Covering a set of points by two axis-parallel boxes

Sergei Bespamyatnikh

Department of Computer Science, University of British Columbia, Vancouver, B.C. Canada V6T 1Z4

Michael Segal

Department of Mathematics and Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

Abstract

In this paper we consider the following covering problem. Given a set S of n points in d-dimensional space, $d \geq 2$, find two axis-parallel boxes that together cover the set S and minimize the maximum of measures of boxes, where the measure is a monotone function of the box. We present a simple algorithm for finding boxes in $O(n \log n + n^{d-1})$ time using linear space.

1 Introduction

We consider the following *min-max two box* problem. Given a set S of n points in d-dimensional space, $d \geq 2$, find two axis-parallel boxes b_1 and b_2 that together cover the set S and minimize the maximum of measures $\mu(b_1)$ and $\mu(b_2)$, where μ is a monotone function of the box, i.e. $b_1 \subseteq b_2$ implies $\mu(b_1) \leq \mu(b_2)$. Examples of the box measure μ are the volume of the box, the perimeter of the box, the length of the diagonal etc. We assume that the dimension d is fixed and the measure of the box can be computed in $O(1)$ time.

The min-max two box problem is the "covering problem" such as the classical problem of finding the minimum enclosed circle [9]. On other hand it belongs to "partition problems" where we are interested in partitioning a set of points into two subsets in such a way to optimize some function of the "sizes" of two subsets [1,4,7,8].

This problem is also closely related to the rectilinear p-center problems (and in particular to the 2-center problem). The p-center problem is defined as follows.

Let $\mathcal{R}$ be the set of compact convex regions with nonempty interior in the plane, where every region $r \in \mathcal{R}$ is assigned a *scaling point* c_r in its interior. For $r \in \mathcal{R}$ and a real number $\lambda \geq 0$, let $r(\lambda)$ be the homeothetic copy of r obtained by scaling r by the factor λ about c_r (i.e., $r(\lambda) = \{c_r + \lambda(a - c_r) | a \in r\}$). Finally, $\mathcal{R}(\lambda) = \{\ r(\lambda) |\ r \in \mathcal{R}\}$. The p-center problem for $\mathcal{R}$ looks for

$$\lambda_{\mathcal{R}} = min\{\ \lambda | \mathcal{R}(\lambda) \text{ is } p\text{-pierceable }\}.$$

We call set $\mathcal{R}$ p-pierceable if there exist a set of p points that intersects each member of $\mathcal{R}$. If $\mathcal{R}$ is a set of translates of a square and the scaling points are the respective centers, then we talk about the *rectilinear p-center problem*. If the squares are still axis-parallel but of possible different sizes, then we have the *weighted rectilinear p-center problem*, and if $\mathcal{R}$ is a set of arbitrary axis-parallel rectangles (and the scaling points are also arbitrary), then we face the *general rectilinear p-center problem*.

Results for the above defined problems, for $p = 2$ and $d = 2$, were obtained by [4,3,12,11]. Hershberger and Suri [4] solve the following clustering problem: Given a planar set of points S, a *rectangular measure* μ acting on S and a pair of values μ_1 and μ_2, does there exist a bipartition $S = S_1 \cup S_2$ satisfying $\mu(S_i) \leq \mu_i$ for $i = (1, 2)$? They present an algorithm which solves this problem in $O(n \log n)$ time. Based on this algorithm and using the sorted matrix technique of Frederickson and Johnson [2], Glozman et al. [3] obtained an $O(n \log n)$ time algorithm that solves min-max box problem in the plane. In a very recent paper Sharir and Welzl [12] using LP-type framework and Helly-type results obtained an $O(n)$ expected time algorithm for the general rectilinear 2-center problem, where $d = 2$. The paper of Segal [10] solves the min-max two box problem in three dimensions. The runtime of the algorithm in [10] is $O(n^2 \log n)$.

In this paper we present an efficient algorithm for solving the min-max two box problem in arbitrary dimension $d \geq 2$. The runtime of the algorithm is $O(n \log n + n^{d-1})$. For the plane, the algorithm is actually simpler than one of Glozman et. al because only data structure it used is a list and it takes linear time after presorting.

This paper is organized as follows. Section 2 introduces some notations. Section 3 shows the algorithm for the plane. In Section 4 we present our algorithm for dimension $d \geq 3$. We conclude in Section 5.

2 Preliminaries

Given a set $S = \{p_1, \ldots, p_n\}$ of n points in $R^d, d > 2$, find two axis-parallel boxes b_1 and b_2 that together cover the set S and minimize the expression: $\max(\mu(b_1), \mu(b_2))$, where μ is a monotone function of the box, i.e. $\mu(b_1) \leq \mu(b_2)$ if $b_1 \subseteq b_2$. For simplicity we consider the general case of distinct coordinates,

i.e. the projection S onto any coordinate axis is a set of n distinct points. Initially we sort the points of S according to each of the coordinates.

Given a set of points S, the *bounding box* of S, denoted by $bb(S)$, is the smallest axis-parallel box that contains S. The bounding box of S is determined by $2d$ points, two from each axis i, $i = 1, \ldots, d$: the leftmost point $l_i(S)$ of S, and the rightmost point $r_i(S)$. We call these points the *determinators* of S. For a box $b = [l_1, r_1] \times \ldots \times [l_d, r_d]$, the points l_i and r_i are also called the *determinators of* b. For a point p, let $x(p)$ and $y(p)$ denote the coordinates of p in the first and the second axes respectively.

3 The algorithm for the plane

Glozman et al. [3] consider three different ways to partition the determinators between two subsets of S

(1) One of the subsets gets three determinators and the other gets one determinator.
(2) Each subset gets two determinators lying on opposite sides of $bb(S)$.
(3) Each subset gets two determinators lying on adjacent sides of $bb(S)$.

We distinguish only 2 cases

(1) One subset gets two determinators lying on adjacent sides of $bb(S)$.
(2) Each subset gets two determinators lying on opposite sides of $bb(S)$.

The algorithm finds covering boxes for the both cases and returns the pair with minimum measure μ. The algorithm for the first case is essentially the same as the algorithm for the problem P2 which is described in Section 4. So it is sufficient to explain the algorithm for the second case.

Consider the box, say b_1, containing the uppermost and bottommost points of S. The second box b_2 contains the set S_2 of the points outside b_1. We can assume that it is the bounding box of these points, i.e. $b_2 = bb(S \setminus b_1)$. In other words two determinators (left and right) of the box b_1 define both boxes. Let l and r denote the left and right determinators of the box b_1. They can be taken among x-coordinates of points in S. The naive approach is to compute the measures of the boxes b_1 and b_2 for each pair of x-coordinates. It takes $\Omega(n^2)$ time in worst case. We can reduce the number of pairs l and r that are candidates for the solution. Let $L = (p_1, \ldots, p_n)$ be the list of points S sorted by x-coordinate. For each left determinator $l = x(p_i)$, the algorithm computes only two candidates $r_1(l)$ and $r_2(l)$. $r_1(l)$ is the largest x-coordinate such that the box b_1 with $r = r_1(l)$ has the measure not greater than the box b_2. $r_2(l)$ is

the smallest x-coordinate (if any) such that the box b_1 with $r = r_2(l)$ has the measure greater than the box b_2. Clearly, $r_1(l) < r_2(l)$ and $r_2(l)$ is the next element of L after $r_1(l)$ if it exists. One can show that r_1 and r_2 are monotone function, i.e. $r_1(l') \geq r_1(l)$ and $r_2(l') \geq r_2(l)$, if $l' > l$.

The monotonicity of r_1 and r_2 allows to obtain at most $4n-2$ times of computing the measure μ. To find r_1 and r_2 for $l = x(p_i)$ the algorithm starts with the value $r = r_2(x(p_{i-1}))$. If the box b_1 with determinators l and r has measure greater than corresponding box b_2, then $r_1(l)$ and $r_2(l)$ coincide with previous values. Otherwise the algorithm increases the pointer r until the box b_1 has the measure greater than the box b_2 or $r = p_n$. The move of the pointer r can be illustrated on the $n \times n$ matrix A. The element $A(i,j)$ is equal 1 if the algorithm compute the measure of the box b_1 with determinators p_i and p_j. Otherwise $A(i,j) = 0$. The unit elements of A corresponds to the path from $(1,1)$ to (n, n_0) where $n_0 = r_2(n)$ or $n_0 = n$ (if $r_2(n)$ does not exist). Each step of the motion of pointer is in left-right or down-up direction. The algorithm visits at most $2n-1$ cells. Each time it computes the measures of two boxes b_1 and b_2. So it measures at most $4n-2$ boxes. We give the formal description of the algorithm.

Algorithm TwoBoxes ($*$ Given points $p_1, \ldots, p_n$ sorted by x-coordinate.
Find the minimum of maximum measure of two boxes covering points such that one box gets the uppermost $y_{\max}$ and bottommost $y_{\min}$ determinators $*$)
$j := 1$
$S_2 := S$
solution:= $\mu_2 := \mu(bb(S_2))$
for $i := 1$ to n **do**
  $\mu_1 = \mu([p_i, p_j] \times [y_{\min}, y_{\max}])$
  if solution$<$ $\min(\mu_1, \mu_2)$ **then** solution:= $\min(\mu_1, \mu_2)$
  while $\mu_1 \leq \mu_2$ **do**
    if $j \geq n$
    then return solution
    $S_2 := S_2 \setminus \{p_j\}$
    $\mu_2 := \mu(bb(S_2))$
    $j := j + 1$
    $\mu_1 = \mu([p_i, p_j] \times [y_{\min}, y_{\max}])$
    if solution$<$ $\min(\mu_1, \mu_2)$ **then** solution:= $\min(\mu_1, \mu_2)$
  $S_2 := S_2 \cup \{p_j\}$
  $\mu_2 := \mu(bb(S_2))$
return solution

It remains to show how to compute the bounding box of set S_2. The set S_2 undergoes both insertions and deletions of points throughout the algorithm, but one point can be deleted from and inserted to S_2 only once. Initially $S_2 = S$. We partition S_2 into two subsets S_2' and S_2'' as defined below:

- S_2' contains the points that were not deleted from S_2
- $S_2'' = S_2 \setminus S_2'$.

The set S_2' is updated only by deletion of points. The set S_2'' is updated only by insertion of points. It is easy to maintain the box $bb(S_2'')$. Using presorting we can maintain the box $bb(S_2')$ in $O(1)$ time. The box $bb(S_2)$ can be computed in $O(1)$ time using the boxes $bb(S_2')$ and $bb(S_2'')$.

Clearly the algorithm takes linear time. So we prove the following theorem.

Theorem 3.1 *The min-max two box problem for n presorted points in the plane can be solved in linear time using linear space.*

4 The algorithm in higher dimensions

The main idea of the algorithm is the reduction of the dimension. We assume that the dimension d is greater than two and reduce it to two. Let the boxes b_1 and b_2 be the solution of the min-max two box problem. Then there is a box which has at least d determinators coinciding with the corresponding determinators of S. We can assume it is b_1. The box b_1 define the smallest box $b_2 = bb(S \setminus b_1)$ containing all points outside b_1. For simplicity we assume the box $b_2 = bb(S \setminus b_1)$ throughout this Section. The boxes b_1 and b_2 are the solution of the following problem.

Problem P1. Given a set S of n points and d determinators of the box b_1, and denote by $b_2 = bb(S \setminus b_1)$. Find the remaining d determinators of b_1 such that the expression $\max(\mu(b_1), \mu(b_2))$ is minimized.

For a set S, there are $\binom{2d}{d}$ ways to fix d determinators. So we have $\binom{2d}{d}$ problems P1. We solve all these problems. The solution of the min-max two box problem can be chosen from $\binom{2d}{d}$ pairs of boxes b_1 and b_2. (Of course we don't need to store all these pairs.)

Now we show how to solve the problem P1. The box b_1 has d fixed determinators and d free determinators. Since the dimension d is greater than 2, there are two free determinators on different axes. Without loss of generality they are right determinators on x and y axes, i.e. $r_1(b_1)$ and $r_2(b_1)$. We want to find these determinators and the remaining $d-2$ free determinators. At first the algorithm define the remaining determinators. They can be determined by points of S (for example i-th left determinator $l_i(b_1)$ is determined by point $p = (p_1, \ldots, p_d) \in S$ if $l_i(b_1) = p_i$). Let us consider all the $(d-2)$-tuples of points S. The algorithm goes through all these tuples and finds the

combination that attains the required minimum.

It is clear that the number of tuples is n^{d-2}. Note that some tuples cannot give the solution because the determinators of b_1 have to satisfy the inequality $l_i < r_i, i = 1, \ldots, d$.

The 2-dimensional problem can be now formulated as

Problem P2. Given a set S of n points and $2d-2$ determinators $l_1, \ldots, l_d$, $r_3, \ldots, r_d$ of the box b_1. Find two determinators r_1 and r_2 of the box b_1 that minimize the expression $\max(\mu(b_1), \mu(bb(S \setminus b_1)))$.

We show that the problem P2 can be solved in $O(n)$ time. The determinators r_1 and r_2 can be chosen from the sets $\{x(p_1), \ldots, x(p_n)\}$ and $\{y(p_1), \ldots, y(p_n)\}$ respectively. Let $b(x, y)$ denote the box $[l_1, x] \times [l_2, y] \times [l_3, r_3] \times \ldots \times [l_d, r_d]$, where $x \in \{x(p_1), \ldots, x(p_n)\}$ and $y \in \{y(p_1), \ldots, y(p_n)\}$. Let S_1 denote the set of points of S in the box $b(x, y)$ and $S_2 = S \setminus S_1$. For each $x \in \{x(p_1), \ldots, x(p_n)\}$, our algorithm finds the largest $y \in \{y(p_1), \ldots, y(p_n)\}$ such that $\mu(b(x, y)) \leq \mu(bb(S \setminus b(x, y)))$. Denote it by $Y_1(x)$. We observe that $Y_1(x)$ is *monotone*.

Observation 4.1 *$Y_1(x)$ is non-increasing function.*

Proof. Consider two x-coordinates $x' < x''$. It is clear that the box $B = b(x'', Y_1(x''))$ contains the box $C = b(x', Y_1(x''))$. Therefore the box $B' = bb(S \setminus B)$ is contained in the box $C' = bb(S \setminus C)$. This implies $\mu(B) \geq \mu(C)$ and $\mu(B') \leq \mu(C')$. $\mu(B) \leq \mu(B')$ by the definition Y_1. Hence $\mu(C) \leq \mu(B) \leq \mu(B') \leq \mu(C')$. It means that $Y_1(x'') \leq Y_1(x')$. ∎

For each $x \in \{x(p_1), \ldots, x(p_n)\}$ our algorithm (for the problem P2) finds the smallest $y \in \{y(p_1), \ldots, y(p_n), \infty\}$ such that $\mu(b(x, y)) > \mu(bb(S \setminus b(x, y)))$ (if it exists). Denote it by $Y_2(x)$.

Observation 4.2 *There exists a solution of problem P2 such that $r_2 = Y_1(r_1)$ or $r_2 = Y_2(r_1)$.*

Proof. Let the boxes $b_1 = b(r_1, r_2)$ and $b_2 = bb(S \setminus b_1)$ be the solution of the problem P2. If $\mu(b_1) \leq \mu(b_2)$ then $r_2 \leq Y_1(r_1)$. Using arguments like one of Observation 4.1 we can show

$$\mu(b_1) \leq \mu(b(r_1, Y_1(r_1))) \leq \mu(bb(S \setminus b(r_1, Y_1(r_1)))) \leq \mu(b_2).$$

We can enlarge the box b_1 to the box $b(r_1, Y_1(r_1))$. This gives a solution with $r_2 = Y_1(r_1)$.

In the case $\mu(b_1) > \mu(b_2)$ we can take $r_2 = Y_2(r_1)$. ∎

For each $x \in \{x(p_1), \ldots, x(p_n)\}$ and $y \in \{Y_1(x), Y_2(x)\}$, we compute $\max(\mu(b(x,y)), \mu(bb(S \setminus b(x,y))))$. Then we compute minimum value of these numbers. It gives the solution of the problem P2.

Now we explain how to achieve $O(n)$ running time. Let b_1^*, b_2^* be the pair of boxes which are candidates for the solution. Let us consider the moment when we have computed $Y_1(x)$ and $Y_2(x)$ for some x. Using the fact that the points of S are sorted separately in each of the coordinates we get the next value $x' > x$ in this order. For the point p of S with x-coordinate x', we perform the following operations.

First we check whether p can lie in b_1. If p cannot lie in b_1 (the i-th coordinate of p is less than l_i for some i, or the i-th coordinate of p is greater than r_i, for some $i > 2$) then p remains in S_2 and we pass it. Otherwise compare $y(p)$ and $Y_1(x)$. If $y(p) > Y_1(x)$ then p remains in S_2 (by monotonicity of Y_1) and we pass it. Otherwise we delete p from S_2 and insert it into S_1. For the determinators $r_1 = x'$ and $r_2 = Y_1(x)$ of the box b_1, compute $\mu_1 = \mu(b_1)$ and $\mu_2 = \mu(bb(S_2))$. If $\max(\mu(b_1), \mu(b_2)) < \max(\mu(b_1^*), \mu(b_2^*))$, then set $b_1^* = b_1$ and $b_2^* = b_2$. There are two possible cases.

Case 1: $\mu_1 \leq \mu_2$. In this case $Y_1(x') = Y_1(x)$ and $Y_2(x') = Y_2(x)$ by monotonicity of Y_1. It should be noted that we do not need to compute the rectangular measure μ for the pair x' and $Y_2(x)$ since it is greater than or equal to the measure of the boxes b_1 and b_2 determined by the pair x and $Y_2(x)$. We only have to compute the rectangular measure $\max(\mu_1, \mu_2)$ for the pair x' and $Y_1(x)$ and update the current solution if its measure is greater than $\max(\mu_1, \mu_2)$.

Case 2: $\mu_1 > \mu_2$. In this case $Y_1(x') < Y_1(x)$ and $Y_2(x') < Y_2(x)$. Let $y = Y_1(x)$. In order to find $Y_1(x')$ and $Y_2(x')$ we perform the following operations as long as $\mu_1 > \mu_2$.

- For each point $p \in S_1$ with $y(p) = y$, move the point p from S_1 to S_2.
- Using the sorted order of S points according to the y-coordinate we move to the next point whose y-coordinate is less than one we currently have.
- Compute $\mu_1 = \mu(b(x', y))$ and $\mu_2 = \mu(bb(S_2))$. If $\max(\mu(b_1), \mu(b_2)) < \max(\mu(b_1^*), \mu(b_2^*))$, then set $b_1^* = b_1$ and $b_2^* = b_2$.

After these operations we have $Y_1(x') = y$ and $Y_2(x')$ is the previous value of y. Now the processing x' is finished.

We start the algorithm at a point x that is less than the x-coordinate of all the points of S, for example $x = \min\{x(p_1), \ldots, x(p_n)\} - 1$. We set $Y_1(x) = \min\{y(p_1), \ldots, y(p_n)\}$. Initially we have $S_1 = \emptyset$, $\mu_1 = 0$, $S_2 = S$ and $\mu_2 = \mu(bb(S))$, Y_2 can have any value because it is used only after S_1 becomes non-empty.

Ignoring the time spent on computing μ_1 and μ_2 the algorithm takes $O(n)$ time. It remains to show how to compute μ_2. Recall $\mu_2 = \mu(bb(S_2))$. The maintenance of the bounding box of S_2 is described in Section 3.

We summarize with the following theorem

Theorem 4.3 *The min-max two box problem in d-dimensional space, $d \geq 2$, can be solved in time $O(n \log n + n^{d-1})$ using $O(n)$ space.*

Proof. We spend $O(n \log n)$ time in sorting S according to all its coordinates. As it was pointed above, we solve $\binom{2d}{d}$ problems P1 in order to obtain a solution for the min-max two box problem. Each problem P1 is solved by considering all the $(d-2)$-tuples of points S. There are n^{d-2} such tuples. For each such a tuple related to the problem P2 which we solve in $O(n)$ time. ∎

5 Conclusions

In this paper we present an efficient algorithm for solving min-max two box problem. The efficiency of the algorithm is based on the monotonicity of the evaluated function in the problem. It would be interesting to find some connection between this problem and the problems considered by Jaromczyk and Kowaluk [5] and Segal [6]. Computing lower bounds for this problem and problems in [5,6] are still open questions.

References

[1] Te. Asano, B. Bhattacharya, J. M. Keil, and F. Yao. Clustering algorithms based on minimum and maximum spanning trees. In *Proc. 4th Annu. ACM Sympos. Comput. Geom.*, pages 252–257, 1988.

[2] G. N. Frederickson and D. B. Johnson. The complexity of selection and ranking in $x+y$ and matrices with sorted columns. *J. Comput. Syst. Sci.*, pages 197–208, 1982.

[3] A. Glozman, K. Kedem, and G. Shpitalnik. On some geometric selection and optimization problems via sorted matrices. 955:26–37, 1995.

[4] J. Hershberger and S. Suri. Finding tailored partitions. *J. Algorithms*, pages 431–463, 1991.

[5] J. W. Jaromczyk and M. Kowaluk. Orientation independent covering of point sets in r^2 with pairs of rectangles or optimal squares. In *European Workshop on Comp. Geometry*, Muenster, Germany, 1996.

[6] M. Katz, K. Kedem, and M. Segal. Constrained square-center problems. In *Scandinavian Workshop on Algorithm Theory*, Lecture Notes in Computer Science, 1998.

[7] J. S. B. Mitchell and E. L. Wynters. Finding optimal bipartitions of points and polygons. In *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes Comput. Sci.*, pages 202–213. Springer-Verlag, 1991.

[8] C. Monma and S. Suri. Partitioning points and graphs to minimize the maximum or the sum of diameters. In *Graph Theory, Combinatorics and Applications (Proc. 6th Internat. Conf. Theory Appl. Graphs)*, volume 2, pages 899–912, New York, NY, 1991. Wiley.

[9] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, NY, 1985.

[10] M. Segal. *unpublished manuscript.*

[11] M. Segal. On piercing of axis-parallel rectangles and rings. In *European Symposium on Algorithms*, volume 1284 of *Lecture Notes in Computer Science*, pages 430–442. Springer-Verlag, 1997.

[12] Micha Sharir and Emo Welzl. Rectilinear and polygonal p-piercing and p-center problems. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 122–132, 1996.