \;\geqslant\; \text{area}(\text{bb}(\tilde{P})).$$

This contradicts the optimality of R^* for covering k' points. This finishes the proof that $R^* \cap P$ is contained in Q_{t^*}, and therefore the second property holds.

It remains to show that the construction of the sets Q_p, for all $p \in P$, can be done in $O(kn + n \log n)$ time. For this we use Lemma 1 a few times, as follows. Let P^+ and P^- be the points above and below ℓ, respectively. We also define $\overline{P^+} = \{\overline{p} \mid p \in P^+\}$ and $\overline{P^-} = \{\overline{p} \mid p \in P^-\}$. The point sets $P_q^{\rightarrow}$ for all $q \in P^- \cup \overline{P^+}$, are obtained using Lemma 1 with $A = P^-$ and $B = P^- \cup \overline{P^+}$. The sets $P_q^{\rightarrow}$ for all $q \in P^+ \cup \overline{P^-}$, the sets $P_q^{\leftarrow}$ for all $q \in P^- \cup \overline{P^+}$, and the sets $P_q^{\leftarrow}$ for all $q \in P^+ \cup \overline{P^-}$, are computed in a similar way, using symmetric versions of Lemma 1. □

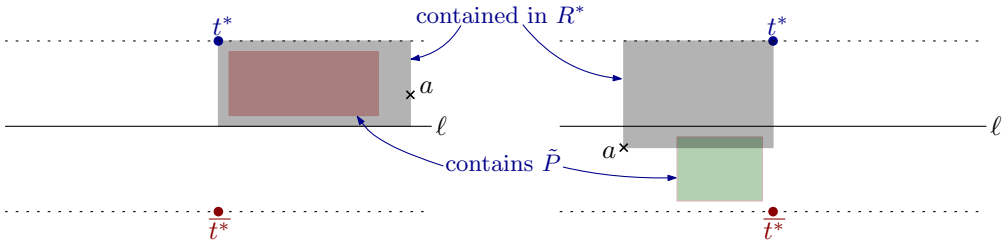

Figure 4: Part of the proof of Lemma 3 where we show that a point a outside $R^* \cap Q_{t^*}$ cannot exist. On the left we have the case when $\tilde{P} = P^{\rightarrow}_{t^*}$ and on the right the case $\tilde{P} = P^{\leftarrow}_{t^*}$.

Lemma 4. *Let P be a set of n points, let ℓ be a horizontal line, and let k be a positive integer. After $O(nk + n \log n)$ preprocessing time, we can compute, for any given $k' \leqslant k$, a value $V(P, \ell, k')$ with the following properties in $O(nk^2)$ time:*

- $\text{area}^*(P, k') \leqslant V(P, \ell, k')$;
- *if* $\text{area}^*(P, k') = \Psi(P, \ell, k')$, *then* $V(P, \ell, k') = \text{area}^*(P, k')$.

Proof. We compute the sets Q_p, indexed by $p \in P$, using Lemma 3. This finishes the preprocessing and takes time $O(kn + n \log n)$ time.

Now suppose that we are given a value $k' \leqslant k$. For each $p \in P$, we use Lemma 2 to find the value $\Phi(Q_p, p, k')$ in $O(|Q_p|^2) = O(k^2)$ time. We return the value $V(P, \ell, k') = \min\{\Phi(Q_p, p, k') \mid p \in P\}$. The computation takes $O(nk^2)$ time.

Since for each $p \in P$ the value $\Phi(Q_p, p, k')$ is the area of a box containing k' points of P, we have $V(P, \ell, k') \geqslant \text{area}^*(P, k')$. If $\text{area}^*(P, k') = \Psi(P, \ell, k')$, then Lemma 3 guarantees that $\text{area}^*(P, k') = \Phi(Q_{p_0}, p_0, k')$ for some $p_0 \in P$, and therefore

$$\text{area}^*(P, k') \;=\; \Phi(Q_{p_0}, p_0, k') \;\geqslant\; \min\{\Phi(Q_p, p, k') \mid p \in P\} \;=\; V(P, \ell, k').$$

We conclude that $V(P, \ell, k') = \text{area}^*(P, k')$. □

Theorem 5. *Given a set of n points P and a value k, we can preprocess P in $O(nk \log n + n \log^2 n)$ time such that, for any given $k' \leqslant k$, we can find in $O(nk^2 \log n)$ time a minimum-area box that contains at least k' points of P.*

Proof. Consider a horizontal line ℓ such that at most half of the points of P are above ℓ and at most half of the points are below ℓ. Let P^+ and P^- be the subset of P above and below ℓ, respectively. For any number of points k', where $1 \leqslant k' \leqslant n$ we have

$$\text{area}^*(P, k') \;=\; \min\,\{\; \Psi(P, \ell, k'),\; \text{area}^*(P^+, k'),\; \text{area}^*(P^-, k')\;\}.$$

Indeed, an optimal solution containing k' points is either intersected by ℓ or it contains points from only one of the sets P^+ and P^-. This is the basis for an algorithm based on divide and conquer.

In the preprocessing, we use Lemma 4 for P, ℓ and k, which takes $O(nk+n\log n)$ time, and then recursively preprocess P^+ and P^-. Since the recursion has $\log n$ levels and since any two point sets at the same level of the recursion are disjoint, we spend $O(nk\log n + n\log^2 n)$ time in preprocessing.

When we are given a value k', we compute $\text{area}^*(P,k')$ using the same recursive pattern. At the first level, with the point set P and the line ℓ, we spend $O(nk^2)$ time to compute $V(P,\ell,k')$, using Lemma 4. Then we go on to compute $\text{area}^*(P^+,k')$ and $\text{area}^*(P^-,k')$ recursively, using our already-done preprocessing. Finally, we return the minimum of $V(P,\ell,k')$, $\text{area}^*(P^+,k')$ and $\text{area}^*(P^-,k')$. By Lemma 4, we always have $\text{area}^*(P,k') \leqslant V(P,\ell,k')$ and, when $\text{area}^*(P,k') = \Psi(P,\ell,k')$, we also have $\text{area}^*(P,k') = V(P,\ell,k')$. It follows that

$$\text{area}^*(P,k') \;=\; \min\left\{\, V(P,\ell,k'),\ \text{area}^*(P^+,k'),\ \text{area}^*(P^-,k') \,\right\}$$

and thus we are returning the correct value of $\text{area}^*(P,k')$. Since we have $\log n$ levels in the recursion, we spend $O(nk^2\log n)$ time in total. □

Corollary 6. *Given a set of n points P and a value k we can find in $O(nk^2\log n + n\log^2 n)$ time a minimum-area box that contains at least k points of P.*

Proof. We apply Theorem 5 with $k' = k$. In this scenario we can get rid of the preprocessing step and, at each level of the recursion, compute the values $\Phi(Q_p,p,k)$ immediately after generating the sets Q_p. □

3 Maximizing the number of points for a given area

We now turn to the problem of finding the maximum number of points that can be covered by a box of a area $\alpha > 0$. As mentioned in the introduction, let $\kappa^*(P,\alpha)$ be this number of points. We first compute a constant-factor approximation to $\kappa^*(P,\alpha)$. Then we explain how to obtain a $(1+\varepsilon)$-approximation using an algorithm whose running time depends on the value $\kappa^*(P,\alpha)$. Finally, we use random sampling to get a $(1+\varepsilon)$-approximation to $\kappa^*(P,\alpha)$ in near-linear time for a fixed $\varepsilon > 0$.

3.1 A 4-approximation algorithm

For a horizontal line ℓ and a point $p \notin \ell$, let $R^{\rightarrow}_{\alpha}(p,\ell)$ be the box that has area α, has p as a corner, has an edge contained in ℓ, and contains points with x-coordinates larger than p_x. Let $R^{\leftarrow}_{\alpha}(p,\ell)$ be the box defined in the same way, but with points with x-coordinates smaller than p_x. See Figure 5. Let $\mathcal{R}_\alpha(\ell)$ be the set of boxes $\bigcup_{p\in P}\{R^{\rightarrow}_{\alpha}(p,\ell), R^{\leftarrow}_{\alpha}(p,\ell)\}$. Let $\kappa^*(P,\ell,\alpha)$ be the maximum number of points of P covered by a box of area α that intersects the line ℓ.

Lemma 7. *There is some $R \in \mathcal{R}_\alpha(\ell)$ such that $|P\cap R| \geqslant \kappa^*(P,\ell,\alpha)/4$.*

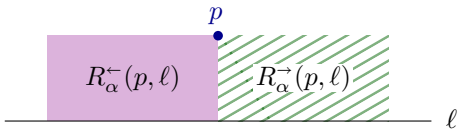

Figure 5: The boxes $R^{\leftarrow}_{\alpha}(p,\ell)$ and $R^{\rightarrow}_{\alpha}(p,\ell)$, the boxes have area α.

Proof. Let R^* be a box of area α covering $\kappa^*(P,\ell,\alpha)$ points and intersecting ℓ. Let ℓ^+ and ℓ^- denote the closed half-planes above and below ℓ, respectively. Define $R^+ := \mathcal{R}^* \cap \ell^+$ and $R^- := \mathcal{R}^* \cap \ell^-$, and assume without loss of generality that $|R^+ \cap P| \geqslant |R^* \cap P|/2$. Let p be the point in $P \cap R^*$ with maximum y-coordinate. Then $R^+ \subset R^{\rightarrow}_{\alpha}(p,\ell) \cup R^{\leftarrow}_{\alpha}(p,\ell)$. Hence, at least one of $R^{\rightarrow}_{\alpha}(p,\ell)$ or $R^{\leftarrow}_{\alpha}(p,\ell)$ must contain $|R^+ \cap P|/2 \geqslant |R^* \cap P|/4$ points. □

Theorem 8. *Given a set of n points P and a value $\alpha > 0$, we can compute in $O(n \log^2 n)$ time a value $\kappa(P,\alpha)$ such that $\kappa^*(P,\alpha)/4 \leqslant \kappa(P,\alpha) \leqslant \kappa^*(P,\alpha)$.*

Proof. We preprocess P for box counting queries [21]. The preprocessing takes $O(n \log n)$ time and for each query box R we can report $|R \cap P|$ in $O(\log n)$ time.

Then we proceed with a recursive algorithm. Take a line ℓ that splits P into two sets P^+ and P^- of roughly equal size. Note that

$$\kappa^*(P,\alpha) \;=\; \max\,\{\,\kappa^*(P^+,\alpha),\ \kappa^*(P^-,\alpha),\ \kappa^*(P,\ell,\alpha)\,\}.$$

We build the set of boxes $\mathcal{R}_\alpha(\ell)$ in $O(n)$ time. For each box $R \in \mathcal{R}_\alpha(\ell)$ we query the data structure to obtain $|R \cap P|$. Thus, we obtain $\kappa(P,\ell,\alpha) = \max\{|R \cap P| \mid R \in \mathcal{R}_\alpha(\ell)\}$ in $O(n \log n)$ time. By Lemma 7, we have $\kappa^*(P,\ell,\alpha)/4 \leqslant \kappa(P,\ell,\alpha) \leqslant \kappa^*(P,\ell,\alpha)$. Then, we return the best between the value $\kappa(P,\ell,\alpha)$ and the values $\kappa(P^+,\alpha), \kappa(P^-,\alpha)$ obtained recursively for P^+ and P^-, respectively. Since at each level of the recursion the point sets being considered are disjoint, we spend $O(n \log n)$ time at each level of the recursion, for a total $O(n \log^2 n)$ for the whole algorithm.

Since the algorithm only considers boxes of area α, the returned value is obviously at most $\kappa^*(P,\alpha)$. On the other hand, by induction we have $\kappa(P^+,\alpha) \geqslant \kappa^*(P^+,\alpha)/4$ and $\kappa(P^-,\alpha) \geqslant \kappa^*(P^-,\alpha)/4$. Together with $\kappa(P,\ell,\alpha) \geqslant \kappa^*(P,\ell,\alpha)/4$ we obtain that

$$\begin{aligned}
\kappa(P,\alpha) &= \max\{\kappa(P^+,\alpha),\ \kappa(P^-,\alpha),\ \kappa(P,\ell,\alpha)\} \\
&\geqslant \max\{\kappa^*(P^+,\alpha)/4,\ \kappa^*(P^-,\alpha)/4,\ \kappa^*(P,\ell,\alpha)/4\} \\
&= \kappa^*(P,\alpha)/4.
\end{aligned}$$

□

3.2 Properties of random sampling

For the rest of this section, let P be a set of n points in the plane, and let ε be a real value with $0 < \varepsilon < 1$. We use relative approximations [15] to show that a random sample from P can be used to count the points of P inside each box, assuming that the box has enough points. We use s for the cardinality of the sample.

Lemma 9. *Suppose that κ satisfies $\kappa^*(P,\alpha) \leqslant \kappa$. Let $s = \min\{n, \frac{c}{\varepsilon^2}\frac{n}{\kappa}\log n\}$, where c is an appropriate absolute constant, and let S be a random sample of P with s points. Then with probability at least $1 - 1/n$ the following properties hold simultaneously:*

- *For each box R of area at most α*
$$\left|\frac{|P\cap R|}{n} - \frac{|S\cap R|}{s}\right| \;\leqslant\; \varepsilon\cdot\frac{\kappa}{n};$$
- $\kappa^*(S,\alpha) = O\big((1/\varepsilon^2)\log n\big)$.

Proof. One can prove this using Chernoff bounds as we did in our first preprint [9]. A more compact proof uses relative approximations, as described next.

We consider the case when $s = \frac{c}{\varepsilon^2}\frac{n}{\kappa}\log n$. For the other case we have $S = P$ and $\kappa < (c/\varepsilon^2)\log n$, so the claims trivially hold.

Let $\mathcal{R}$ be the family of all boxes of area at most α. A subset $S \subseteq P$ is a *relative (ρ,ε)-approximation* for $(P,\mathcal{R})$ if
$$\forall R\in\mathcal{R}: \quad \left|\frac{|P\cap R|}{|P|} - \frac{|S\cap R|}{|S|}\right| \;\leqslant\; \varepsilon\cdot\max\left\{\frac{|P\cap R|}{|P|},\rho\right\}.$$
Har-Peled and Sharir [15, Theorem 2.11] show that the results of Li et al. [17] imply the following: A random sample of P of size at least
$$\frac{c'}{\varepsilon^2\rho}\left(\delta\log\frac{1}{\rho} + \log\frac{1}{q}\right),$$
where c' is an appropriate absolute constant, is a relative (ρ,ε)-approximation for $(P,\mathcal{R})$ with probability at least $1-q$. Here δ is the VC-dimension of the range space $(P,\mathcal{R})$; in our case $\delta \leqslant 4$. Setting $\rho = \kappa/n$ and $q = 1/n$, we obtain that a random sample of size at least
$$s = \frac{c'}{\varepsilon^2\rho}\left(\delta\log\frac{1}{\rho} + \log\frac{1}{q}\right) \;\leqslant\; \frac{c'n}{\varepsilon^2\kappa}\left(4\log\frac{n}{\kappa} + \log n\right) \;\leqslant\; \frac{5c'}{\varepsilon^2\kappa}n\log n$$
is a relative $(\kappa/n,\varepsilon)$-approximation for $(P,\mathcal{R})$ with probability at least $1-1/n$. The constant c in the statement of the lemma is then $5c'$.

It remains to show that, if S is a relative $(\kappa/n,\varepsilon)$-approximation for $(P,\mathcal{R})$, then both properties in the lemma hold. Since S is a relative $(\kappa/n,\varepsilon)$-approximation we have
$$\forall R\in\mathcal{R}: \quad \left|\frac{|P\cap R|}{n} - \frac{|S\cap R|}{s}\right| \;\leqslant\; \varepsilon\cdot\max\left\{\frac{|P\cap R|}{n},\frac{\kappa}{n}\right\} = \varepsilon\cdot\frac{\kappa}{n},$$
where in the last step we used $|P\cap R| \leqslant \kappa^*(P,\alpha) \leqslant \kappa$. This shows the first item.

For the second item we note that, for any box R of area at most α, we have
$$|S\cap R| \;\leqslant\; s\big(\frac{|P\cap R|}{n} + \varepsilon\cdot\frac{\kappa}{n}\big) \;\leqslant\; \frac{s}{n}\big(\kappa+\varepsilon\kappa\big) = (1+\varepsilon)\frac{s\kappa}{n} = (1+\varepsilon)\frac{c}{\varepsilon^2}\log n.$$
It follows that $\kappa^*(S,\alpha) = O\big((1/\varepsilon^2)\log n\big)$. □

3.3 A $(1-\varepsilon)$-approximation algorithm

We start by giving an output-sensitive $(1-\varepsilon)$-approximation algorithm whose running time depends quadratically on the size of the output.

Lemma 10. *Given a set P of n points, a value $\alpha > 0$, and a parameter ε with $0 < \varepsilon < 1$, we can compute in $O\left(n(\kappa^*)^2 \log n \log(1/\varepsilon) + n\log^2 n\right)$ time a box R of area at most α that covers at least $(1-\varepsilon)\kappa^*$ points of P, where $\kappa^* = \kappa^*(P,\alpha)$.*

Proof. Using Theorem 8 we compute a 4-approximation value κ_a satisfying $\kappa^*/4 \leqslant \kappa_a \leqslant \kappa^*$. We apply Theorem 5 with the value $4\kappa_a$, which is an upper bound for κ^*. We spend $O(n\kappa_a \log n + n\log^2 n) = O(n\kappa^* \log n + n\log^2 n)$ time in the preprocessing and then, for any given $k' \leqslant \kappa_a$, we can compute $\text{area}^*(P,k')$ in $O\left(n(\kappa^*)^2 \log n\right)$ time.

Consider the set K of values of the form $\kappa_a + i\cdot\varepsilon\kappa_a$, with $i \in \mathbb{N}$, inside the interval $[\kappa_a, 4\kappa_a]$. We perform binary search on K to find the value $\tilde{k} \in K$ such that $\text{area}^*(P,\tilde{k}) \leqslant \alpha$ but $\text{area}^*(P,\tilde{k}+\varepsilon\kappa_a) > \alpha$. We then have

$$\kappa_a \;\leqslant\; \tilde{k} \;\leqslant \kappa^* \;\leqslant\; \tilde{k} + \varepsilon\kappa_a \;\leqslant\; \tilde{k} + \varepsilon\kappa^*,$$

and thus

$$\tilde{k} \;\geqslant\; \kappa^* - \varepsilon\kappa^* \;=\; (1-\varepsilon)\kappa^*.$$

Since K has $O(1/\varepsilon)$ values, the binary search performs $O(\log(1/\varepsilon)$ steps. At each step we have to compute $\text{area}^*(P,k')$ for some value $k' \leqslant 4\kappa_a$, which takes $O\left(n(\kappa^*)^2 \log n\right)$ time. In total, we spend $O\left(n(\kappa^*)^2 \log n \log(1/\varepsilon)\right)$ time after $O(n\kappa^* \log n + n\log^2 n)$ preprocessing time. □

Theorem 11. *Given a set of n points P in the plane and a value $\alpha > 0$, let $\kappa^*(P,\alpha)$ be the maximum number of points from P that can be covered with a box of area α. Given a parameter ε, where $0 \leqslant \varepsilon \leqslant 1/2$, with probability at least $1 - 1/n$ we can find in $O\left((n/\varepsilon^4)\log^3 n \log(1/\varepsilon)\right)$ time a box $\tilde{R}$ of area α that covers at least $(1-\varepsilon)\kappa^*(P,\alpha)$ points from P.*

Proof. Using Theorem 8 we compute in $O(n\log^2 n)$ time a value κ_a satisfying

$$\kappa^*(P,\alpha)/4 \;\leqslant\; \kappa_a \;\leqslant\; \kappa^*(P,\alpha).$$

Set $\kappa = 4\kappa_a$, so that $\kappa^*(P,\alpha) \leqslant \kappa$, set $s = \min\{n, \frac{c}{\varepsilon^2}\frac{n}{\kappa}\log n\}$, and take a sample S of P with s points. Henceforth we assume that S satisfies the properties of Lemma 9, which occurs with probability at least $1 - 1/n$.

Using Lemma 10 for the sample S, we compute a box $\tilde{R}$ of area α covering at least $(1-\varepsilon)\kappa^*(S,\alpha)$ points of S. We return $\tilde{R}$.

Let us analyze the running time of the algorithm. By Lemma 9, we have $\kappa^*(S,\alpha) = O\big((1/\varepsilon^2)\log n\big)$. This means that the algorithm of Lemma 10 takes time

$$O\left(|S|\,(\kappa^*(S,\alpha))^2 \log|S| \log(1/\varepsilon) + |S|\log^2|S|\right).$$

Substituting the value $\kappa^*(S,\alpha) = O\big((1/\varepsilon^2)\log n\big)$ and $|S| \leqslant n$ we get the time bound

$$O\left(|S|\left((1/\varepsilon^2)\log n\right)^2 \log|S| \log(1/\varepsilon) + |S|\log^2|S|\right) \;=\; O\left((n/\varepsilon^4)\log^3 n \log(1/\varepsilon)\right).$$

Finally, we analyze the approximation error. Let R^* be an optimal solution for P: A box of area α that contains $\kappa^*(P,\alpha)$ points of P. Since $\tilde{R}$ covers $(1-\varepsilon)\kappa^*(S,\alpha)$ points of the sample S, Lemma 9 implies that

$$\begin{aligned}
\frac{|P\cap\tilde{R}|}{n} &\geqslant \frac{|S\cap\tilde{R}|}{s} - \varepsilon\frac{\kappa}{n} \\
&\geqslant \frac{(1-\varepsilon)\kappa^*(S,\alpha)}{s} - \varepsilon\frac{\kappa}{n} \\
&\geqslant (1-\varepsilon)\frac{|S\cap R^*|}{s} - \varepsilon\frac{\kappa}{n} \\
&\geqslant (1-\varepsilon)\frac{|P\cap R^*|}{n} - \varepsilon\frac{\kappa}{n} - \varepsilon\frac{\kappa}{n} \\
&= (1-\varepsilon)\frac{\kappa^*(P,\alpha)}{n} - 2\varepsilon\frac{\kappa}{n}.
\end{aligned}$$

This means that

$$|P\cap\tilde{R}| \geqslant (1-\varepsilon)\kappa^*(P,\alpha) - 2\varepsilon\kappa,$$

and using that $\kappa = 4\kappa_a \leqslant 4\kappa^*(P,\alpha)$, we get

$$|P\cap\tilde{R}| \geqslant (1-\varepsilon)\kappa^*(P,\alpha) - 2\varepsilon(4\kappa^*(P,\alpha)) \geqslant (1-9\varepsilon)\kappa^*(P,\alpha).$$

Repeating the analysis with $\varepsilon' = \varepsilon/9$ in place of ε, the result follows. □

3.4 Can we make the algorithm deterministic?

To make our approximation algorithm deterministic, we would need a deterministic construction of a relative (ρ,ε)-approximation with respect to boxes. It is currently unclear if a relative approximation of the desired size can be computed deterministically in an efficient manner. Another option would be to use ε-approximations for boxes. Given a set of points P in the plane, a subset $A \subseteq P$ is a δ-approximation[1] with respect to boxes if

$$\forall \text{ boxes } R: \quad \left|\frac{|R\cap P|}{|P|} - \frac{|R\cap A|}{|A|}\right| \;\leqslant\; \delta.$$

δ-approximations are good for counting the number of points inside each box with an error of $\delta|P|$. After we have a constant-factor approximation κ_a to the value $\kappa^*(P,\alpha)$, we could thus use a δ-approximation with $\delta = \varepsilon\kappa_a$. There are δ-approximations with respect to boxes of size roughly $O(1/\delta)$, which would be better than the random sample we are currently using. However, constructing such a δ-approximation takes roughly $O(n/\delta^3)$ time; see Phillips [19] for the latest results. For example, when $\kappa_a = \Theta(\kappa^*) = \Theta(\sqrt{n})$, this means that we need roughly $\tilde{O}(n^{5/2})$ time. Thus, building δ-approximations deterministically in near-linear time is the current bottleneck for this approach.

[1] We use δ rather than ε here to avoid confusion with the different roles of ε.

4 Conclusions

There are several avenues that can be pursued to improve our results:

Improving Lemma 2 directly improves our time bounds for computing area$^*(P,k)$. One approach would be to reduce the problem of Lemma 2 to the following problem: Maintain a set of $O(k)$ points on the real line under insertions such that, after each insertion, we can recover the smallest interval that contains k of the points. Moreover, we know the order of the insertions in advance. If we can handle each insertion in $o(k)$ time, then the result in Lemma 2 can be improved, and consequently area$^*(P,k)$ can be computed faster.

One may also try to compute the k values area$^*(P,1)$, ..., area$^*(P,k)$ faster than using the algorithm for each $k' \in \{1,\ldots,k\}$ independently. In particular, if in Lemma 2 we could compute all the values $\Phi(Q,q,1), \Phi(Q,q,2), \ldots, \Phi(Q,q,|Q|)$ in $o(|Q|^3)$ time, then a better algorithm could be obtained for this problem.

The following additional open problems remain:

- Is it possible to derandomize the algorithm described in Section 3 (see the discussion in Section 3.4)?.
- In $\mathbb{R}^3$, can we find the smallest box covering k points in time roughly $O(nk^3)$? Note that any running time of the form $\tilde{O}(nk^c)$, for some constant c, would lead to a near-linear-time randomized $(1-\varepsilon)$-approximation algorithm for the dual problem of covering as many points as possible with a box of given volume.
- Is there a non-trivial lower bound, such as $\Omega(nk)$, for computing area$^*(P,k)$ exactly?

Acknowledgments

Some parts of this work have been done at the Fourth Annual Workshop on Geometry and Graphs, held at the Bellairs Research Institute in Barbados, March 6–11, 2016. The authors are grateful to the organizers and to the participants of the workshop, especially to Luis Barba for suggesting the problem. We would also like to thank Xavier Goaoc for fruitful discussions on the subject.

References

[1] Pankaj K. Agarwal, Torben Hagerup, Rahul Ray, Micha Sharir, Michiel H. M. Smid, and Emo Welzl. Translating a planar object to maximize point containment. In *Proc. 10th Annual European Symposium (ESA)*, volume 2461 of *Lecture Notes in Computer Science*, pages 42–53. Springer, 2002.

[2] Alok Aggarwal, Hiroshi Imai, Naoki Katoh, and Subhash Suri. Finding k points with minimum diameter and related problems. *J. Algorithms*, 12(1):38–56, 1991.

[3] Hee-Kap Ahn, Sang Won Bae, Erik D. Demaine, Martin L. Demaine, Sang-Sub Kim, Matias Korman, Iris Reinbacher, and Wanbin Son. Covering points by disjoint boxes with outliers. *Comput. Geom.*, 44(3):178–190, 2011.

[4] Boris Aronov, Esther Ezra, and Micha Sharir. Small-size ε-nets for axis-parallel rectangles and boxes. *SIAM J. Comput.*, 39(7):3248–3282, 2010.

[5] Boris Aronov and Sariel Har-Peled. On approximating the depth and related problems. *SIAM J. Comput.*, 38(3):899–921, 2008.

[6] Sergio Cabello, José Miguel Díaz-Báñez, and Pablo Pérez-Lantero. Covering a bichromatic point set with two disjoint monochromatic disks. *Comput. Geom.*, 46(3):203–212, 2013.

[7] Sandip Das, Partha P. Goswami, and Subhas C. Nandy. Smallest k-point enclosing rectangle and square of arbitrary orientation. *Inf. Process. Lett.*, 94(6):259–266, 2005.

[8] Amitava Datta, Hans-Peter Lenhof, Christian Schwarz, and Michiel H. M. Smid. Static and dynamic algorithms for k-point clustering problems. *J. Algorithms*, 19(3):474–503, 1995.

[9] Mark de Berg, Sergio Cabello, Otfried Cheong, David Eppstein, and Christian Knauer. Covering many points with a small-area box. *CoRR*, abs/1612.02149, 2016.

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

[11] Alon Efrat, Micha Sharir, and Alon Ziv. Computing the smallest k-enclosing circle and related problems. *Comput. Geom.*, 4:119–136, 1994.

[12] David Eppstein and Jeff Erickson. Iterated nearest neighbors and finding minimal polytopes. *Discrete Comput. Geom.*, 11(3):321–350, 1994.

[13] Sariel Har-Peled and Soham Mazumdar. Fast algorithms for computing the smallest k-enclosing circle. *Algorithmica*, 41(3):147–157, 2005.

[14] Sariel Har-Peled and Benjamin Raichel. Net and prune: A linear time algorithm for Euclidean distance problems. *J. ACM*, 62(6):44, 2015.

[15] Sariel Har-Peled and Micha Sharir. Relative (p, ε)-approximations in geometry. *Discrete & Computational Geometry*, 45(3):462–496, 2011.

[16] Haim Kaplan, Sasanka Roy, and Micha Sharir. Finding axis-parallel rectangles of fixed perimeter or area containing the largest number of points. In *Proc. 25th Annual European Symposium (ESA)*, volume 87 of *LIPIcs*, pages 52:1–52:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[17] Yi Li, Philip M. Long, and Aravind Srinivasan. Improved bounds on the sample complexity of learning. *J. Comput. Syst. Sci.*, 62(3):516–527, 2001.

[18] Jiří Matoušek. On enclosing k points by a circle. *Inf. Process. Lett.*, 53(4):217–221, 1995.

[19] Jeff M. Phillips. Algorithms for ε-approximations of terrains. In *Proc. 35th International Colloquium, Automata, Languages and Programming, (ICALP)*, volume 5125 of *Lecture Notes in Computer Science*, pages 447–458. Springer, 2008.

[20] Michael Segal and Klara Kedem. Enclosing k points in the smallest axis parallel rectangle. *Inform. Process. Lett.*, 65(2):95–99, 1998.

[21] Dan E. Willard. New data structures for orthogonal range queries. *SIAM J. Comput.*, 14(1):232–253, 1985.