International Journal of Computational Geometry & Applications

3-PIERCING OF d-DIMENSIONAL BOXES AND HOMOTHETIC TRIANGLES

EYAL ASSA*
School of Mathematical Sciences, Tel Aviv University, Tel-Aviv 69978, Israel

MATTHEW J. KATZ†
Department of Mathematics and Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

Received October 1997
Revised April 1998
Communicated by J. Mitchell

ABSTRACT

In the p-piercing problem, a set $\mathcal{S}$ of n d-dimensional objects is given, and one has to compute a piercing set for $\mathcal{S}$ of size p, if such a set exists. We consider several instances of the 3-piercing problem that admit linear or almost linear solutions: (i) If $\mathcal{S}$ consists of axis-parallel boxes in $\mathcal{R}^d$, then a piercing triplet for $\mathcal{S}$ can be found (if such a triplet exists) in $O(n \log n)$ time, for $3 \leq d \leq 5$, and in $O(n^{\lfloor \frac{d}{3} \rfloor} \log n)$ time, for $d \geq 6$. Based on the solutions for $3 \leq d \leq 5$, efficient solutions are obtained to the corresponding 3-center problem — Given a set $\mathcal{P}$ of n points in $\mathcal{R}^d$, compute the smallest edge length λ such that $\mathcal{P}$ can be covered by the union of three axis-parallel cubes of edge length λ. (ii) If $\mathcal{S}$ consists of homothetic triangles in the plane, or of 4-oriented trapezoids in the plane, then a piercing triplet can be found in $O(n \log n)$ time.

Keywords: geometric optimization, p-center, piercing set

1. Introduction

Let $\mathcal{S}$ be a set of n objects in $\mathcal{R}^d$, and let p be a positive integer. $\mathcal{S}$ is *p-pierceable* if there exists a set $\mathcal{A}$ of p points in $\mathcal{R}^d$ that intersects each of the objects in $\mathcal{S}$. We say that $\mathcal{A}$ is a *piercing set for $\mathcal{S}$*. The *p-piercing problem* is to determine whether $\mathcal{S}$ is p-pierceable, and, if so, to produce a piercing set of size p for $\mathcal{S}$. The p-piercing problem is **NP**-complete if p is part of the input [8], therefore we are mainly interested in instances of the problem in which p is a small constant.

The p-piercing problem is closely related to the well known *p-center problem* (given a set $\mathcal{P}$ of n points in $\mathcal{R}^d$, compute the smallest radius r^* such that $\mathcal{P}$ can be covered by p balls of radius r^*). The decision version of the p-center problem (i.e., determine whether $\mathcal{P}$ can be covered by p balls of a given radius r) is transformable

*assa@math.tau.ac.il
†matya@cs.bgu.ac.il

to a p-piercing problem, where the objects are the n balls of radius r centered at the points of $\mathcal{P}$. This connection is one of the reasons for the recent interest in the p-piercing problem [1,7,9,10].

Sharir and Welzl [10] studied several p-piercing problems for rectangles and c-oriented polygons. (More precisely, 3-,4-, and 5- piercing of rectangles, and 2-piercing of c-oriented polygons). Katz and Nielsen [7] considered several 2-piercing problems, where their main result applies to d-dimensional c-oriented polytopes. Very recently Segal [9] has improved the result of Sharir and Welzl for the case of 4-piercing of rectangles. In this paper we adopt the approach of Sharir and Welzl and obtain several new results some of which pertain to objects in higher dimensions, as mentioned in the abstract. In Section 2 we consider the case of 3-piercing of axis-parallel boxes in $\mathcal{R}^d$, and in Section 3 we deal with 3-piercing of homothetic triangles or 4-oriented quadrilaterals in the plane.

2. 3-Piercing of d-Dimensional Boxes

Let $\mathcal{S}$ be a set of n axis-parallel boxes in $\mathcal{R}^d$. We wish to determine whether $\mathcal{S}$ is 3-pierceable, and, if so, to produce three piercing points. We present efficient algorithms for $d = 3, 4, 5$. For $d = 4, 5$, an $O(n \log n)$ algorithm is presented, and for $d = 3$, an $O(n)$ algorithm is presented, assuming the boxes are already sorted in each of the six orientations. The advantage of the latter algorithm, is that it allows us to save a logarithmic factor in the solution of the corresponding 3-center problem in $\mathcal{R}^3$.

2.1. Preliminaries

A perpendicular hyperplane traverses $\mathcal{S}$. Assume there exists a hyperplane h that is perpendicular to one of the axes, say, the x_1-axis, and that intersects all the boxes in $\mathcal{S}$, and let P be a piercing set for $\mathcal{S}$. Consider a point $a \in P$. By projecting a on h, we do not leave any of the boxes (although we may enter some new boxes), so we may replace the point a with its projection on h. Therefore, one can solve the p-piercing problem for $\mathcal{S}$ by solving the p-piercing problem for the set of $(d-1)$-dimensional boxes consisting of the projections of the boxes in $\mathcal{S}$ on h. This latter problem can be handled in the same manner (recursively) until we reach the following case.

No perpendicular hyperplane traverses $\mathcal{S}$. A box in $\mathcal{R}^d$ has $2d$ facets. If q is a point inside a box, then any axis-parallel line through q intersect exactly two facets of the box. Consider the x_1-axis. If we move on the x_1-axis in the positive direction from $-\infty$, we meet the projections of the facets that are perpendicular to the x_1-axis (as points on the axis). Define f_1^+ to be the rightmost left facet among these facets, and f_1^- to be the leftmost right facet among these facets. In a similar way define $f_2^+, \ldots, f_d^+$ and $f_2^-, \ldots, f_d^-$. For a facet f_i^*, let H_i^* denote the closed half-space that is defined by the hyperplane containing f_i^* and that contains the corresponding box. The (closure of the) intersection of the complements of these

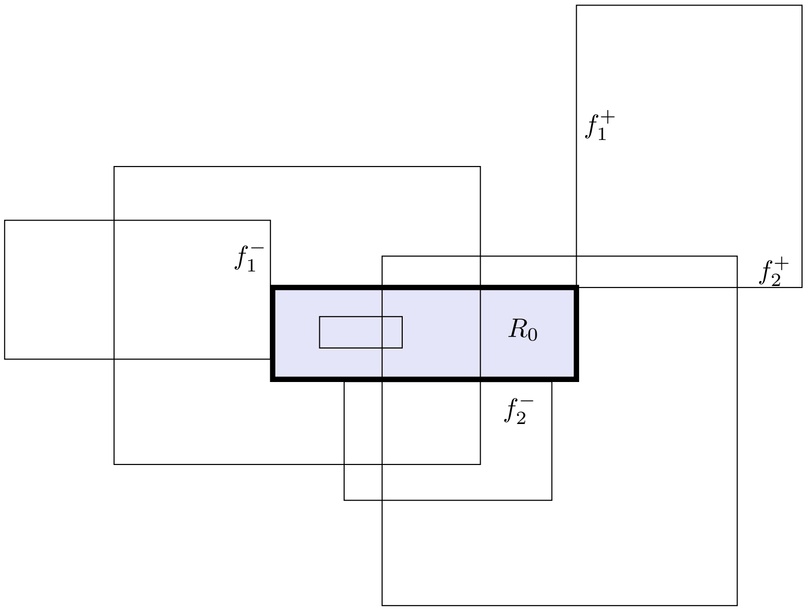

Figure 1: The location domain R_0 of a set of rectangles

$2d$ half-spaces is a d-dimensional box, denoted R_0 and called the *location domain* (see Figure 1). Notice that each f_i^* has a non-empty intersection with the facet on the boundary of R_0 that is defined by f_i^*.

Observation 1: If $\mathcal{A}$ is a piercing set for $\mathcal{S}$, then each of the half-spaces H_i^* contains a point of $\mathcal{A}$.

Observation 2: Assume that $\mathcal{S}$ is p-pierceable (and that there is no perpendicular hyperplane that traverses $\mathcal{S}$). Then there exists a piercing set of p points such that each facet on the boundary of the location domain R_0 contains a piercing point.

It is easy to prove these observations: Each half-space H_i^* fully contains a box, therefore, in order to pierce all the boxes, one must place a piercing point in each half-space H_i^*. Since we are dealing with boxes, we may replace these piercing points with their projections on the corresponding facets of ∂R_0. (Notice that if a point lies in the intersection of k of the half-spaces, then its projection on the boundary of R_0 lies in a $d-k$ face, which is contained in each of the k half-spaces.)

2.2. 2-piercing of d-dimensional boxes

Given a set of n boxes in $\mathcal{R}^d$ we wish to determine whether this set is 2-pierceable, and, if so, to produce a pair of piercing points. Although the following algorithm was implied already by Danzer and Grünbaum [2] (see also Ref. 7) and follows easily from the results in Ref. 10, we describe it here since it plays an important role in the subsequent 3-piercing algorithms. Let us assume that the location domain R_0 exists.

(It is easy to determine in $O(dn)$ time whether R_0 exists by selecting the "leftmost" or "rightmost" facet for each of the $2d$ orientations. If it does not exist, then we obtain in $O(n)$ time a $(d-1)$-dimensional 2-piercing problem, see discussion above.) The location domain R_0 can easily be constructed in time $O(dn)$. According to the above observations, if the set of boxes is 2-pierceable, then we can find a piercing pair such that each of the facets on the boundary of R_0 contains a piercing point. But this is only possible if the 2 piercing points are the two endpoints of one of the diagonals of R_0. Thus, we check for each of the 2^{d-1} diagonals of R_0 whether its two endpoints pierce the entire set or not. The total complexity of the algorithm is $O(2^d n)$ time.

2.3. 3-piercing of 3-dimensional boxes

Let $\mathcal{S}$ be a set of n boxes in $\mathcal{R}^3$. Again, we may assume that R_0 exists. We can easily check whether there exists a piercing triplet such that one of the piercing points is a vertex of R_0: There are 8 vertices, and for each vertex we check whether the remaining unpierced boxes can be pierced by 2 points using the algorithm of the preceding subsection. The complexity of this stage is $O(n)$. Next assume that a piercing triplet where one of the points is a vertex of R_0 does not exist. The only other possibility is that the 3 piercing points lie in the interior of 3 different edges (i.e., 1-dimensional faces) of ∂R_0, so that together they pierce all facets of R_0. The number of legal threesomes of edges is 8 (4 possibilities for an edge parallel to the x-axis, 2 for an edge parallel to the y-axis, and the edge parallel to the z-axis is determined). Let us now assume that we have chosen one of these threesomes s^1, s^2, s^3. We maintain a set $\mathcal{S}'$ (initially $\mathcal{S}' = \mathcal{S}$) and three segments t^1, t^2, t^3 (initially $t^i = s^i$), so that, at any point, $\mathcal{S}$ is pierceable by 3 points, one in each s^i, if and only if $\mathcal{S}'$ is pierceable by 3 points, one in each t^i.

Let b be a box in $\mathcal{S}'$. We can perform the following operation, if applicable, without affecting this invariant.

Operation: If b fully contains one of the segments t^i, then remove b from $\mathcal{S}'$. Otherwise, if b intersects only one of the segments t^i, then replace t^i with $t^i \cap b$ and remove b from $\mathcal{S}'$.

Assume we apply the above operation to the boxes in $\mathcal{S}'$ in arbitrary order until there is no box in $\mathcal{S}'$ to which we can apply the operation. Then we end up in one of the following three situations:

1. $\mathcal{S}'$ is empty. In this case, it is obvious that the set $\mathcal{S}$ is 3-pierceable, and a piercing set for $\mathcal{S}$ is obtained by taking one (arbitrary) point from each segment t^i.

2. There is a box in $\mathcal{S}'$ that is disjoint from all the segments t^i. In this case, it is obvious that the set $\mathcal{S}$ is not 3-pierceable.

3. Neither 1 nor 2 is true. That is, $\mathcal{S}'$ is still not empty and there is no box in $\mathcal{S}'$ that is disjoint from all the segments t^i. This implies that every box

in $\mathcal{S}'$ intersects at least two of the three segments t^i. Notice however that a box cannot intersect all three segments because such a box must fully contain one of the segments, and therefore the operation above can be applied to it. Therefore we may assume that all the boxes remaining in $\mathcal{S}'$ intersect exactly two of the three segments t^i. In this case we may also conclude that $\mathcal{S}$ is 3-pierceable as shown in the next paragraph.

Assume without loss of generality that t^1 is the segment parallel to the z-axis, and choose its top endpoint to be the first piercing point. As the second piercing point we take the furthest (from t^1) of the two endpoints of t^2, where t^2 is the segment from the top face of R_0. (That is, if we move along the frame of R_0 from the top endpoint of t^1 towards t^2, taking the shortest path, then the second endpoint of t^2 that is encountered is chosen as the second piercing point.) In the same manner we pick the third piercing point as the furthest of the two endpoints of t^3 from t^2. Actually the process we just described yields two disjoint sets of points (the 3 endpoints that were chosen and the 3 remaining endpoints), and both qualify as a piercing set due to the following reason: Consider a box that intersects two segments t^i and t^j, it must contain one endpoint of each of these segments, and according to the process described one of the endpoints belongs to the first set and the other to the second set.

Notice that the number of steps in the above process is $O(n^2)$, because, repeatedly, we might need to consider $O(n)$ boxes before encountering the next box to which we may apply the operation. In order to perform this process efficiently, we first sort the intersections of the facets of the boxes in $\mathcal{S}$ with each of the edges of R_0. Now, whenever a segment t^i is shortened, we have to detect the boxes that lose contact with it and the boxes that come to fully contain it. This is easily done by noticing that each of those boxes must have a facet that intersects the removed part of t^i, so we just traverse the removed part and process each of the boxes that have a facet intersecting that part. Since a box is processed in this way only a constant number of times, the actual execution of the sequence of operations takes linear time (after the initial sorting). We thus have:

Theorem 1 *(i) The 3-piercing problem for n boxes in $\mathcal{R}^3$ can be solved in time $O(n \log n)$. (ii) If the facets of the boxes are already sorted along the axes, then the problem can be solved in time $O(n)$.*

2.4. 3-piercing of d-dimensional boxes, $d = 4, 5$

Given a set $\mathcal{S}$ of n boxes in $\mathcal{R}^d$, $d = 4, 5$, we wish to determine whether $\mathcal{S}$ is 3-pierceable, and, if so, to compute a piercing triplet. Again, we may assume that the location domain R_0 exists, and that $\mathcal{S}$ is not 3-pierceable if we require that one of the piercing points is a vertex of R_0. We cannot use here the method of the previous subsection, since it is not necessarily true anymore that there exist 3 points lying on edges (i.e., 1-dimensional faces) of the boundary of R_0 that form a piercing set for $\mathcal{S}$, whenever $\mathcal{S}$ is 3-pierceable. However, if $\mathcal{S}$ is 3-pierceable, we can still find a piercing set such that at least one of its points lies on an edge of R_0. This is true since in order to pierce the 8 (if $d = 4$) or 10 (if $d = 5$) boxes defining the facets of R_0, at least one of the piercing points must lie in the intersection of 3 or 4 of these boxes, respectively. This point can be moved to the appropriate edge of R_0 without exiting any of the boxes in $\mathcal{S}$ (since the boxes are axis-parallel). Notice that if $d \geq 6$ this is false; it is possible that every piercing triplet does not have a point on an edge of R_0.

Let t be an edge of R_0. We search for a piercing triplet with a point on t. In the worst case, this search is performed $d \cdot 2^{d-1}$ times, once for each edge of R_0. The boxes in $\mathcal{S}$ induce a partition of t into at most $2n - 1$ 'atomic' segments. Let $\mathcal{S}_j$ denote the subset of boxes in $\mathcal{S}$ that do not contain the j-th atomic segment t_j. Notice that the subsets $\mathcal{S}_j$ and $\mathcal{S}_{j+1}$ differ by at most one box (assuming general position for clarity of presentation only); we either enter a new box when moving from t_j to t_{j+1} in which case this box is added to $\mathcal{S}_j$ to obtain $\mathcal{S}_{j+1}$, or we exit a box in which case this box is deleted from $\mathcal{S}_j$ to obtain S_{j+1}. We traverse the edge t, and check for each atomic segment t_j whether the subset $\mathcal{S}_j$ is 2-pierceable.

We next describe a dynamic data structure of size $O(n \log n)$ that enables us to determine in $O(\log n)$ time whether the current subset $\mathcal{S}_j$ is 2-pierceable. Such a data structure should support the following two basic operations:

1. For a pre-stored subset $\mathcal{S}'$ of $\mathcal{S}$, find the facets $f_1^+, f_1^-, ..., f_d^+, f_d^-$ defining its location domain.
2. For a pre-stored subset $\mathcal{S}'$ of $\mathcal{S}$ and for a query point p, find the subset $\mathcal{S}'_p \subseteq \mathcal{S}'$ of boxes that are not pierced by p.

These operations are decomposable: If $\mathcal{S}'$ is given as the disjoint union of a collection of subsets of $\mathcal{S}$, then the output of an operation on $\mathcal{S}'$ can be obtained from the outputs of the same operation on the subsets in the collection, in time that is proportional to the number of these subsets. For the first operation, we select from the outputs for the subsets the most extreme facet in each orientation, and for the second operation we take the union of the outputs for the subsets.

In order to perform these operations efficiently, we construct for each of the $2d$ orientations a binary search tree. In the tree for the positive (resp. negative) x_i-orientation, the boxes are stored in the leaves of the tree, sorted by the projections of their appropriate facets on the x_i-axis in increasing (resp. decreasing) order. An internal node v represents the canonical subset $\mathcal{S}'_v$ which consists of the boxes stored at the leaves of the subtree rooted at v, and stores the region (box) I_v which is the intersection of the boxes in $\mathcal{S}'_v$.

Performing operations of the first type is easy; from each tree we take the appropriate facet of the box stored in the rightmost leaf. These facets define the location domain R_0. For operations of the second type, let $p = (p_1, \ldots, p_d)$ be a query point. Notice that if p does not lie in a box, then p does not lie in at least one of the $2d$ half planes defined by the facets of the box. Thus, we search in each of the trees with the appropriate coordinate of p. For example, in the tree for the positive x_1-orientation we search with the coordinate p_1, and obtain, in a standard way, the boxes that lie to the right of p as a collection of $O(\log n)$ canonical subsets. The union of the $O(2d \log n)$ subsets is the desired output.

We now describe the complete 3-piercing algorithm for an edge t of R_0.

1. Divide t into at most $2n-1$ atomic intervals as described above. Place a piercing point in the first interval t_1, and compute the corresponding set $\mathcal{S}_1$ of boxes that are not pierced by this point. If $\mathcal{S}_1$ is 2-pierceable then we are done, otherwise construct the $2d$ binary search trees for the set $\mathcal{S}_1$.

2. Iterate over the atomic intervals of t moving from the current interval to its next neighbor. At interval t_j, obtain the set $\mathcal{S}_j$ from the set $\mathcal{S}_{j-1}$ by adding or deleting a single box, and update the search trees accordingly. The cost of an update is $O(\log n)$.

3. Let R_0^j denote the location domain of $\mathcal{S}_j$. R_0^j can be computed in constant time from the trees corresponding to $\mathcal{S}_j$ as described above.

4. We determine whether $\mathcal{S}_j$ is 2-pierceable as follows. We pick a vertex p of R_0^j and check whether the opposite vertex q of R_0^j pierces all the boxes of $\mathcal{S}_j$ that are not pierced by p, i.e., whether q lies in the intersection of $\mathcal{S}_{j,p}$. We repeat this if necessary for each of the diagonal pairs. Above, we showed how to compute the set $\mathcal{S}_{j,p}$ in $O(\log n)$ time. $\mathcal{S}_{j,p}$ was obtained as a collection of $O(\log n)$ canonical sets. To check whether q lies in the intersection of $\mathcal{S}_{j,p}$, we only need to check for each of the boxes I_v associated with the canonical sets defining $\mathcal{S}_{j,p}$ whether it is pierced by q.

The time complexity of the above algorithm is clearly $O(n \log n)$. We thus obtain the following theorem.

Theorem 2 *The 3-piercing problem for n boxes in $\mathcal{R}^d$, $d = 4, 5$, can be solved in time $O(n \log n)$.*

Using the same approach for higher dimensions, it is easy to see that we may assume that one of the piercing points lies on a face of R_0 of dimension $\lfloor \frac{d}{3} \rfloor$. Instead of n atomic intervals, we have a partition of this face into $O(n^{\lfloor \frac{d}{3} \rfloor})$ atomic regions, which leads to an algorithm of complexity $O(n^{\lfloor \frac{d}{3} \rfloor} \log n)$.

Corollary 1 *The 3-piercing problem for boxes in $\mathcal{R}^d$, $d \geq 6$, can be solved in time $O(n^{\lfloor \frac{d}{3} \rfloor} \log n)$.*

We can also solve the corresponding (rectilinear) 3-center problem efficiently: Given a set $\mathcal{P}$ of n points in $\mathcal{R}^d$, compute the smallest edge length λ such that $\mathcal{P}$ can be covered by the union of three axis-parallel cubes of edge length λ. Sharir and Welzl [10] applied the matrix searching technique of Frederickson and Johnson [3,4,5] to solve efficiently the planar rectilinear p-center problem. They proved that given an algorithm for the rectilinear p-piercing problem, one can easily obtain, using matrix searching, an algorithm for the rectilinear p-center problem that issues only $O(\log n)$ calls to the piercing algorithm (see also Glozman et al. Ref. 6). Their proof is only for the planar case, but it is easy to extend it to the d-dimensional case: We simply construct d (rather than 2) difference matrices, one for each dimension, and conduct the search for the smallest side-length (of a cube) in each of these matrices. Since d is a constant we still issue only $O(\log n)$ calls to the piercing procedure. Thus, recalling Theorem 1 and Theorem 2 we obtain the following theorem:

Theorem 3 *The rectilinear* 3*-center problem can be solved in* $O(n \log n)$ *time in* $\mathcal{R}^3$*, and in* $O(n \log^2 n)$ *time in* $\mathcal{R}^4$ *or* $\mathcal{R}^5$*.*

Remark: Using similar arguments, we can show that if there exists a piercing set of size 4 or 5 for a set of n boxes in $\mathcal{R}^3$, then there exists such a set such that one of its points lies on a 1-dimensional edge of the location domain. However, this observation does not lead to near-linear solutions to the corresponding piercing problems, since, for example, in the 4-piercing problem it is possible that all piercing triplets for S_j consist of three points lying in the interiors of 1-dimensional edges of R_0^j, which implies that we cannot determine whether S_j is 3-pierceable in sub-linear time.

3. 3-Piercing of Homothetic Triangles and 4-Oriented Quadrilaterals

The problem considered in this section is the following: Given a set $\mathcal{S}$ of n c-oriented (convex) polygons in the plane, determine whether it is 3-pierceable, and, if it is, produce a piercing triplet for $\mathcal{S}$. For each orientation θ, let l_θ be the oriented line of orientation θ that satisfies the following conditions:

1. l_θ contains an edge (of orientation θ) of some polygon in $\mathcal{S}$ (such that the polygon lies to its left).

2. None of the polygon edges of orientation θ is contained in the open left half-plane defined by l_θ.

Informally l_θ is the most extreme line among the lines of orientation θ supporting a polygon in $\mathcal{S}$. Put $\mathcal{L} = \{l_{\theta_1}, l_{\theta_2}, \ldots, l_{\theta_c}\}$, and denote the arrangement of $\mathcal{L}$ by $\mathcal{A}(\mathcal{L})$. For $c = 3, 4$ we have the following lemma (see Figure 2).

Lemma 1 *Let $\mathcal{S}$ be a set of 4-oriented (convex) quadrilaterals. (We allow degenerate quadrilaterals, i.e., triangles.) If $\mathcal{S}$ is 3-pierceable, then there exists a piercing triplet such that one of the points lies on a line of $\mathcal{L}$.*

Proof. Let $\{p_1, p_2, p_3\}$ be a piercing set for $\mathcal{S}$. Let $\mathcal{S}_i \subseteq \mathcal{S}$ be the subset of quadrilaterals in $\mathcal{S}$ that are pierced by p_i, and let $\Delta_i = \cap_{s \in \mathcal{S}_i} s$, $i = 1, 2, 3$. Assume that $\Delta_i \cap \mathcal{L} = \emptyset$, $i = 1, 2, 3$. (Otherwise, if $\Delta_i \cap \mathcal{L} \neq \emptyset$, then we can replace p_i with a point on $\Delta_i \cap \mathcal{L}$.)

Assuming general position, Δ_1 is either a triangle or a quadrilateral. (Otherwise, Δ_1 might be a vertex or a line segment, but even so it is still determined by at least three edges of three different orientations, so the proof below holds for degenerate scenes as well.) In both cases, since $\Delta_1 \cap \mathcal{L} = \emptyset$, at least three of the four (closed) half planes defined by the lines in $\mathcal{L}$ are disjoint from Δ_1. Since the points p_2, p_3 pierce these three half planes and the objects inducing them, one of them, say p_2, must lie in the interior of the intersection of two of the three half planes and pierce the two corresponding quadrilaterals. By our assumption the region Δ_2 is fully

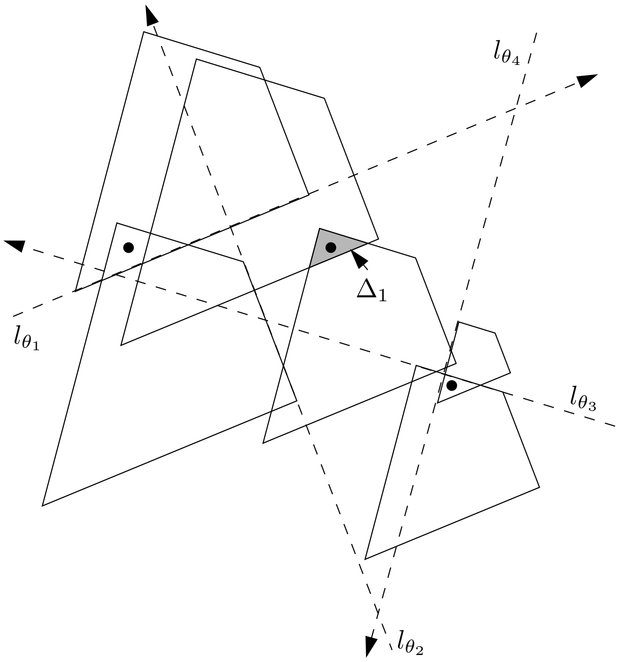

Figure 2: A set of 4-oriented quadrilaterals, the corresponding set $\mathcal{L}$, and the region Δ_1

contained in the interior of this intersection. But this is impossible since there are only two possible orientations for the edges of the boundary of Δ_2, while Δ_2 is either a triangle or a quadrilateral. $\square$

We now assume that $\mathcal{S}$ is either a set of homothetic triangles or a set of 4-oriented trapezoids. The lemma above allows us to continue in a way similar to the way that was chosen for the case of 3-piercing of boxes. We pick a line of $\mathcal{L}$, divide it into at most $2n-1$ atomic intervals, and try to place a piercing point in one of these intervals, by traversing the intervals one after the other and checking for the current interval whether the remaining set of unpierced objects is 2-pierceable. If necessary we repeat this procedure for the rest of the lines in $\mathcal{L}$.

Thus we need to solve the dynamic 2-piercing problem for homothetic triangles (alternatively, 4-oriented trapezoids). Sharir and Welzl [10] showed that if $\mathcal{S}$ can be pierced by two points, then it can be pierced by two points lying on the lines of $\mathcal{L}$. Moreover, we may assume that one of these points is a vertex of $\mathcal{A}(\mathcal{L})$. Based on this observation we show below how to perform an update in time $O(\log n)$, which leads to an $O(n \log n)$ algorithm for the case of 3-piercing of homothetic triangles (alternatively, 4-oriented trapezoids).

Lemma 2 *The* 2*-pierceability property of a set of* n *homothetic triangles (alternatively, 4-oriented trapezoids) can be maintained dynamically in* $O(\log n)$*-time per update.*

Proof. We prove the lemma for the case of homothetic triangles. The proof for the case of 4-oriented trapezoids is essentially identical. Let $\mathcal{S}$ be a set of n homothetic triangles in the plane. We need to show how the 2-pierceability property is maintained (i.e., whether $\mathcal{S}$ is 2-pierceable or not) when a homothetic triangle is either inserted to or deleted from $\mathcal{S}$.

The dynamic data structure we are about to describe is similar to the one used for the case of 3-piercing of boxes (where $d = 4, 5$). It consists of three balanced binary trees, one for each orientation. Assume without loss of generality that the first orientation is the positive vertical orientation (so the triangles lie to the left of their vertical edge). Then the first tree is built as follows. Let $x_1 < x_2 < \cdots < x_n$ be the projections of the vertical edges on the x-axis. Store these values in the leaves of a balanced binary tree; x_1 in the leftmost leaf, and so on. A node u of the tree represents a subset $\mathcal{S}_u$ of $\mathcal{S}$ (i.e. $\mathcal{S}_u$ consists of the triangles in $\mathcal{S}$ for which the projection of their vertical edge is stored in a leaf of the subtree rooted at u). We associate with u the region $T_u = \cap \mathcal{S}_u$ (which is either empty or a homothetic triangle). The tree can be constructed in a bottom-up manner in $O(n)$ time, after sorting the leaf values in $O(n \log n)$ time. The second and third trees are constructed in a similar way.

We need to determine in logarithmic time whether $\mathcal{S}$ is 2 pierceable. Notice first that we can obtain the three extreme lines (i.e., the lines of $\mathcal{L}$) in constant time, since they are the lines corresponding to the minimal values stored in the three trees. Now let v be a vertex of the arrangement $\mathcal{A}(\mathcal{L})$. We need to determine whether the subset $\mathcal{S}_v$ of $\mathcal{S}$ consisting of the triangles in $\mathcal{S}$ that are not pierced by v has a non-empty intersection. We can obtain $\mathcal{S}_v$ (implicitly) as the union of a logarithmic number of represented subsets (as explained above for 3-piercing of boxes), and since we have the intersection of each of these subsets, we can compute $\cap \mathcal{S}_v$ in logarithmic time. Using standard techniques we can update this data-structure when a triangle is either inserted to or deleted from $\mathcal{S}$ in logarithmic time. Thus the 2-pierceability property can be maintained in logarithmic time. $\Box$

We thus obtain the following theorem:

Theorem 4 *Let $\mathcal{S}$ be a set of n homothetic triangles, or of n 4-oriented trapezoids. Then it is possible to find a piercing triplet for $\mathcal{S}$ (if such a triplet exists) in time $O(n \log n)$.*

Remark: Although Lemma 1 is proven for 4-oriented quadrilaterals, we had to restrict ourselves to homothetic triangles or 4-oriented trapezoids, since in general we do not know how to maintain the 2-pierceability property efficiently (i.e., it is not true in general that if $\mathcal{S}$ is 2-pierceable, it can be pierced by two points lying on the lines of $\mathcal{L}$, one of which is a vertex of $\mathcal{A}(\mathcal{L})$).

Acknowledgment

We would like to thank Franck Nielsen and Micha Sharir for helpful discussions on the contents of this paper.

1. E. Assa, On Rectilinear and Polygonal p-Center and p-Piercing Problems, M.Sc. thesis, Tel-Aviv University, 1997.

2. L. Danzer and B. Grünbaum, Intersection properties of boxes in $\mathcal{R}^d$, *Combinatorica*, 2(3) (1982), 237–246.

3. G.N. Frederickson and D.B. Johnson, The complexity of selection and ranking in $X+Y$ and matrices with sorted rows and columns, *J. Comput. Syst. Sci.*, 24 (1982), 197–208.

4. G.N. Frederickson and D.B. Johnson, Finding kth paths and p-centers by generating and searching good data structures, *J. Algorithms*, 4 (1983), 61–80.

5. G.N. Frederickson and D.B. Johnson, Generalized selection and ranking: sorted matrices, *SIAM J. Comput.*, 13 (1984), 14–30.

6. A. Glozman, K. Kedem and G. Shpitalnik, On some geometric selection and optimization problems via sorted matrices, *Proc. 4th Workshop on Algorithms and Data Structures*, 1995, 26–37.

7. M.J. Katz and F. Nielsen, On piercing sets of objects, *Proc. 12th ACM Symp. on Computational Geometry*, 1996, 113–121.

8. N. Meggido and K.J. Supowit, On the complexity of some common geometric location problems, *SIAM J. Comput.*, 13 (1984), 182–196.

9. M. Segal, On the piercing of axis-parallel rectangles and rings, *Proc. 5th European Symp. on Algorithms*, 1997, 430–442.

10. M. Sharir and E. Welzl, Rectilinear and polygonal p-piercing and p-center problems, *Proc. 12th ACM Symp. on Computational Geometry*, 1996, 122–132.