Rectilinear and Polygonal p-Piercing and p-Center Problems*

Micha Sharir† Emo Welzl‡

November 13, 1996

Abstract

In the *p-piercing problem*, we are given a collection of regions, and wish to determine whether there exists a set of p points that intersects each of the given regions. We give linear or near-linear algorithms for small values of p in cases where the given regions are either axis-parallel rectangles or convex c-oriented polygons in the plane (i.e., convex polygons with sides from a fixed finite set of directions).

We also investigate the planar *rectilinear (and polygonal) p-center problem*, in which we are given a set S of n points in the plane, and wish to find p axis-parallel congruent squares (isothetic copies of some given convex polygon, respectively) of smallest possible size whose union covers S. We also study several generalizations of these problems.

New results are a linear-time solution for the rectilinear 3-center problem (by showing that this problem can be formulated as an LP-type problem and by exhibiting a relation to Helly numbers). We give $O(n \log n)$-time solutions for 4-piercing of translates of a square, as well as for the rectilinear 4-center problem; this is worst-case optimal. We give $O(n \operatorname{polylog} n)$-time solutions for 4- and 5-piercing of axis-parallel rectangles, for more general rectilinear 4-center problems, and for rectilinear 5-center problems. 2-pierceability of a set of n convex c-oriented polygons can be decided in time $O(c^2 n \log n)$, and the 2-center problem for a convex c-gon can be solved in $O(c^5 n \log n)$ time. The first solution is worst-case optimal when c is fixed.

1 Introduction

The problems. Let $\mathcal{R}$ be a set of n regions in the plane, and let p be a positive integer. $\mathcal{R}$ is called *p-pierceable* if there exists a set of p *piercing points* which intersects every member of $\mathcal{R}$. The *p-piercing problem* is to determine whether $\mathcal{R}$ is p-pierceable, and, if so, to produce a set of p piercing points.

*Both authors acknowledge support by G.I.F. — the German Israeli Foundation for Scientific Research and Development, and by a Max Planck Research Award. Work by Micha Sharir has also been supported by National Science Foundation Grants CCR-94-24398 and CCR-93-11127, and by grants from the U.S.–Israeli Binational Science Foundation, and the Israel Science Fund administered by the Israeli Academy of Sciences. Emo Welzl has also been supported by a Leibniz Prize from the German Research Society, We 1265/5-1. Part of the work on this paper was done while Emo Welzl was at Freie Universität Berlin, and during his participation in the Special Semester on Combinatorial and Computational Geometry, organized by the Mathematical Research Institute of Tel Aviv University.

†School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel, and Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA.

‡Institut für Theoretische Informatik, ETH Zürich, CH-8092 Zürich, Switzerland.

In the *p-center problem* we are given a set S of n points in the plane, some compact convex set C, and a positive integer p. The goal is to find p isothetic copies of C of smallest possible scaling factor, whose union covers S.

This problem arises in the area of facility location in operations research, and many variants of it have been studied [9, 10, 12, 14, 25, 29, 31, 35, 37, 38, 40]. If C is the unit ball of some norm $\|\cdot\|_C$, then the p-center problem seeks a set P of p 'facility points', so that the maximum $\|\cdot\|_C$-distance from a point of S to its nearest facility point is minimized. A natural case is when C is a unit disk, so $\|\cdot\|_C$ is the euclidean distance (*euclidean p-center problem*). When C is a square, then $\|\cdot\|_C$ can be viewed as the ℓ_1- or ℓ_∞-norm (depending on the orientation of the square), and we face the *rectilinear p-center problem.*

A standard reduction from the p-center problem to the p-piercing problem goes as follows. Let $\tilde{C}$ denote the reflection of C (with respect to some interior point, which we assume to be the origin). Define $\mathcal{R}(\lambda) = \{s + \lambda\tilde{C} \mid s \in S\}$, $\lambda \geq 0$. Then we seek the smallest possible λ for which $\mathcal{R}(\lambda)$ is p-pierceable. The piercing points (for the smallest λ) are the locations of the desired facilities; they serve as centers of isothetic copies of λC whose union covers S. Often, the p-center problem is solved by techniques like *parametric searching* [36] or *monotone matrix searching* [16, 17, 18], which run some sort of binary search on λ to locate the optimal solution. The related p-piercing problem for $\mathcal{R}(\lambda)$, for any fixed λ, is then referred to as the *decision problem*, or the *fixed-size problem*, and is used as an 'oracle' to guide the binary search.

Previous results. It is known that the p-piercing and p-center problems are NP-complete when p is part of the input, even if the regions are translates of a square [38], and that they can be solved in polynomial time for any fixed p (assuming that C and the sets to be pierced are simply shaped, see [9, 30]).

The 1-piercing problem is easy for simple regions, and the 1-center problem is well understood at this point and allows optimal linear-time solutions for many types of regions — we will not elaborate on this here (see [34, 41]). The euclidean 2-center problem was recently solved in near-linear time [40] (see also [13]), but no such solution is known for $p \geq 3$. The 2- and 3-piercing problems for a set of axis-parallel rectangles have been solved in linear time [28, 29].[1]

The rectilinear 2-center problem was also known to be optimally solvable in linear time [10], while the best previous solution for the rectilinear 3-center problem takes $O(n \log n)$ time [10, 29]. Actually, the papers [28, 29] consider a slightly more general version, the so-called *weighted rectilinear p-center problem*: Here we are given n axis-parallel squares, not necessarily of the same size. We ask for the smallest common scaling factor, so that the scaled squares (where the scaling is about the center of each square) is still p-pierceable. It is shown in [28] that the weighted rectilinear 2-center problem is solvable in linear time.

2-pierceability of c-oriented convex polygons is closely related to a conjecture of Danzer and Grünbaum [8], which asks for a Helly-type theorem for 2-pierceability of collections of homothets of a convex polygon, and which was recently refuted in [26]. However, it was shown in [26] that such a result exists for the case of homothetic triangles.

For $p > 3$, the best previous bound for the rectilinear p-center problem is $O(n^{p-2} \log n)$

[1] In fact, the results in [29] are stated only for homothets of squares.

[30], although it appears [5] that the technique of [24] for solving the euclidean p-center problem can be extended and applied to the rectilinear case as well, to obtain an $n^{O(\sqrt{p})}$ solution.

No near-linear solutions have previously been known for the p-piercing or p-center problem in the general polygonal case, for $p \geq 2$.

New results. We improve the $O(n \log n)$ bound for the (weighted) rectilinear 3-center problem to optimal linear time. For the (unweighted) rectilinear 4-center problem we give an $O(n \log n)$ solution, and for the rectilinear 5-center problem we describe an $O(n \log^5 n)$ solution, thereby improving on the previous bounds of the form $O(n^2 \log n)$ and $O(n^3 \log n)$, respectively. The 4-center algorithm is also shown to be worst-case optimal. The new bounds for the center problems are based on new results for the corresponding piercing problems: 4-piercing of translates of a square can be decided in $O(n \log n)$ time, 4- (and 5-) piercing of general axis-parallel rectangles can be decided in time $O(n \log^3 n)$ (and $O(n \log^4 n)$, respectively). These results also lead to $O(n \operatorname{polylog} n)$ algorithms for the weighted rectilinear 4- and 5-center problems.

Our solutions also imply that the rectilinear p-center problem, for any $p \geq 5$, can be solved in time $O(n^{p-4} \log^5 n)$. However, as noted above, we believe that much better results can be obtained for larger p's.

The 2-center problem, when C is a convex c-gon (*polygonal 2-center problem*), can be solved in $O(c^5 n \log n)$ time. This is based on 2-piercing of *c-oriented* convex polygons (whose sides draw their directions from a fixed set of c orientations), for which we supply an $O(c^2 n \log n)$ solution, and show that it is worst-case optimal when c is fixed. 2-piercing of a set of homothetic triangles, and the triangular 2-center problem, can both be solved in linear time. The same holds for 4-oriented trapezoids.

Independently, Katz and Nielsen [27] have recently obtained several results on the piercing problems studied in this paper, and also results concerning 2-piercing of boxes, simplices, and c-oriented polytopes in higher dimensions. They have not addressed the related center problems. Some other related results have recently been obtained by Assa [5].

Methods. In order to obtain the linear bound for the 3-center problem, we formulate it as an *LP-type problem* [33], from which a randomized linear-time algorithm follows, which can also be derandomized to a deterministic linear-time solution [6]. The proof of the LP-type properties (and of finite combinatorial dimension—see below) relies on a *Helly-type* result of the following kind [8]: A set of axis-parallel rectangles is 3-pierceable if and only if any 16 (or fewer) of them are 3-pierceable. The connection between Helly-numbers and LP-type problems has been investigated in [3, 4]. The realization of the rectilinear 3-center problem as an LP-type problem seems to be particularly interesting, since there is no formulation of this problem as optimization in a convex domain, which shows that LP-type problems go beyond convex programming (in natural examples).

The basis for the rectilinear piercing results are a number of simple observations on the location of piercing points, which have already been used (in one disguise or another) in previous attacks on the rectilinear problem [10, 28, 29]. We enhance those insights by a carefully planned iterative process (for 4-piercing of translates of a square), and by multilevel orthogonal range-searching data structures (for general rectilinear 4- and 5-piercing). These observations can also be extended (but become slightly more involved) to the case of c-oriented polygons, which leads to an $O(n \log n)$ solution of the 2-piercing problem for such polygons. (The case of triangles or of trapezoids can be formulated as an LP-type problem, where for the case of trapezoids we provide a new Helly-type result, extending the result of [26] for the case of triangles. There are also simple deterministic linear-time solutions for both problems.) For both the standard rectilinear and polygonal cases, the reduction from the center to the piercing problem uses the matrix-searching technique of Frederickson and Johnson [16, 17, 18], in a manner similar to recent applications in [22]. The weighted rectilinear and polygonal cases can be handled by employing standard parametric searching techniques [36].

2 Rectilinear p-Piercing

We are given a set $\mathcal{R}$ of n axis-parallel closed rectangles in the plane and some positive integer p, and we wish to decide whether $\mathcal{R}$ is p-pierceable. We start with some basic observations for general p (see also [10, 28, 29]), and then "specialize" to $p = 2, \ldots, 5$ to obtain linear or near-linear solutions.

An axis-parallel line traverses $\mathcal{R}$. Suppose a vertical line ℓ intersects all rectangles in $\mathcal{R}$ (the case where ℓ is horizontal is treated similarly), and let P be some set of piercing points. Consider $a \in P$. If we move a on a horizontal segment towards ℓ, then we do not leave any of the rectangles (we may enter some). That is, if we replace each $a \in P$ by its horizontal projection a' on ℓ, then the resulting set P' is still piercing. Consequently, the p-piercing problem can be solved by deciding p-pierceability of the set $\mathcal{R}' = \{\ell \cap r \mid r \in \mathcal{R}\}$ of intervals on ℓ. This 1-dimensional piercing problem can be solved in $O(n \log p)$ time.[2] So if there is a vertical (or horizontal) line intersecting all rectangles, the problem can be solved in linear time for any fixed p.

No axis-parallel line traverses $\mathcal{R}$. Let ℓ^L be the vertical line containing the leftmost right edge of a rectangle in $\mathcal{R}$, let ℓ^R be the vertical line containing the rightmost left edge of a rectangle in $\mathcal{R}$, let ℓ^T be the horizontal line containing the highest bottom edge of a rectangle in $\mathcal{R}$, and let ℓ^B be the horizontal line containing the lowest top edge of a rectangle in $\mathcal{R}$. Consider the closed left halfplane H^L bounded by ℓ^L, the closed right halfplane H^R bounded by ℓ^R, the closed top halfplane H^T bounded by ℓ^T, and the closed bottom halfplane H^B bounded by ℓ^B. Let $\tilde{H}^X$ denote the closure of the complement of H^X, for $X = L, R, T, B$. Note that if no axis-parallel line traverses $\mathcal{R}$, then H^L is disjoint from H^R, and H^T is disjoint from H^B, and thus $R_0 := \bigcap_{X \in \{L,R,T,B\}} \tilde{H}^X$ is nonempty (with nonempty interior). R_0 is called *location domain for* $\mathcal{R}$, a term justified in the following simple but crucial observation.

Observation 2.1 *(a) Any set of piercing points of $\mathcal{R}$ must have a point in each of the halfplanes H^X, for $X = L, R, T, B$.*

[2] We are not aware of any explicit reference to this bound; this problem appears as an exercise in [23, p. 193].

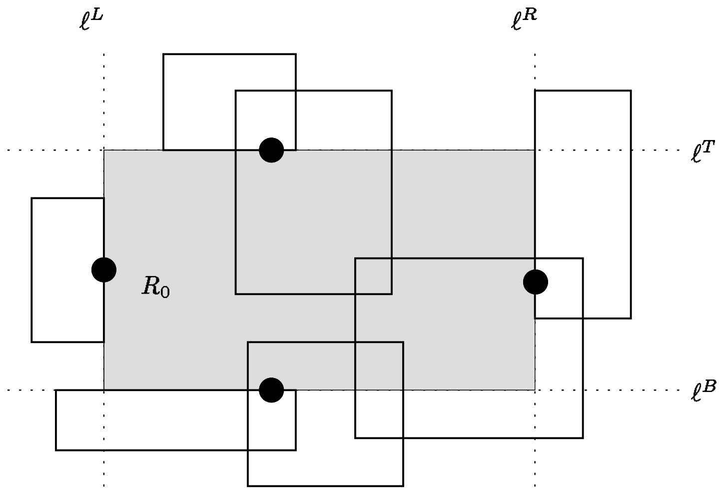

Figure 1: A 4-pierceable set of axis-parallel rectangles and its location domain R_0

(b) Assume that $\mathcal{R}$ is p-pierceable, but has no axis-parallel traversing line. Then there is a set of p piercing points in the location domain R_0. In such a piercing set, each side of the boundary ∂R_0 of R_0 must contain one of the piercing points. (Here we consider the sides as relatively closed, that is, a vertex of ∂R_0 is contained in its two incident sides.)

See Figure 1. (a) is trivially true, since each halfplane H^X fully contains one of the rectangles in $\mathcal{R}$, for $X = L, R, T, B$, and so it must contain a piercing point. For (b), consider a piercing set P and a point $a \in P$ which is not in $\tilde{H}^L$, i.e., it is to the left of ℓ^L. Similar to a previous argument, we consider the horizontal projection a' of a on ℓ^L. The set of rectangles containing a' is a superset of the set of rectangles containing a. Hence we can replace a by a' in P without losing its piercing property, and a' is contained in $\tilde{H}^L$. A similar procedure can be employed if a piercing point is not in $\tilde{H}^R$, $\tilde{H}^T$, or $\tilde{H}^B$. In this way we can move every piercing point that lies outside the location domain R_0 to its boundary (up to two projection steps will do it for each point). After that, the piercing set is contained in R_0, but (a) is still valid, and so each side of R_0 must contain a piercing point.

2.1 2- and 3-piercing

If we want to decide whether a set of axis-parallel rectangles is 2-pierceable, we first determine ℓ^L and ℓ^R. If ℓ^L is not to the left of ℓ^R, then a vertical line (say ℓ^L or ℓ^R) intersects all rectangles and we can invoke the simple 1-dimensional decision procedure. Otherwise, we apply a symmetric procedure for ℓ^T and ℓ^B. What remains is the case that the location domain R_0 is a nonempty rectangle. Observation 2.1(b) states that if $\mathcal{R}$ is 2-pierceable, then it is pierceable by two points in R_0, with one point on each side of R_0, and the only candidates for such a set are the two diagonal pairs of vertices of ∂R_0. We can check these two possibilities in linear time.

For 3-pierceability, we can either apply the 1-dimensional procedure, or R_0 exists, and now, by Observation 2.1(b), a piercing set of 3 points contained in R_0 must contain one of the vertices of ∂R_0. We can take each of those vertices v, and check for 2-pierceability of the rectangles disjoint from v. If one of the four possibilities is successful, then $\mathcal{R}$ is 3-pierceable; otherwise, it is not.

Summing up, we get the following result, which is—at least implicitly— also implied by [10, 28, 29].

Theorem 2.2 *2- and 3-pierceability of a set of n axis-parallel rectangles can be decided in $O(n)$ time.*

Remark. One can make the above algorithms more concise to state and implement by writing down an explicit list of a constant number of *candidate sets* of pairs or triples of piercing points, which depend on the given set $\mathcal{R}$ of rectangles, so that $\mathcal{R}$ is 2- or 3-pierceable if and only if one of these sets is piercing. We will do this below, when we talk about the corresponding 2- and 3-center problems.

2.2 4-piercing of translates of a square

Let $\mathcal{Q}$ be a set of n translates of a square of unit side length, for which we want to decide the existence of 4 piercing points. Again, we may assume that the location domain R_0 exists. We can easily decide whether there is a piercing set of four points, one of which is a vertex of ∂R_0: Take each such vertex in turn, and test for 3-pierceability of the squares disjoint from it, in linear time.

So if all these attempts to find a set of four piercing points fail, and $\mathcal{Q}$ is 4-pierceable, then we know, by Observation 2.1(b), that there must be four piercing points, one in the relative interior of each of the sides of R_0. Denote those sides (relatively open segments) by s^X, for $X = L, R, T, B$.

We manipulate a subset $\mathcal{Q}'$ of $\mathcal{Q}$ (initially $\mathcal{Q}' = \mathcal{Q}$), and four subsegments $t^X \subseteq s^X$, for $X = L, R, T, B$ (initially $t^X = s^X$), while maintaining the following invariant: $\mathcal{Q}$ is pierceable by four points, one in each s^X if and only if $\mathcal{Q}'$ is pierceable by four points, one in each t^X. The invariant is trivially true at the initialization stage.

The operations we perform are the following:

(i) *If a square q in $\mathcal{Q}'$ intersects only one of the segments t^X, then we replace*[3] *t^X by $t^X \cap q$ and remove q from $\mathcal{Q}'$.*

(ii) *If a square q contains one of the segments t^X, then we remove q from $\mathcal{Q}'$.*

Clearly, the invariant is maintained after each operation of type (i) or (ii). Note that we can apply right at the start four operations of type (i), to the four squares defining R_0 (as in Figure 1). We may end up in one of the following three situations:

(A) At some point a square in $\mathcal{Q}'$ is disjoint from all segments t^X. That is, $\mathcal{Q}'$ is not pierceable by points on the segments t^X, and the invariant implies that $\mathcal{Q}$ is not pierceable by points, one on each s^X. With the previous assumptions, we conclude that $\mathcal{Q}$ is not 4-pierceable.

(B) At some point $\mathcal{Q}'$ is empty. We conclude that $\mathcal{Q}$ is 4-pierceable. (Note here that the segments t^X can never vanish in the process.)

[3]Note that, as soon as a segment t^X is shortened, it is not relatively open any more.

(C) Neither (i) nor (ii) can be applied, $\mathcal{Q}'$ is still nonempty, and no square in $\mathcal{Q}'$ is disjoint from all segments t^X. That is, each square intersects at least two of the segments t^X, and no square contains one of those segments. Actually, both conditions combined imply that every square intersects exactly two segments (because, if a square intersects three of the sides of R_0, it must contain one of the sides).

Note that in all we have said so far we never used the fact that we are handling congruent squares; everything also holds for general axis-parallel rectangles. We now claim that, for translates of a square, case (C) already implies 4-pierceability. For that purpose, let P' be the set of counterclockwise endpoints of the segments t^X (the first endpoint of each segment that we encounter as we traverse ∂R_0 in clockwise direction), and let P'' be the set of clockwise endpoints. We want to prove that either P' or P'' is a piercing set for $\mathcal{Q}'$.

Any square in $\mathcal{Q}'$ containing the upper right vertex of R_0 must intersect t^T and t^R; more precisely, it must contain the counterclockwise endpoint of t^R and the clockwise endpoint of t^T. In other words, such a square is pierced by both P' and P''. More generally speaking, a square containing a vertex of the location domain is pierced by both P' and P''.

Any square q in $\mathcal{Q}'$ disjoint from the vertices of R_0 must intersect two opposite sides of R_0 in order to meet two of the segments t^X. Note that if q intersects the top and bottom side of R_0, but is disjoint from the vertices, then those sides must have length exceeding one and so no square in $\mathcal{Q}'$ can intersect both the left and right side of R_0 (because all squares are translates of a unit square). So, without loss of generality, we may assume that all squares disjoint from the vertices intersect the top and bottom sides of R_0.

The segments t^X have length at most one. Hence it is not possible for a square in $\mathcal{Q}'$ to intersect t^T (and t^B) without containing one of its endpoints. If a square contains the counterclockwise endpoint of one of the two segments, and the clockwise endpoint of the other segment, then it is pierced by both P' and P''.

What is left are squares that only contain the clockwise endpoints of t^T and t^B, and squares that only contain the counterclockwise endpoints of t^T and t^B. If only one of the two situations occurs, then P'' (or P', respectively) pierces all squares. Simultaneous occurrence of both situations, however, is impossible: A square q of the first type has the counterclockwise endpoint of t^T to its left, and the counterclockwise endpoint of t^B to its right, and no square of unit side length can contain these two points. We have thus shown that either P' or P'' is a piercing set in situation (C).

For the time analysis, it suffices to consider the algorithm for deciding 4-pierceability for the case that no axis-parallel line intersects all squares, and there is no piercing set containing a vertex of R_0 – those decisions can be made in linear time, and handling either of these situations can be done in linear time, as noted earlier.

Now we simply have to perform operations (i) and (ii) until we end up in one of the situations (A), (B), or (C), when we can decide 4-pierceability in constant time (and produce a piercing set in linear time). In order to execute the operations (i) and (ii) efficiently, we sort the intersections of the boundaries of the squares in $\mathcal{Q}$ with the sides of R_0 along each such side (there are at most $4n$ such intersections). Now, whenever we shorten a segment t^X, we have to detect those squares which lose contact with t^X, or start to contain t^X. Any such square must have a side that intersects the part removed from t^X. We therefore simply traverse this part, and process each intersection point found there in a straightforward manner that takes $O(1)$ time. The most expensive part of the procedure is the sorting along the sides of R_0; all other steps can be performed in $O(n)$ time. Note that if the centers of the squares have already been sorted both by their x-coordinates and by their y-coordinates, then the sorting along the sides of R_0 can be done in $O(n)$ time, by merging the (already sorted) sequences of left sides and of right sides (or of top sides and of bottom sides) of the squares. We will use this observation in the 4-center algorithm, given below. We thus have:

Theorem 2.3 *(a) The 4-piercing problem for n translates of a square can be solved in $O(n \log n)$ time.*
(b) If the centers of the squares are presorted by their x-coordinates and by their y-coordinates, then 4-pierceability can be decided in $O(n)$ time.

2.3 4-piercing of rectangles

We are given a collection $\mathcal{R} = \{r_1, \ldots, r_n\}$ of n axis-parallel rectangles in the plane, and we wish to determine whether $\mathcal{R}$ can be pierced by four points.

Again, as in the preceding subsection, we may restrict ourselves to the situation that all four piercing points lie on the boundary of the location domain, each lying in the relative interior of a distinct side.

This case is handled as follows. Let s denote the top side of R_0, and let t denote the intersection of s with the rectangle whose bottom side lies on the line ℓ^T. Let $I_j = t \cap r_j$, for $j = 1, \ldots, n$. The endpoints of the intervals I_j partition t into at most $2n-1$ 'atomic' intervals. We iterate through these intervals from left to right, and attempt to place the top piercing point p^T in each of them. For an atomic interval J, the set $\mathcal{R}_J$ of rectangles of $\mathcal{R}$ that do not contain p^T remains unchanged as p^T varies within J. When we pass from J to the next interval J', the set $\mathcal{R}_{J'}$ differs from $\mathcal{R}_J$ by a single rectangle being added or removed. For each interval J we check whether $\mathcal{R}_J$ is 3-pierceable. If this is true for at least one interval J, then $\mathcal{R}$ is 4-pierceable; otherwise $\mathcal{R}$ is not 4-pierceable.

We next describe a dynamic data structure on the rectangles in $\mathcal{R}_J$, using which we can determine whether $\mathcal{R}_J$ is 3-pierceable in $O(\log^3 n)$ time; the structure can also be updated in $O(\log^3 n)$ time when a rectangle is added to or is removed from $\mathcal{R}_J$. Recalling the algorithm for testing for 3-pierceability, we see that the data structure has to support an efficient implementation of two main operations:

(a) For a given pre-stored ('canonical') subset of rectangles of $\mathcal{R}$, find the corresponding lines ℓ^L, ℓ^R, ℓ^T, ℓ^B.

(b) For a given pre-stored subset $\mathcal{R}'$ of rectangles of $\mathcal{R}$ and for a query point v, find the set $\mathcal{R}'_v$ of rectangles of $\mathcal{R}'$ that do not contain v.

In a single step of the algorithm, both operations will be performed on a polylogarithmic number of canonical subsets, organized in a three-level structure (described in detail shortly), whose union is equal to a specific subset $\mathcal{R}^*$. It is important to observe that both operations (a) and (b) are *decomposable*, in the sense that the output for $\mathcal{R}^*$ can be easily obtained from the outputs for the canonical subsets of $\mathcal{R}^*$, in time proportional to the number of subsets: For (a), one simply has to take the leftmost of all the lines ℓ^L produced for each canonical subset, and symmetrically for the other three lines. The output of (b) for $\mathcal{R}^*$ is simply the union of the outputs for the canonical subsets of $\mathcal{R}^*$. (This may cause a canonical subset to appear more than once in the output for $\mathcal{R}^*$, but this does not affect the way further steps of the algorithm are performed, as follows easily from the description given below.)

To perform both (a) and (b) on a canonical set $\mathcal{R}'$, we prepare four balanced binary search trees, storing, respectively, the left, right, top, and bottom edges of the rectangles of $\mathcal{R}'$, sorted by their x-coordinates (for the first two trees) or by their y-coordinates (for the last two trees). Each node u of each of these trees represents a canonical subset $\mathcal{R}'_u$ of $\mathcal{R}'$, consisting of all rectangles stored at the leaves of the subtree rooted at u. Unless we are at the bottommost (third) level of the structure, we construct, in the next deeper level of the structure, a similar structure for each of the sets $\mathcal{R}'_u$.

Performing an operation of type (a) is easy: simply pick up the first or last leaf of each tree, as appropriate, and use the corresponding rectangle sides to construct the desired lines. Performing an operation of type (b) is also easy. Suppose the query point v has coordinates (v_x, v_y). Search with v_x in the first two trees, and with v_y in the last two trees. In the first tree, obtain all rectangles whose left side lies in the halfplane $x > v_x$, as the disjoint union of $O(\log n)$ subtrees. Apply symmetric procedures to the three other trees. In total, we obtain the output to (b) as the (not necessarily disjoint) union of $O(\log n)$ canonical subsets (that is, subtrees of the four trees).

In order to facilitate simple and efficient updating of the structure, we use a slightly modified variant, in the spirit of the structure in [39]. First we construct the above three-level structure for the entire set $\mathcal{R}$, using minimum-height binary trees at each level. Let $\mathcal{R}'$ be the current set to be maintained. We treat $\mathcal{R}'$ as the set of all n rectangles, except that some of them 'exist' and some are 'missing'. Each leaf is labeled as 'existing' or 'missing', and each internal node is labeled as 'existing' whenever at least one of the leaves in its subtree is 'existing'. Finding the leftmost (or rightmost) existing leaf is easy to do in $O(\log n)$ time. It is also easy to perform an operation of type (b) in $O(\log n)$ time, and obtain its output as a collection of $O(\log n)$ ('existing', i.e., nonempty) subtrees. Inserting or deleting a rectangle (into a single tree of the structure) is now easy to do, in $O(\log n)$ time, by going to the leaf storing the rectangle, switch its status from 'missing' to 'existing' or vice versa, and updating the status of nodes along the path from the leaf to the root, in constant time per node, in a straightforward bottom-up manner. Since the structure has several levels, we also need to update the deeper-level structures of the nodes that were updated in the current level. Each such updating is done recursively in the same manner. Since the structure has three levels, each update takes $O(\log^3 n)$ time.

We can now describe the full algorithm:

(i) Construct the rectangle R_0 and obtain the partition of the top interval t, as defined above, into up to $2n-1$ atomic intervals.

(ii) Iterate over these intervals from left to right. For each interval J, maintain the set $\mathcal{R}_J$ of rectangles of $\mathcal{R}$ not containing J. Initialize the data structure with the set of rectangles not containing the left endpoint of t.

(iii) Let $R_0(J)$ denote the location domain for $\mathcal{R}_J$. It is easily seen that the lines incident to the left, right and bottom sides of $R_0(J)$ are the same lines ℓ^L, ℓ^R, ℓ^B defining R_0, and only the top side of $R_0(J)$ changes (assuming general position, it is now lower than ℓ^T). Find the new top side (an operation of type (a)). (Note that this side can get as low as ℓ^B; in this case ℓ^B traverses $\mathcal{R}_J$. Testing for 3-pierceability of $\mathcal{R}_J$ in this case can also proceed along the steps outlined below, appropriately customized to this special case.)

(iv) It is easily seen from our assumptions on the location of the piercing points that one of the three piercing points of $\mathcal{R}_J$, if they exist, can be assumed to be one of the two top vertices of $R_0(J)$. We attempt to place the second piercing point at each of these vertices. Let v be that vertex, and, without loss of generality, assume it is the top-left vertex of $R_0(J)$.

(v) We compute the set $\mathcal{R}_{(J,v)}$ of all rectangles in $\mathcal{R}_J$ that do not contain v; this is an operation of type (b), and its output consists of $O(\log n)$ canonical subsets, whose union is $\mathcal{R}_{(J,v)}$. We need to test whether $R_{J,v}$ is 2-pierceable. Again, our assumptions on the piercing points are easily seen to imply that the two piercing points can be assumed to be the top-right vertex and the bottom-left vertex of the corresponding location domain $R_0(J,v)$. It is also easily verified that the bottom and right edges of $R_0(J,v)$ lie on the same lines ℓ^B, ℓ^R as the corresponding sides of the original R_0.

(vi) We find the left side of $R_0(J,v)$. This is done by computing the leftmost right edge of each of the canonical subsets produced in step (v) above, accessing the corresponding trees in the second level of our data structure, and by finding the leftmost of those output edges. In the same manner we find the top side of $R_0(J,v)$. The total cost of this step is $O(\log^2 n)$. Let w and z denote, respectively, the top-right vertex and the bottom-left vertex of $R_0(J,v)$.

(vii) We thus need to determine whether $\mathcal{R}_{(J,v)}$ is 2-pierceable by the points w and z. To do this, we construct the subset $\mathcal{R}_{(J,v,w)}$ of those rectangles of $\mathcal{R}_{(J,v)}$ that do not contain w, and then finally test whether $\mathcal{R}_{(J,v,w)}$ is pierceable by z. To construct $\mathcal{R}_{(J,v,w)}$, we access the second-level trees of each of the canonical subsets produced in step (v), and perform a type (b) operation (with w as a query point) on each of them. The union of all resulting canonical subsets is the desired $\mathcal{R}_{(J,v,w)}$. We next determine whether each of these subsets is pierced by z. To do so, we access the third-level structure of each subset, and determine whether z lies to the right of all the left edges of the rectangles in the subset, to the left of all the right edges, above all the bottom edges and below all the top edges. If all this holds, for each canonical subset, then $\mathcal{R}_{(J,v,w)}$ is pierceable by z; otherwise, it is not pierceable by z. The total cost of this step is easily seen to be $O(\log^3 n)$.

(viii) If $\mathcal{R}_J$ has been determined not to be 3-pierceable, we move to the next atomic interval J' of t. We update our data structure with the insertion or deletion of the rectangle by which $\mathcal{R}_J$ and $\mathcal{R}_{J'}$ differ, and repeat the above steps to J'. As mentioned above, the cost of the update is $O(\log^3 n)$.

(ix) If none of the $\mathcal{R}_J$'s is found to be 3-pierceable, we conclude that $\mathcal{R}$ is not 4-pierceable.

The overall running time of the algorithm is easily seen to be $O(n \log^3 n)$. Hence we have:

Theorem 2.4 *The 4-piercing problem for n axis-parallel rectangles can be solved in $O(n \log^3 n)$ time.*

2.4 5-piercing

Let $\mathcal{R}$ be a collection of n axis-parallel rectangles, as above. We want to determine whether $\mathcal{R}$ can be pierced by five points. Let R_0 denote the location domain of $\mathcal{R}$ as defined above. Arguing as in the preceding subsections, we may assume that one of the following situations occurs:

(i) One of the piercing points lies at a vertex of R_0.

(ii) The five piercing points all lie on the boundary of R_0, but none of them lies at a vertex. Each side of R_0 contains at least one point (and one side contains two points).

(iii) Four of the piercing points lie on the boundary of R_0, with one point lying in the relative interior of each side, and the fifth point lies in the interior of R_0.

Testing for case (i) is easy: We try each of the four vertices of R_0 as the first piercing point, find the set of rectangles not containing that vertex, and test whether this set is 4-pierceable, using the preceding algorithm. This takes $O(n \log^3 n)$ time.

To test for case (ii), we guess a side s of R_0 that contains just one piercing point, say it is the top side, and apply a procedure similar to that given for the 4-piercing problem. That is, we construct the interval t and its partition into atomic subintervals, as above, and iterate over these subintervals from left to right, maintaining the set $\mathcal{R}_J$ of rectangles not containing the current interval J. It is easy to check that one of the four remaining piercing points must lie at one of the two top vertices of $R_0(J)$, the location domain for the set $\mathcal{R}_J$. We compute $R_0(J)$ and attempt to place the second piercing point at either of its top vertices, call it v. We find the set $\mathcal{R}_{(J,v)}$ of those rectangles of $\mathcal{R}_J$ that do not contain v, and test whether this set is 3-pierceable, following the procedure described in the preceding section. The data structure that we need here is almost identical to that used above, except that now it has four levels instead of three. Omitting the easy missing details, we obtain a procedure that runs in $O(n \log^4 n)$ time.

To test for case (iii), let q be the piercing point lying inside R_0. Either at least one of the piercing points on the left and right sides of R_0 lies above q or at least one of these points lies below q. See Figure 2. Suppose that one of them lies above q. We then 'guess' the piercing point lying on the top side of R_0, using the interval t and its partition into atomic subintervals, as above. For each atomic interval J, we maintain the set $\mathcal{R}_J$ of rectangles not containing J, and observe that, in the assumed configuration, one of the four piercing points of $\mathcal{R}_J$, if they exist, must be a (top) vertex of $R_0(J)$. This is because $R_0(J)$ differs from R_0 only by its top side, which is lower, and must pass through one of the two piercing points lying on the left and right sides of R_0, because the highest of these two points lies above q, by assumption. It therefore suffices to test whether $\mathcal{R}_J$ is 4-pierceable, with one piercing point lying at a vertex of $R_0(J)$. This can be tested as in case (ii) above, implying that the testing for case (iii) can also be done in $O(n \log^4 n)$ time. Hence we obtain:

Theorem 2.5 *The 5-piercing problem for a set of n axis-parallel rectangles can be solved in $O(n \log^4 n)$ time.*

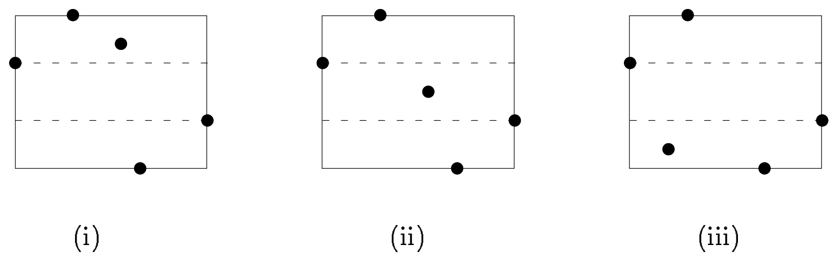

Figure 2: Guessing the bottom piercing point (in cases (i,ii)) or the top piercing point (in cases (ii,iii)) causes one of the 4 remaining piercing points to lie at a vertex of the location domain $R_0(J)$.

Remark. The above approach fails for the 6-piercing problem. One reason is that two of the piercing points might lie inside R_0 in such a way that, no matter which of the four border piercing points we eliminate, the resulting 5-piercing subproblem need not have a piercing point at a vertex of the associated location domain. In this case we do not know how to test for this 5-pierceability in polylogarithmic time. The best algorithm for 6-pierceability that we can design takes near quadratic time (it guesses simultaneously two piercing points on the boundary of R_0). We do not know whether this problem can be solved in subquadratic time.

Remark. Extending the last observation, one can easily show that the general rectilinear p-piercing problem, for $p \geq 5$, can be solved in time $O(n^{p-4}\log^5 n)$. However, as already remarked in the introduction, we expect that much faster algorithms can be obtained.

3 Rectilinear p-Center Problems

We choose the following set-up of the problem: Let $\mathcal{R}$ be a set of n compact convex regions with nonempty interior, 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 homothetic copy of r obtained by scaling r by the factor λ about c_r (i.e., $r(\lambda) = \{c_r + \lambda(a - c_r) \mid a \in r\}$). Finally, $\mathcal{R}(\lambda) = \{r(\lambda) \mid r \in \mathcal{R}\}$.

The p-center problem for $\mathcal{R}$ asks for

$$\lambda_{\mathcal{R}} := \min\{\lambda \mid \mathcal{R}(\lambda) \text{ is } p\text{−pierceable}\} .$$

If $\mathcal{R}$ is a set of translates of a square and the scaling points are the respective centers, then we talk about the (standard) *rectilinear p-center problem.* If the squares are still axis-parallel but of possibly different sizes (and again the scaling points are the centers), 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.*

3.1 General rectilinear 2- and 3-center problems

Given a set $\mathcal{R}$ of axis-parallel rectangles with scaling points, we want to determine $\lambda''_{\mathcal{R}}$ (and $\lambda'''_{\mathcal{R}}$), the smallest λ for which $\mathcal{R}(\lambda)$ is 2-pierceable (3-pierceable, respectively). We will use two tools for establishing a linear time bound, the LP-type framework and Helly-type results, which we review first.

LP-type problems. An *LP-type* problem is a pair (H, w) where H is a finite set (whose elements are called *constraints*), and w is a mapping from 2^H into some totally ordered set, so that the following conditions are satisfied:

(*Monotonicity*) For any F, G with $F \subseteq G \subseteq H$, we have $w(F) \leq w(G)$.
(*Locality*) For any $F \subseteq G \subseteq H$ with $w(F) = w(G)$ and any $h \in H$,

$$w(G) < w(G \cup \{h\}) \text{ implies that also } w(F) < w(F \cup \{h\}).$$

A *basis* B is a set of constraints with $w(B') < w(B)$ for all proper subsets B' of B. B is *a basis of* G, for $G \subseteq H$, if $B \subseteq G$ is a basis and $w(B) = w(G)$.

LP-type problems serve as a framework to solve certain optimization problems. Solving an LP-type problem (H, w) means to compute a basis of H, using the following *basic operations*.

(*Violation test*) For a constraint h and a basis B, test whether h is violated by B, i.e., whether $w(B) < w(B \cup \{h\})$.

(*Basis computation*) For a constraint h and a basis B, compute a basis of $B \cup \{h\}$.

For the efficiency of algorithms solving LP-type problems, the following parameter is crucial: The maximum cardinality of any basis is called the *combinatorial dimension of* (H, w).

Lemma 3.1 ([33]) *An LP-type problem of combinatorial dimension d with $n > d$ constraints can be solved by a randomized algorithm with an expected number of at most $2^{d+3}(n-d)$ basic operations.*

Remark. Combining the algorithm with other randomized algorithms by Clarkson [7] and Gärtner [20], and applying a more careful analysis, the expected running time (in terms of basic operations) of the algorithm can be reduced to $O(dn + e^{O(\sqrt{d \log d})})$; see, e.g., [21].

Helly-type results for rectilinear p-piercing. For $p \geq 1$, let $h_{p,d}$ be the smallest integer such that a finite set of axis-parallel boxes in d dimensions is p-pierceable, if and only if any $h_{p,d}$ (or fewer) of the boxes are p-pierceable.

Lemma 3.2 [8] (cf [11], Theorem 5.5) *(a) $h_{p,1} = p+1$, for any $p \geq 1$.*
(b) $h_{1,2} = 2$, $h_{2,2} = 5$, $h_{3,2} = 16$, and $h_{p,2}$ is undefined for $p \geq 4$.
(c) For $d \geq 3$, one has $h_{1,d} = 2$, $h_{2,d} = 3d$ if d is even and $h_{2,d} = 3d - 1$ if d is odd, and $h_{p,d}$ is undefined for $p \geq 3$.

In two dimensions, the result stays the same if we restrict ourselves to translates of a square instead of arbitrary axis-parallel rectangles. For our purpose, this yields the following corollary.

Corollary 3.3 *If $\mathcal{R}$ is a set of axis-parallel rectangles with scaling points, then there exists a subset $\mathcal{G}_2$ of at most 5 rectangles such that $\lambda''_{\mathcal{R}} = \lambda''_{\mathcal{G}_2}$ and there is a subset $\mathcal{G}_3$ of at most 16 rectangles such that $\lambda'''_{\mathcal{R}} = \lambda'''_{\mathcal{G}_3}$.*

For a proof, note that for any $\lambda < \lambda''_{\mathcal{R}}$, $\mathcal{R}(\lambda)$ is not 2-pierceable, and so there must be some set of at most 5 rectangles in $\mathcal{R}(\lambda)$ which is not 2-pierceable. Hence, there must be a set $\mathcal{G}$ of at most 5 rectangles in $\mathcal{R}$, such that $\mathcal{G}(\lambda)$ is not 2-pierceable for any $\lambda < \lambda''_{\mathcal{R}}$ and so $\lambda''_{\mathcal{G}} = \lambda''_{\mathcal{R}}$. An analogous argument holds for 3-piercing.

The relation between finite Helly numbers and LP-type problems of constant combinatorial dimension has been worked out in some generality in [3, 4]. Roughly speaking, as shown there, constant combinatorial dimension implies a finite Helly number property, but not vice versa! We will give below details concerning this relation in the special case of the rectilinear 2- and 3-center problems.

The p-center problem in one dimension as an LP-type problem. Before considering the 2-dimensional case, let us comment on the 1-dimensional p-center problem. A fairly involved technique of Frederickson [15] gives a linear-time solution, provided that *the points are already sorted* along the real line (this result actually works for arbitrary trees, where sorting comes for free). The actual running time of Frederickson's algorithm is $O(n)$. However, it follows from Lemma 3.2(a), combined with the machinery of [3, 4], that the 1-dimensional p-center problem is an LP-type problem, for any p (see below for a similar analysis in two dimensions), and thus can be solved in randomized expected $O(n)$ time, for any fixed p. In this bound, though, the combinatorial dimension is $O(p)$, so the constant of proportionality depends 'subexponentially' on p (see [33]), so Frederickson's solution (including the presorting stage) is far better unless p is very small.

The 2-center problem as an LP-type problem. (Recall that the 2-center problem already has a linear-time solution [10]. It is treated here both to highlight the use of the LP-type machinery to solve this problem, and to serve as a basis for the solution of the 3-center problem.) If we simply define $w(\mathcal{G}) := \lambda''_{\mathcal{G}}$ for $\mathcal{G} \subseteq \mathcal{R}$, then the system $(\mathcal{R}, w)$ is not LP-type because locality may be violated. To see this, consider the case where $\mathcal{R}$ consists of unit squares with scaling points at their centers, and represent each subset of $\mathcal{R}$ by the set of the centers of its squares. Now consider the sets $F = \{(1,0),(0,1),(-1,0),(0,-1)\}$ and $G = F \cup \{(0.5, 0.5)\}$, and let $h = (-0.5, 0.5)$. We have $w(G) = w(F) = 1$: either set is 2-pierceable as is (i.e., with scaling factor 1), but neither is 2-pierceable with a smaller scaling factor. However, as is easily checked, $w(G \cup \{h\}) = 1.5 > w(G)$, while $w(F \cup \{h\}) = 1 = w(F)$, so locality is indeed violated.

To overcome this difficulty, we will define a unique canonical 2-piercing set of a set of rectangles, provided it is 2-pierceable, and use this canonical set as part of the value of w. Recall the definition, given in the beginning of Section 2, of the lines ℓ^X, for $X = L, R, T, B$, for a nonempty set $\mathcal{R}$ of axis-parallel rectangles. Define the vertex v^{XY} as the intersection of ℓ^X and ℓ^Y for $X = L, R$ and $Y = B, T$.

Given a set $\mathcal{R}$ of axis-parallel rectangles, we define six *2-piercing candidate sets* by

$$C^{(0)} = \emptyset, C^{(1)} = \{v^{RT}\}, C^{(2)} = \{v^{LT}, v^{RT}\},$$
$$C^{(3)} = \{v^{RB}, v^{RT}\}, C^{(4)} = \{v^{LB}, v^{RT}\}, C^{(5)} = \{v^{LT}, v^{RB}\} .$$

Suppose that $\mathcal{R}$ is 2-pierceable. If $\mathcal{R}$ is empty then $C^{(0)}$ pierces $\mathcal{R}$. If $\mathcal{R}$ is 1-pierceable, then $C^{(1)}$ pierces $\mathcal{R}$. If there is a horizontal line intersecting all rectangles, then $C^{(2)}$ pierces $\mathcal{R}$, and if a vertical line intersects all rectangles, then $C^{(3)}$ pierces $\mathcal{R}$. If no axis-parallel line intersects all rectangles, then the location domain exists, and $C^{(4)}$ or $C^{(5)}$ is piercing. We summarize in a lemma.

Lemma 3.4 *$\mathcal{R}$ is 2-pierceable if and only if one of the 2-piercing candidate sets pierces $\mathcal{R}$.*

The *canonical 2-piercing set $P_\mathcal{R}$ of $\mathcal{R}$* is the 2-piercing candidate of smallest index that pierces $\mathcal{R}$, and the index of this candidate is called the *characteristic 2-piercing index $\chi_\mathcal{R}$ of $\mathcal{R}$*. If $\mathcal{R}$ is not 2-pierceable, then we set $\chi_\mathcal{R} = 6$ and $P_\mathcal{R} = \perp$, with $\perp$ representing a nominal undefined value.

Now let $\mathcal{R}$ be a set of axis-parallel rectangles with scaling points assigned. For $\mathcal{G} \subseteq \mathcal{R}$, we define $w(\mathcal{G})$ as the tuple $(\lambda''_\mathcal{G}, -x^L, x^R, -y^B, y^T, \chi_{\mathcal{G}(\lambda''_\mathcal{G})})$, where x^L, x^R are the x-coordinates of ℓ^L, ℓ^R, and y^B, y^T are the y-coordinates of ℓ^B, ℓ^T, as defined for the scaled set $\mathcal{G}(\lambda''_\mathcal{G})$. Note that $w(\mathcal{G})$ describes a unique candidate piercing set, including the exact coordinates of the piercing point(s). The ordering on such tuples is lexicographic, with the first entry most significant.

We have to show monotonicity of w. As for the first component, it is clear that $\lambda''_\mathcal{F} \leq \lambda''_\mathcal{G}$ for $\mathcal{F} \subseteq \mathcal{G} \subseteq \mathcal{R}$. The following lemma clarifies that adding a rectangle to a set $\mathcal{G}$ cannot decrease its characteristic 2-piercing index.

Lemma 3.5 *Let $\mathcal{R}$ be a 2-pierceable set of axis-parallel rectangles.*
(0) $\chi_\mathcal{R} = 0$ iff $\mathcal{R}$ is empty.
(1) $\chi_\mathcal{R} = 1$ iff $\mathcal{R}$ is nonempty and 1-pierceable.
(2) $\chi_\mathcal{R} = 2$ iff $\mathcal{R}$ is not 1-pierceable, and there is a horizontal line intersecting all rectangles.
(3) $\chi_\mathcal{R} = 3$ iff $\mathcal{R}$ is not 1-pierceable, and there is a vertical line intersecting all rectangles.
(4) $\chi_\mathcal{R} = 4$ iff there is no axis-parallel line intersecting all rectangles, and $C^{(4)}$ is piercing.
(5) $\chi_\mathcal{R} = 5$ iff there is no axis-parallel line intersecting all rectangles, and $C^{(4)}$ is not piercing.

The conditions for (2) and (3) are disjoint because, if there is both a vertical and a horizontal line intersecting all rectangles, then the set is 1-pierceable.

A short scan through the items in the lemma shows that adding a rectangle cannot decrease the characteristic 2-piercing index, with the step from (5) to (4) being the only non-trivial case. To see also this case, assume that we add a rectangle to $\mathcal{R}$ with $\chi_\mathcal{R} = 5$, and the resulting set $\hat{\mathcal{R}}$ is pierceable by its fourth candidate $\hat{C}^{(4)} = \{\hat{v}^{LB}, \hat{v}^{RT}\}$. Compare the points with the fourth candidate $C^{(4)} = \{v^{LB}, v^{RT}\}$ of $\mathcal{R}$. $\hat{v}^{LB}$ must lie in the third quadrant anchored at v^{LB}, and thus moving from $\hat{v}^{LB}$ to v^{LB} we cannot exit a rectangle in $\mathcal{R}$. Similarly, v^{RT} pierces all rectangles in $\mathcal{R}$ which are pierced by $\hat{v}^{RT}$. So if $\hat{C}^{(4)}$ is piercing $\hat{\mathcal{R}}$ (and thus $\mathcal{R}$), then $C^{(4)}$ is piercing $\mathcal{R}$, contradicting the assumption that $\chi_\mathcal{R} = 5$.

So the first component in the w-value of a set cannot decrease if a new rectangle is added. The same holds for all the other components, as is easily verified for the next four components, and as follows from the lemma for the last component. We have thus established monotonicity of w.

As for locality, consider $\mathcal{F} \subseteq \mathcal{G} \subseteq \mathcal{R}$ with $w(\mathcal{F}) = w(\mathcal{G}) = (\lambda, -x_1, x_2, -y_1, y_2, \chi)$ and a rectangle r with a scaling point. Note that $w(\mathcal{G}) < w(\mathcal{G} \cup \{r\})$, iff $r(\lambda)$ is disjoint from the piercing candidate set described by $w(G)$, which is equivalent to the condition $w(\mathcal{F}) < w(\mathcal{F} \cup \{r\})$. This proves locality.

We are left with the task of proving that the combinatorial dimension of $(\mathcal{R}, w)$ is bounded by a fixed constant. That is, we need to bound the number of rectangles in a set $\mathcal{G} \subseteq \mathcal{R}$ that determine $w(\mathcal{G})$. At most five rectangles determine $\lambda''(\mathcal{G})$. At most four other rectangles may be needed to fix the lines ℓ^X, for $X = L, R, T, B$ (and thus fix the points v^{XY}, for $X = L, R$ and $Y = B, T$). If $\chi_{\mathcal{G}} = i$, then we may need up to $i - 1$ additional rectangles to enforce that index (for each $1 \leq j < i$ we need a rectangle $r \in \mathcal{G}$, so that $r(\lambda''_{\mathcal{G}})$ is disjoint from $C^{(j)}$). We obtain an upper bound of $5 + 4 + 4 = 13$ on the maximal size of a basis (remember that $\chi_{\mathcal{G}(\lambda''_{\mathcal{G}})} < 6$). The actual bound is smaller, since we can 'recycle' some of the rectangles in the argument.

The two types of basic operations are easy to implement in constant time. Lemma 3.1 implies:

Theorem 3.6 *The general rectilinear 2-center problem for n rectangles with scaling points can be solved in expected $O(n)$ time by a randomized algorithm.*

The 3-center problem as an LP-type problem. Given a nonempty 3-pierceable set $\mathcal{R}$ of axis-parallel rectangles, let $v_{\mathcal{R}}$ be the first point v in the sequence $(v^{RT}, v^{RB}, v^{LT}, v^{LB})$ such that the set of rectangles in $\mathcal{R}$ disjoint from v, denoted by $\mathcal{R} \ominus v$, is 2-pierceable ($v_{\mathcal{R}}$ exists; see Subsection 2.1). We put $\xi_{\mathcal{R}} = i$, if $v_{\mathcal{R}}$'s occurrence in this sequence is in position i, for $1 \leq i \leq 4$. We extend this notion to $\xi_{\emptyset} = 0$, $v_{\emptyset} = \perp$, and for $\mathcal{R}$ not 3-pierceable, $\xi_{\mathcal{R}} = 5$, $v_{\mathcal{R}} = \perp$.

Let $\mathcal{R}$ be a set of axis-parallel rectangles with scaling points. For $\mathcal{G} \subseteq \mathcal{R}$, we define

$$u(\mathcal{G}) = (\lambda'''_{\mathcal{G}}, -x^L, x^R, -y^B, y^T, \xi_{\mathcal{G}'''}, \tilde{w}(\mathcal{G}''' \ominus v_{\mathcal{G}'''})),$$

where $\mathcal{G}''' = \mathcal{G}(\lambda'''_{\mathcal{G}})$, where x^L, x^R, y^B, and y^T are as defined above, for the scaled set $\mathcal{G}'''$, and where $\tilde{w}$ is the tuple w without its first component (here there is no need to specify another scaling factor). As before, one can show that $u(\mathcal{G})$ describes a unique candidate piercing set.

Ordering on tuples $(\lambda, -x_1, x_2, -y_1, y_2, \xi, \tilde{w})$ is again lexicographic from left to right. Monotonicity follows easily, as in the 2-center case. Locality follows again, because $u(\mathcal{G})$ encodes an explicit piercing set which determines whether $u(\mathcal{G} \cup \{r\}) > u(\mathcal{G})$ for some r (this happens iff $r(\lambda'''_{\mathcal{G}})$ is not pierced by that set).

For the combinatorial dimension, we have to find a bound on the number of rectangles in a set $\mathcal{G}$ necessary for determining $u(\mathcal{G})$: 16 rectangles enforce $\lambda'''_{\mathcal{G}}$; 4 rectangles fix the lines ℓ^X; for each $\xi = 2, 3, 4$, we need up to 5 rectangles for each preceding vertex v^{XY}, to show that the rectangles disjoint from v^{XY} are not 2-pierceable, thus enforcing the value of ξ. Finally, we need 4+4 more rectangles to enforce the value of $\tilde{w}$ (see the previous analysis). This gives a bound of $16 + 4 + 3 \times 5 + 8 = 43$ (which, again, can be improved by a more careful analysis). It requires an actual implementation to decide whether the randomized algorithm derived from the LP-type framework provides a practical method for the problem. At this point we have the following theoretical result.

Theorem 3.7 *The general rectilinear 3-center problem for n rectangles with scaling points can be solved in expected $O(n)$ time by a randomized algorithm.*

The algorithms in Theorems 3.6 and 3.7 can be derandomized at the cost of an increase in the constant of proportionality only; this is done using the technique of [6]. (Note that this technique also works in an abstract setting, which requires some extra condition beyond the LP-type axioms: For each basis B, let $\mathcal{G}_B$ be the set of all constraints not violated by B. Then we require that the set system $\{\mathcal{G}_B\}_B$ have finite VC-dimension. It is easily verified that this condition holds for the above center problems.)

Remark. One can apply Lemma 3.2(c), using a machinery similar to that above, to obtain a linear-time randomized algorithm for the general rectilinear 2-center problem in any dimension d. See also a related piercing result in [27].

3.2 Rectilinear 4- and 5-center problems

Let S be a set of n points in the plane. We want to find the smallest λ, so that S can be covered by the union of four (or five) axis-parallel squares of side length λ. We solve this problem by providing a general scheme for transforming a p-piercing algorithm for translates of squares to a rectilinear p-center algorithm. It is based on the matrix-searching technique of Frederickson and Johnson [16, 17, 18] (a previous recent and similar application of that technique can be found in [22]).

Specifically, we observe that if S can be covered by the union of p squares of size λ, then it can also be covered by the union of p squares of size smaller than λ, unless λ is equal to the difference between either the x-coordinates or the y-coordinates of a pair of points in S.

We therefore consider two $n \times n$ matrices X and Y, defined as follows. Let $(x_1, \ldots, x_n)$ be the sequence of x-coordinates of the points of S, sorted in increasing order. Then $X_{ij} = x_j - x_i$ if $j > i$ and $X_{ij} = 0$ otherwise. The matrix Y is defined symmetrically by the sorted y-coordinates of the points of S. We note that both matrices are *monotone*, in the sense that each row (resp. each column) is a monotone nondecreasing (resp. nonincreasing) sequence, and that the optimal λ^* is an entry of one of these matrices. Moreover, comparing λ^* with another λ can be done by applying a p-piercing algorithm to the set $\mathcal{Q}(\lambda) = \{s+\lambda Q \mid s \in S\}$, where Q is an axis-parallel square of size 1, centered at the origin. That is, $\mathcal{Q}(\lambda)$ is p-pierceable if and only if $\lambda \geq \lambda^*$. We can therefore apply the matrix searching technique of Frederickson and Johnson, as described in [16, 17, 18]. It finds the optimal λ^* by making only $O(\log n)$ calls to the p-piercing decision procedure. That is, we have shown:

Theorem 3.8 *If the p-piercing problem for n translates of a square can be solved in time $t_p(n)$, then the rectilinear p-center problem can be solved in $O(t_p(n) \log n)$ time.*

Based on Theorems 2.3(b) and 2.5, we get:

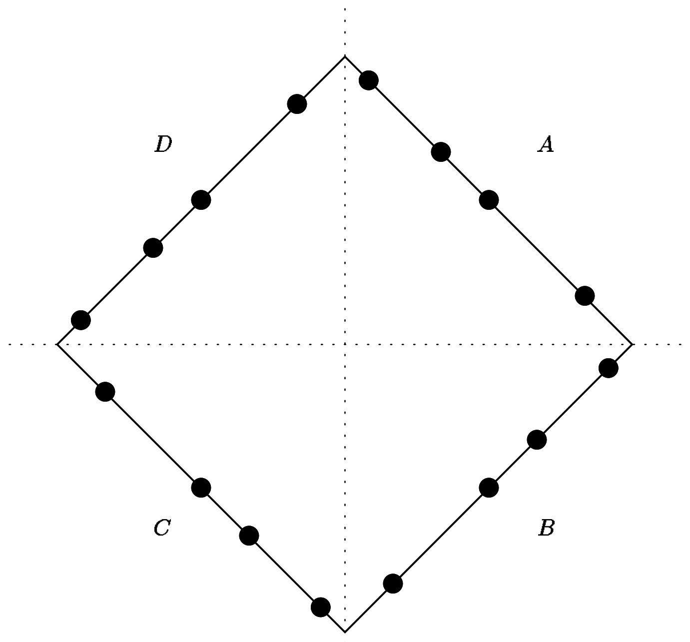

Figure 3: Transforming MAX-GAP to the rectilinear 4-center problem

Corollary 3.9 *The rectilinear 4-center problem can be solved in time $O(n \log n)$, which is worst-case optimal, and the rectilinear 5-center problem can be solved in time $O(n \log^5 n)$.*

The proof of the lower bound for the 4-center problem goes as follows. We reduce the MAX-GAP problem to the rectilinear 4-center problem. In the MAX-GAP problem we are given real numbers $0 \le x_1, x_2, \ldots, x_n \le 1$, and want to compute the maximum gap between two successive numbers in the sorted sequence. As is well known [32], the MAX-GAP problem has an $\Omega(n \log n)$ lower bound in the algebraic computation tree model.

Given an instance $0 \le x_1, x_2, \ldots, x_n \le 1$ of the MAX-GAP problem, we map each x_i to the four points

$$a_i = (x_i, 1 - x_i), \quad b_i = (1 - x_i, -x_i), \quad c_i = (-x_i, -1 + x_i), \quad d_i = (-1 + x_i, x_i) .$$

Let $A = \{a_i\}_{i=1}^n$, $B = \{b_i\}_{i=1}^n$, $C = \{c_i\}_{i=1}^n$, and $D = \{d_i\}_{i=1}^n$, and put $S = A \cup B \cup C \cup D$. All the points of S lie on the edges of a square K; see Figure 3. Add the four vertices of K to S, to obtain a new set S_0 of $4n+4$ points. Note that we can cover S_0 by four axis-parallel squares of size (edge length) 1. Moreover, if there is an index i such that two of the points a_i, b_i, c_i, d_i belong to the same covering axis-parallel square, then its size must be at least 1. Hence, in looking for a smaller solution, we may assume that, for each i, each square contains exactly one of the points a_i, b_i, c_i, d_i. Also, each such smaller square must contain a unique vertex of K. Let us sort each of the sets A, B in decreasing y order, and each of the sets C, D in increasing y order. It follows that one of the covering squares contains a suffix of D and a prefix of A, another square contains a suffix of A and a prefix of B, another square contains a suffix of B and a prefix of C, and the last square contains a suffix of C

and a prefix of D. Moreover, all four prefixes are images of the same subset of the points x_i, and thus so are all four suffixes. Suppose, with no loss of generality, that the numbers $x_1, \ldots, x_n$ are sorted in this order, that the prefixes are images of the numbers $x_1, \ldots, x_i$, and that the suffixes are images of $x_{i+1}, \ldots, x_n$. It follows that the smallest possible size of these squares is $1 - (x_{i+1} - x_i)$. Hence, to find the smallest size of four axis-parallel squares that cover S_0, we have to compute the maximum gap in the sequence $x_1, \ldots, x_n$. This implies the asserted lower bound.

We can also derive an $\Omega(n \log n)$ bound for the 4-piercing problem of axis-parallel squares. We use a similar reduction to that given above, except that for each point $p \in S_0$ we draw an axis-parallel square of size $\lambda < 1$ and centered at p. Using the same arguments as above, we see that this set is 4-pierceable if and only if the maximum gap in the sequence $x_1, \ldots, x_n$ is at least $1 - \lambda$. Since this latter GAP-EXISTENCE problem also has an $\Omega(n \log n)$ lower bound in the algebraic decision tree model [32][4], the claim follows.

Remark. The Frederickson-Johnson technique fails in the weighted case, where we need to apply full-fledged parametric searching [36] (see also [1, 2]). This results in $O(n \operatorname{polylog} n)$ solutions for both the weighted (or general) rectilinear 4- and 5-center problems. This technique requires an efficient parallel implementation of a generic simulation of the 4-piercing algorithm provided in Theorem 2.4, or of the 5-piercing algorithm provided in Theorem 2.5. Both implementations are not straightforward, since both algorithms employ a sequential iteration over the corresponding sequence of atomic intervals. Nevertheless, this iteration can be parallelized, using persistent range trees and related techniques. The other steps of the algorithms are fairly easy to parallelize. For the sake of brevity, we do not provide any further details, and leave them to the interested reader.

4 Polygonal 2-Piercing and 2-Center Problems

In this section we consider the 2-piercing and 2-center problems for convex polygons. In the 2-center problem, we are given a set S of n points in the plane, and a convex c-gon P, and we want to cover S with two homothetic copies of P whose maximum size is as small as possible. An equivalent formulation of the problem is as follows. Let $\tilde{P}$ be the reflected image of P through the origin. We want to find the smallest scaling factor λ for which the collection $\mathcal{P}(\lambda) = \{s + \lambda \tilde{P} \mid s \in S\}$ of isothetic polygons is 2-pierceable.

As a matter of fact, we consider the following more general 2-piercing problem: A collection $\mathcal{Q}$ of convex polygons is called *c-oriented* if the orientations of the edges of each $Q \in \mathcal{Q}$ belong to a fixed set of c orientations. (We will also allow degenerate cases, in which some members of $\mathcal{Q}$ may have fewer than c edges; we treat these cases by regarding each member of $\mathcal{Q}$ as a convex c-gon, where some of its sides have zero length.)

[4] Actually, an $\Omega(n \log n)$ lower bound is established in [32] for the UNIFORM-GAP problem: Given n real numbers $x_1, \ldots, x_n$ and a real $\delta > 0$, determine whether the gap between any pair of consecutive elements in the sorted sequence of the x_i's is exactly δ. It is easy to reduce the UNIFORM-GAP problem to the GAP-EXISTENCE problem, which thus also has the same lower bound in the algebraic decision tree model.

4.1 Polygonal 2-piercing

Let $\mathcal{Q} = \{Q_1, \ldots, Q_n\}$ be a collection of c-oriented convex polygons. We want to determine whether $\mathcal{Q}$ is 2-pierceable. Unlike the case where the given orientations come from the sides of a square (which is the rectilinear 2-piercing problem treated earlier), there is no general Helly-type result for 2-piercing of $\mathcal{Q}$ (such a result was conjectured by Danzer and Grünbaum [8] and recently refuted by Katchalski and Nashtir [26]). (It is also shown in [26] that there is a constant u such that, for any convex c-gon Q and corresponding collection $\mathcal{Q}$, if any cu (or fewer) polygons in $\mathcal{Q}$ are 2-pierceable then $\mathcal{Q}$ is 3-pierceable. Unfortunately, this result does not lead to any LP-type formulation of the corresponding 2-center problem.)

Nevertheless, such a Helly-type result was established in [26] for the case where $\mathcal{Q}$ is a set of homothetic triangles: If any 9 (or fewer) triangles in $\mathcal{Q}$ are 2-pierceable then $\mathcal{Q}$ is 2-pierceable. This result implies that the 2-center problem for homothets of a given triangle can be solved in expected linear time, using the machinery of LP-type problems, in a manner similar to that employed in Section 3.1. However, our analysis will also imply a simpler (and deterministic) linear-time solution for triangles.

For each edge orientation θ, let L_θ be the directed line at orientation θ that satisfies the following properties:

- L_θ passes through an edge e of some polygon $Q_\theta \in \mathcal{Q}$ (including degenerate edges, as above), so that Q_θ lies to the left of L_θ.
- No edge at orientation θ of any polygon in $\mathcal{Q}$ is contained in the open halfplane to the left of L_θ.

For simplicity of presentation, we assume, as above, that the polygons in $\mathcal{Q}$ are in general position, meaning that no two distinct edges of the polygons lie on the same line. Again, it is easily seen that the analysis given below continues to hold for collections $\mathcal{Q}$ not in general position. Let $\mathcal{L}$ be the resulting collection of c lines, and let $\mathcal{A}(\mathcal{L})$ denote their arrangement. See Figure 4. Note that if $\mathcal{Q}$ can be pierced by two points a, b, then the closed left halfplane bounded by each of the lines L_θ must contain either a or b (or else the corresponding polygon Q_θ will not be pierced).

Lemma 4.1 *(a) If $\mathcal{Q}$ can be pierced by two points, then it can be pierced by a pair of points lying on the lines of $\mathcal{L}$.*
(b) For $\mathcal{Q}$ a 2-pierceable collection of homothetic triangles or 4-oriented trapezoids, we may further assume that one of the piercing points lies at a vertex of $\mathcal{A}(\mathcal{L})$.

Proof: (a) Let a and b be two points that pierce $\mathcal{Q}$. We will show that a can be moved to an appropriate edge g of $\mathcal{A}(\mathcal{L})$, on the boundary of the face containing a, so that, during this motion, we do not exit any polygon in $\mathcal{Q}$. A similar property also holds for b, and this clearly implies the assertion of the lemma. The following claim is immediate from the definitions:

Claim: (i) Suppose that a lies in the left halfplane of some line L_θ. If a moves arbitrarily within this halfplane, it cannot exit any polygon of $\mathcal{Q}$ through an edge at orientation θ.
(ii) Suppose that a lies in the right halfplane of some line L_θ. If a moves within this halfplane

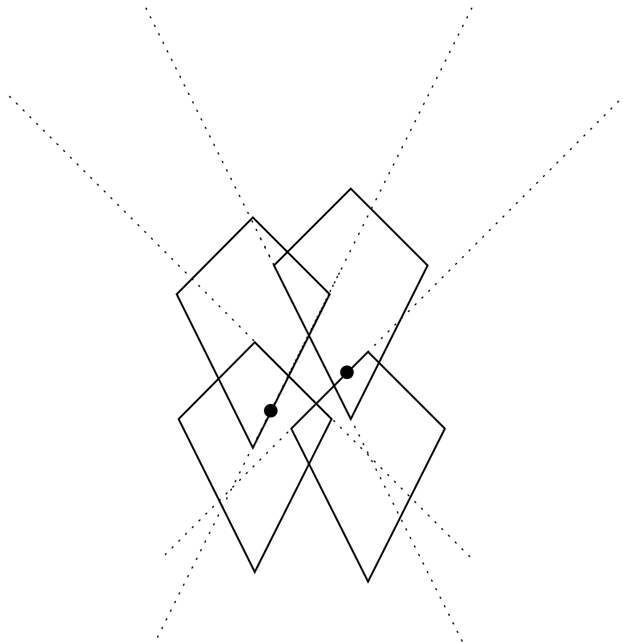

Figure 4: The arrangement $\mathcal{A}(\mathcal{L})$ for a 2-pierceable set of 4-oriented (actually isothetic) quadrilaterals

so that its distance from L_θ *decreases*, it cannot exit any polygon of $\mathcal{Q}$ through an edge at orientation θ.

Partition the set of the c given orientations into $\{\theta_1, \ldots, \theta_i\}$ and $\{\theta'_1, \ldots, \theta'_j\}$ (with $i + j = c$), so that a lies in the closed left halfplanes of $L_{\theta_1}, \ldots, L_{\theta_i}$, and in the open right halfplanes of $L_{\theta'_1}, \ldots, L_{\theta'_j}$. Let f be the face of $\mathcal{A}(\mathcal{L})$ containing a. The intersection of the closed left halfplanes bounded by $L_{\theta'_1}, \ldots, L_{\theta'_j}$ cannot be empty, because it must contain the point b (recall that each such halfplane contains either a or b, and none of them contains a). It follows that the intersection K of the corresponding open right halfplanes (which contains f) must be unbounded. Let ρ be a ray contained in K. Then it follows that, if we move from a in the direction opposite to ρ, we do not move away from any line $L_{\theta'_t}$. Hence, by the above claim, we reach the boundary of f without exiting any polygon in $\mathcal{Q}$.

(b) Before beginning the proof of the second part, we first note that there are situations in which one cannot enforce any piercing point to lie at a vertex of $\mathcal{A}(\mathcal{L})$, already for $c = 4$. Such a situation is shown in Figure 5. In this figure, it is easily checked that there is no pair of piercing points, one of which is a vertex of $\mathcal{A}(\mathcal{L})$. Each of the two piercing points shown in the figure lies on an edge of $\mathcal{A}(\mathcal{L})$, and if we attempt to move it towards any endpoint of that edge, it leaves some quadrilateral of $\mathcal{Q}$.

Nevertheless, the property that one can enforce one piercing point to lie at a vertex of $\mathcal{A}(\mathcal{L})$ does hold for $c = 3$ (the case of homothetic triangles), and for 4-oriented trapezoids. This is easy to show for the case of triangles, using the same analysis as above. The proof for the case of 4-oriented trapezoids goes as follows.

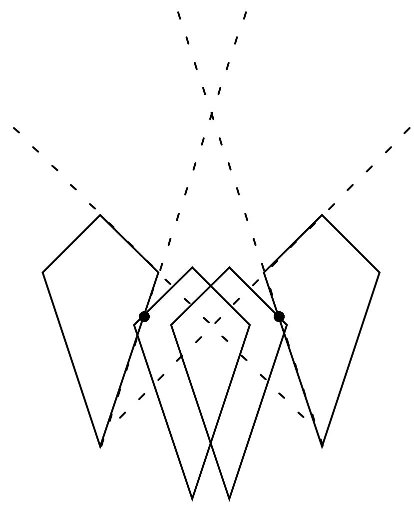

Figure 5: A configuration of a 2-pierceable collection of 4-oriented quadrilaterals where no piercing point can lie at a vertex of $\mathcal{A}(\mathcal{L})$

Suppose, with no loss of generality, that the bases of the given trapezoids are horizontal. The set $\mathcal{L}$ consists of four lines, two of which are horizontal and denoted by ℓ_1 and ℓ_2, and the other two are denoted by ℓ_3 and ℓ_4. Each line $\ell \in \mathcal{L}$ has an associated halfplane ℓ^+ that contains the trapezoid with a side lying on ℓ. As above, each halfplane ℓ^+ must contain at least one piercing point.

We use the following rule, which is a reformulation of the preceding analysis: Let p be a point in the plane. If we move p arbitrarily, so that its distance from ℓ^+ does not increase, for every $\ell \in \mathcal{L}$, then p does not exit any trapezoid of $\mathcal{Q}$.

Suppose that $\{p, q\}$ is a piercing set of $\mathcal{Q}$, so that each of these points lies on a line in $\mathcal{L}$. If one of these points, say p, belongs to all four halfplanes ℓ^+, then p pierces all trapezoids, and we can easily move it, within this intersection, to a vertex of $\mathcal{A}(\mathcal{L})$, which still pierces $\mathcal{Q}$. If one of these points, say p, belongs to three halfplanes ℓ^+ (but not to all four), then let v denote the vertex of the face containing p in $\mathcal{A}(\mathcal{L})$, such that v is nearest to the fourth line ℓ. We can then move p to v, so that its distance from ℓ does not increase, which, by the above rule, allows us to replace p by v in the piercing set.

In the remaining case, each of p and q lies in exactly two halfplanes ℓ^+. It is impossible that one of these points lies in $\ell_1^+ \cap \ell_2^+$, because the other point would also have to lie in one of these halfplanes (in this case, the union $\ell_1^+ \cup \ell_2^+$ is the entire plane). We may thus assume, with no loss of generality, that $p \in \ell_1^+ \cap \ell_3^+$ and that $q \in \ell_2^+ \cap \ell_4^+$. If one of the points, say p, lies on one of the horizontal lines, then we can move it along this line towards ℓ_4, and this motion is easily seen to be 'safe', in the sense that it will not cause p to exit any trapezoid, and it will eventually reach a vertex of $\mathcal{A}(\mathcal{L})$.

We may therefore assume that $p \in \ell_3$ and $q \in \ell_4$. Let w be the point of intersection of ℓ_3 and ℓ_4, and suppose that the vertical distance between p and w is larger than the vertical distance between q and w. We claim that moving p along ℓ_3 towards w is safe in the above sense. Indeed, if this were not the case, then p would be moving away from ℓ_2, in which case q, which lies in ℓ_2^+, would have to lie vertically further from w, contrary to assumption. This motion of p would eventually move it to a vertex of $\mathcal{A}(\mathcal{L})$, completing the proof of the second part of the lemma. □

Lemma 4.1(b) leads to the following new Helly-type result, which extends a similar result of [26], established for homothetic triangles.

Lemma 4.2 *Let $\mathcal{Q}$ be a collection of 4-oriented trapezoids. If every subcollection of* 22 *(or fewer) trapezoids of $\mathcal{Q}$ is 2-pierceable, then $\mathcal{Q}$ is 2-pierceable.*

Proof: Suppose to the contrary that this property holds but $\mathcal{Q}$ is not 2-pierceable. Let Q_1, Q_2, Q_3, Q_4 be the four trapezoids that define the lines of $\mathcal{L}$. The arrangement $\mathcal{A}(\mathcal{L})$ has six vertices. For each vertex v, let $\mathcal{Q}_v$ denote the subset of trapezoids of $\mathcal{Q}$ that do not contain v. By Lemma 4.1(b), $\mathcal{Q}_v$ is not 1-pierceable, that is, its intersection is empty. By the standard Helly theorem, $\mathcal{Q}_v$ contains three trapezoids whose intersection is empty. Collect six such triples of trapezoids, one for each vertex v of $\mathcal{A}(\mathcal{L})$, and add the four trapezoids $Q_1, \ldots, Q_4$, to obtain a subcollection $\mathcal{Q}^*$ of 22 trapezoids. By the above construction, and by Lemma 4.1(b), it follows that $\mathcal{Q}^*$ is not 2-pierceable, contrary to assumption. □

Renarks. (a) The number 22 in the above lemma can probably be made smaller—we have made no attempt to minimize its value.

(b) We leave it as an open problem to extend the results of Lemmas 4.1(b) and 4.2 to other families of c-oriented polygons.

Lemma 4.1(a) reduces the search for the two piercing points to a constant number of 1-dimensional searches (along the lines of $\mathcal{L}$). This leads to the following simple algorithm for finding the piercing points (when they exist):

(1) Fix a line $L_\theta \in \mathcal{L}$, and intersect each polygon in $\mathcal{Q}$ with L_θ, to obtain a system of intervals (some of which might be empty) along L_θ.

(2) Sort the endpoints of these intervals along L_θ, and obtain a partitioning of L_θ into *atomic intervals*, each delimited by two adjacent endpoints.

(3) Iterate through these atomic intervals, maintaining the set of polygons $\mathcal{Q}_I$ not containing the current interval I. To decide whether one of the piercing points a can be placed in I, we need to determine whether the polygons in $\mathcal{Q}_I$ have nonempty intersection. To determine this efficiently, we maintain, for each edge orientation θ, a priority queue H_θ of the edges at orientation θ of the polygons in $\mathcal{Q}_I$, in the order of the intercepts of the lines containing them. More precisely, e' precedes e'' in H_θ if the halfplane bounded by the line passing through e' and containing its polygon is contained in the corresponding halfplane for e''. We refer to these halfplanes as *inner halfplanes*. We pick the smallest element of each of these queues, and consider the c inner halfplanes bounded by the lines passing through these c edges. Clearly, the intersection of $\mathcal{Q}_I$ is nonempty if and only if these c halfplanes have a nonempty intersection, which can be tested in constant time.

(4) Since $\mathcal{Q}_I$ changes from one atomic interval to the other only by the insertion or deletion of a polygon (assuming general position), the priority queues can be updated in $O(c \log n)$ time for each atomic interval.

We have thus shown the upper bound of part (a) of the following

Theorem 4.3 *(a) Given a set $\mathcal{Q}$ of n convex c-oriented polygons, one can determine, in $O(c^2 n \log n)$ time, whether $\mathcal{Q}$ is 2-pierceable. For c fixed, this is worst-case optimal, in the algebraic decision tree model.*
(b) Let S be a set of n points in the plane, and let Q be a convex c-gon. Assume that the points of S are presorted in each of the directions orthogonal to the c orientations of the edges of Q. Then, for any given $\lambda > 0$, we can decide 2-pierceability of the set $\{s + \lambda Q \mid s \in S\}$ in linear time.
(c) If $\mathcal{Q}$ is a set of n homothetic triangles, or of n 4-oriented trapezoids, then 2-pierceability of $\mathcal{Q}$ can be decided in $O(n)$ time.

Proof: Part (c) easily follows from the fact that we can place one piercing point at a vertex of $\mathcal{A}(\mathcal{L})$, as argued in Lemma 4.1(b).

The lower bound in part (a) is proved as follows. We show that the problem of determining whether a collection of 4-oriented quadrilaterals is 2-pierceable requires $\Omega(n \log n)$ time in the algebraic decision tree model. The quadrilaterals in $\mathcal{Q}$ are constructed such that the set $\mathcal{L}$ is as shown in Figure 5, and such that the two piercing points must lie in the relative interiors of the edges e, e' of $\mathcal{A}(\mathcal{L})$, as in Figure 5. Moreover, we also assume that, for any choice of segments $g \subseteq e$, $g' \subseteq e'$, there exists a quadrilateral Q with the four given side orientations, such that $Q \cap e = g$ and $Q \cap e' = g'$. It is indeed easy to construct a collection of $O(1)$ quadrilaterals for which the above properties hold, and the actual collection $\mathcal{Q}$ will be formed by introducing additional quadrilaterals that intersect both e and e'.

Hence the problem, under these additional assumptions, is equivalent to the following one: Given two sequences of intervals, $(I_1, \ldots, I_n)$ and $(J_1, \ldots, J_n)$, on two respective lines, determine whether there exists a pair p, q of points, one on each line, such that for each $k = 1, \ldots, n$, either $p \in I_k$ or $q \in J_k$. We present a reduction from the Gap-Existence problem, already considered in Subsection 3.2: Given reals $x_1, \ldots, x_n$, and $\delta > 0$, determine whether the maximum gap in their sorted sequence is at least δ. As noted above, this problem has an $\Omega(n \log n)$ lower bound in the algebraic decision tree model.

Given an instance of the Gap-Existence problem, we first shift and scale it, so that $\min_i x_i = 0$ and $\max_i x_i = 1$. Put $\alpha = (1-\delta)/2$ (we may of course assume that $\delta < 1$), and define, for each $k = 1, \ldots, n$,

$$I_{2k-1} = [x_k - \alpha,\ x_k + \alpha]\,, \quad I_{2k} = [x_k - 1 - \alpha,\ x_k - 1 + \alpha]\,, \quad J_{2k-1} = I_{2k}\,, \quad J_{2k} = I_{2k-1}\,.$$

Let p be a point on the line containing the I-intervals. It pierces those intervals I_{2k-1} for which $x_k \in [p - \alpha,\, p + \alpha]$, and intervals I_{2k} for which $x_k \in [1 + p - \alpha,\, 1 + p + \alpha]$. See Figure 6. Note that the gap between these two intervals is δ. Hence, if the maximum gap in the sorted sequence of the x_i's is $\geq \delta$, we can find a point p such that for each $k = 1, \ldots, n$, either $x_k \leq p + \alpha$ or $x_k \geq 1 + p - \alpha$, and such that each of these inequalities is satisfied at least once. It follows that in this case $p - \alpha \leq 0$ and $1 + p + \alpha \geq 1$, which implies that for each $k = 1, \ldots, n$, either I_{2k-1} or I_{2k} is pierced by p (but not both). Hence, if we choose the same point p on the other line too, we pierce one interval out of each pair (I_ℓ, J_ℓ), for $\ell = 1, \ldots, 2n$. It is easy to see that the converse also holds: If there exists a solution p, q to the above instance of the 2-piercing problem, then each x_k lies in the union of $[p - \alpha,\, p + \alpha]$ and $[1 + p - \alpha,\, 1 + p + \alpha]$. Moreover, since the length of each of these intervals is $1 - \delta$, no interval can contain all points x_k. This implies that the maximum gap in the sequence x_k is at least δ. This completes the proof of the asserted lower bound in (a).

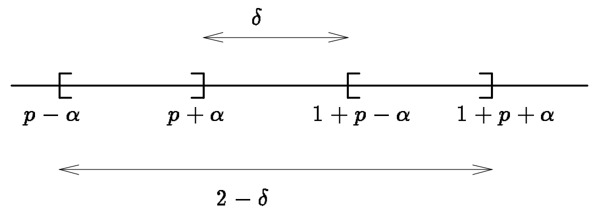

Figure 6: The intervals related to a piercing point

We next prove part (b). We pick a line $L \in \mathcal{L}$, and intersect each $Q \in \mathcal{Q}$ with L, to obtain a system of intervals. We classify the endpoints of these intervals into c subsets, according to the orientations of the polygon edges that are incident to these points. For each subset, the order of its points along L can be obtained in linear time from the presorted order of all the polygon edges with the corresponding orientation. By merging the resulting c ordered sequences, in linear time, we obtain the sorted sequence of the endpoints of all the atomic intervals along L. Let us denote this sequence symbolically by $(1, 2, \ldots, 2n)$ (without loss of generality, we assume that these points are distinct, and that every $Q \in \mathcal{Q}$ intersects L).

Next consider the maintenance of the priority queues. Let H_θ be one of these queues. Consider all the polygon edges that have orientation θ. By assumption, their order in the direction orthogonal to θ is given in advance, and let us denote them symbolically, in this order, as $1, 2, \ldots, n$.

We can formulate the maintenance of H_θ by the algorithm given in (a) as follows. We are given a system of n horizontal segments $I_k = [(a_k, k), (b_k, k)]$, for $k = 1, \ldots, n$, where $a_k < b_k$ are integers in $\{1, \ldots, 2n\}$. The goal is to compute their lower envelope (pointwise minimum). Indeed, the x-axis in this setup represents the line L, and the intervals $(j, j+1)$, for $j = 1, \ldots, 2n-1$, are the atomic intervals. For each k, the interval I_k represents the k-th polygon edge at orientation θ, where a_k and b_k represent the points at which the boundary of the polygon containing that edge intersects L. See Figure 7 for an illustration. Note that in this formulation, we regard the problem as an *off-line* problem, where the sequence of operations is given in advance, which is indeed the case.

This abstract problem can be solved in linear time, using the disjoint set union algorithm of Gabow and Tarjan [19]. Specifically, let $s_j = (j, j+1)$, for $j = 1, \ldots, 2n-1$. We maintain a partition of $\{s_1, \ldots, s_{2n-1}\}$ into disjoint sets. Initially, the sets are the singletons $\{s_j\}$, for $j = 1, \ldots, 2n-1$. The data structure that we use supports operations of the form $\mathbf{find}(i)$, which returns the set containing the atomic interval s_i, and $\mathbf{union}(i, i+1)$, which merges the sets $\mathbf{find}(i)$ and $\mathbf{find}(i+1)$ into a new common set. In addition, each set A has a pointer to the rightmost atomic interval that it contains, and a flag that indicates whether A is a singleton set. Each singleton set stores an additional pointer to the interval I_k that attains the lower envelope over it. Initially, these pointers are all null.

We process the intervals I_k in increasing y order. To process $I_k = [(a_k, k), (b_k, k)]$, we compute $\sigma = \mathbf{find}(a_k)$. Suppose that the rightmost atomic interval of σ is s_q. If I_k also covers s_{q+1} (that is, if $b_k > q+1$), we perform $\mathbf{union}(q, q+1)$, find the rightmost interval

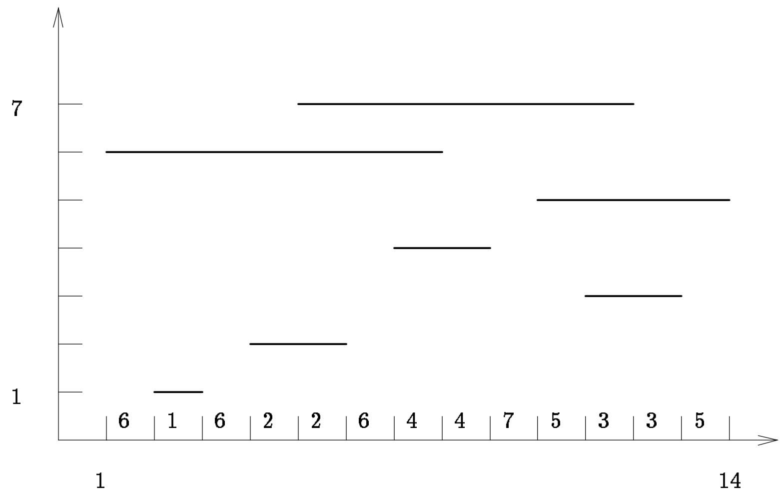

Figure 7: The maintenance of a priority queue H_θ as an abstract disjoint set union problem

that this new set contains, and keep iterating in this manner until all of I_k is exhausted. If any of the sets encountered in this process is a singleton, we record in it that the lower envelope over it is attained by I_k. When we finish processing all the I_k's, each atomic interval s_j stores the interval I_k attaining the lower envelope over it (or, if s_j still stores the initial null value, no interval I_k covers s_j).

Using the data structure of [19], this processing can be done in $O(n)$ time. This completes the proof of part (b).

We remark that the model of computation used in (b) is different from the algebraic decision tree model, since the algorithm of [19] uses the bits of the input data to access various look-up tables. Still, since the input data consists only of integers not larger than $2n$, there is no real explicit dependence on the bits of the original input data. $\Box$

Remark: Lemma 4.1(a) can be extended to higher dimensions. A family $\mathcal{Q}$ of convex polytopes in $\mathbb{R}^d$ is called *c-oriented* if the normal directions of the facets of these polytopes belong to a fixed set of c directions. Let $\mathcal{Q}$ be a family of n c-oriented convex polytopes. For each of the c facet normal directions u, let h_u be the hyperplane orthogonal to u that contains a facet of some polytope $Q \in \mathcal{Q}$, so that no facet with the same normal direction of another polytope lies in the halfspace bounded by h_u and containing Q. Let $\mathcal{H}$ be the resulting collection of c hyperplanes. If $\mathcal{Q}$ is 2-pierceable, then, arguing as in the proof of Lemma 4.1(a), $\mathcal{Q}$ can be pierced by two points, each lying on a hyperplane of $\mathcal{H}$. We can therefore reduce the 2-piercing problem to the following problem: We fix a hyperplane $h \in \mathcal{H}$, and form the collection $\mathcal{Q}_h = \{Q \cap h \mid Q \in \mathcal{Q}\}$. We then want to determine whether there exists $a \in h$ such that the collection $\mathcal{Q}_{h,a}$ of polytopes not containing a is 1-pierceable. This problem can be solved in time $O(n^{d-1})$, by constructing the arrangement $\mathcal{A}(\mathcal{Q}_h)$ within h, and then by traversing its cells, passing from each cell to an adjacent cell, updating the set of polytopes not containing the current cell (by inserting or deleting a single polytope), and testing whether these polytopes have a nonempty intersection, in a manner similar to that used in the 2-dimensional case. We leave it to the reader to fill in the easy details. We do not know if there is a faster algorithm for this problem. For example, can the problem be solved in subquadratic time for $d = 3$ dimensions? Related algorithms for 2-piercing in higher dimensions are given in [27]. Our algorithm is faster than those in [27] when $c \geq 2d - 2$.

4.2 Polygonal 2-center problems

We next consider the 2-center problem, as formulated at the beginning of this section. The case of triangles is easy, and can be solved by a deterministic linear-time algorithm (replacing the LP-type technique mentioned above). This follows from (a) Lemma 4.1(b), which says that we can place one piercing point at a vertex of $\mathcal{A}(\mathcal{L})$, and (b) the fact that each line $L_\theta \in \mathcal{L}$ is determined by a fixed point, which does not change as the scaling factor λ varies. A similar technique applies to the case of trapezoids.

For the general polygonal case, we use the Frederickson-Johnson's matrix searching technique, as in subsection 3.2. This is done as follows. Let S be the given set of n points, and let $\mathcal{P}(\lambda) = \{s + \lambda\tilde{P} \mid s \in S\}$, for any $\lambda > 0$, be the corresponding set of isothetic polygons, as defined in the beginning of this section. First observe, as in the preceding paragraph, that for each edge orientation θ of $\tilde{P}$, the line L_θ for $\mathcal{P}(\lambda)$ passes through a side at orientation θ of a polygon $s_\theta + \lambda\tilde{P}$, for a fixed point $s_\theta \in S$ that does not depend on λ. Next, examining the 2-piercing algorithm for $\mathcal{P}(\lambda)$, as given above, it can be shown that the 2-pierceability of $\mathcal{P}(\lambda)$ can change only at values of λ for which three polygon sides become concurrent, with at least one of these sides lying on one of the lines in $\mathcal{L}$. Indeed, the above remark implies that the order of polygon edges at a fixed orientation θ in the direction orthogonal to θ remains unchanged as λ increases, so, as long as no concurrence of this kind occurs, the algorithm performs exactly the same sequence of operations. This implies that the optimal value of λ must correspond to such a concurrence.

Consider therefore such a concurrence of three sides having three respective orientations θ, θ', θ''.

Lemma 4.4 *Let s, s', s'' be three distinct points in S. Let λ_0 be such that the edges of the three polygons $s + \lambda_0\tilde{P}$, $s' + \lambda_0\tilde{P}$, $s'' + \lambda_0\tilde{P}$, at the respective orientations θ, θ', θ'', are concurrent. Then one can write $\lambda_0 = \lambda_1(s, s') + \lambda_2(s, s'')$ (so that the first term does not depend on s'' and the second term does not depend on s').*

Proof: Let w be the point common to these three edges. The copy $w + \lambda_0 P$ of the original polygon P has three sides, whose orientations are the reflections of θ, θ', θ'', which pass through the points s, s', s'', respectively. Let L, L', L'' be the lines containing these sides, and let $u = L \cap L'$, $v = L \cap L''$. Then uv is proportional to λ, and we can write $uv = us + sv$ (it is easy to see that s lies in uv); see Figure 8. Clearly, us depends only on s and s' (and the orientations θ, θ'), and sv depends only on s and s'' (and the orientations θ, θ''), which readily implies the lemma. □

We next fix three edge orientations θ, θ', θ'', fix the point s_θ as defined above (there are c choices for this point), and construct a matrix M, whose rows correspond to the points $s' \in S$, sorted in increasing order of $\lambda_1(s_\theta, s')$, as defined above, and whose columns correspond to the points $s'' \in S$, sorted in increasing order of $\lambda_2(s_\theta, s'')$. We put $M_{ij} = \lambda_1(s_\theta, s'_i) + \lambda_2(s_\theta, s''_j)$, where s'_i is the point corresponding to the i-th row of M, and s''_j is

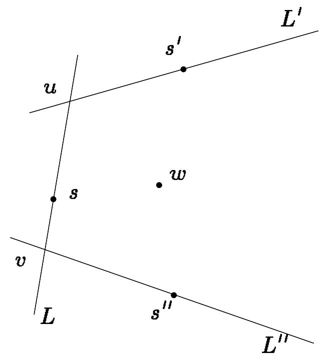

Figure 8: The proof of Lemma 4.4

the point corresponding to the j-th column. If one piercing point lies on L_θ at the optimal λ, and λ corresponds to a concurrency as in Lemma 4.4, then the Frederickson-Johnson technique will find λ, making only $O(\log n)$ 'oracle calls', each of which costs only $O(n)$ time, after an appropriate preprocessing, as in Theorem 4.3(b). Repeating this for each triple of edge orientations, we obtain:

Theorem 4.5 *The polygonal 2-center problem for a set of n points in the plane and a convex c-gon can be solved in $O(c^5 n \log n)$ time. The triangular and trapezoidal 2-center problems can be solved in (deterministic) linear time.*

We remark that more general polygonal 2-center problems can be solved in $O(n \operatorname{polylog} n)$ time, using the parametric searching technique [36], as was also done in Section 3.2, for the general rectilinear 4- and 5-center problems. Here too we omit the details, and leave it to the interested reader to work them out.

5 Discussion

In this paper we have considered several rectilinear and polygonal piercing and center problems in the plane, and have given optimal algorithms for some of those. We have given optimal solutions for the rectilinear p-piercing and p-center problems, for $p \leq 4$. The 3-center problem has been solved by representing it as an LP-type problem, and it might be useful to obtain a more direct solution. For $p = 5$, we have near-linear algorithms for the rectilinear 5-piercing and 5-center problems, and it might be interesting to improve their running time by some polylogarithmic factors. For $p \geq 6$, the exponential dependence on p of the running time of a solution to either of these problems begins to 'show up', and more significant improvements might be possible. For example, can the rectilinear 6-piercing or 6-center problems be solved in subquadratic time? in near-linear time? When p becomes much larger, the interest is in designing algorithms with improved (albeit exponential) dependence of their running time on p, like the current work in [5] mentioned above. Another direction for further research is to obtain efficient approximation algorithms, that either approximate the number of centers up to some small constant factors, or approximate the optimal size of the covering squares, again up to some small constant factor, or both.

Another direction for further research is to design linear-time solutions of the rectilinear 3-center or of the triangular 2-center problems, that would be efficient in practice. Here one could either show that the actual combinatorial dimensions of these problems are actually much smaller than the upper bounds given in this paper, or alternatively find more direct and faster approaches to these problems.

The polygonal p-piercing and p-center problems have been studied here only for the case $p = 2$, so one faces many more open problems here. For example, can the polygonal 3-piercing problem be solved in near-linear time? An $O(n \log n)$-time solution for the triangular 3-piercing problem has recently been noted by Assa, and independently by Katz and Nielsen.

Another collection of open problems is to extend our study to higher dimensions. Some extensions of this sort, to 2-piercing of various objects in higher dimensions, have been obtained by Katz and Nielsen [27]. Assa, and independently Katz and Nielsen, have recently obtained an $O(n \log n)$-time solution for 3-piercing of axis-parallel boxes. Analogous problems exist for c-oriented polyhedra. For example, can 2-piercing of c-oriented polyhedra in 3-space be decided in subquadratic time (cf. the quadratic-time solution given above).

Acknowledgments. We wish to thank Jirka Matoušek, Otfried Schwarzkopf, Günter Rote, Arik Tamir and Eyal Assa for helpful discussions and suggestions on the problems studied in this paper.

References

[1] P.K. Agarwal and M. Sharir, Algorithmic techniques for geometric optimization, in: Lecture Notes in Computer Science, Vol. 1000, Springer-Verlag, 1995, pp. 234–253.

[2] P.K. Agarwal, M. Sharir and S. Toledo, New applications of parametric searching in computational geometry. *J. Algorithms* **17** (1994), 292–318.

[3] N. Amenta, Helly-type theorems and generalized linear programming, *Discrete Comput. Geom.* **12** (1994), 241–261.

[4] N. Amenta, Bounded boxes, Hausdorff distance, and a new proof of an interesting Helly-type theorem, *Proc. 10th ACM Symp. Comput. Geom.* (1994), 340–347.

[5] E. Assa, in preparation.

[6] B. Chazelle and J. Matoušek, On linear-time deterministic algorithms for optimization problems in fixed dimensions, *Proc. 4th SIAM-ACM Symp. Discr. Algo.* (1993), 281–290.

[7] K. Clarkson, Las Vegas algorithms for linear and integer programming, *J. ACM* 42 (1995), 488–499.

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

[9] Z. Drezner, The p-center problem: Heuristics and optimal algorithms, *J. Oper. Res. Soc.* **35** (1984), 741–748.

[10] Z. Drezner, On the rectangular p-center problem, *Naval Res. Logist. Quart.* **34** (1987), 229–234.

[11] J. Eckhoff, Helly, Radon, and Carathéodory type theorems, in *Handbook of Convex Geometry* (P.M. Gruber, J.M. Wills, Eds.) **A** (1993), 389–448.

[12] J. Elzinga and D. W. Hearn, Geometrical solution for some minimax location problems, *Transportation Sci.* **6** (1972), 379–394.

[13] D. Eppstein, Faster construction of planar 2-centers, Tech. Report 96-12, Dept. Comp. Sci., University of California, Irvine, 1996.

[14] R. L. Francis and J. A. White, *Facility Layout and Location*, Prentice Hall, Englewood Cliffs, New Jersey (1974).

[15] G. Frederickson, Optimal algorithms for tree partitioning, *Proc. 2th SIAM-ACM Symp. Discr. Algo.* (1991), 168–177.

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

[17] G. Frederickson and D. Johnson, Finding the k-th shortest paths and p-centers by generating and searching good data structures, *J. Algorithms* 4 (1983), 61–80.

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

[19] H. Gabow and R. Tarjan, A linear time algorithm for a special case of disjoint set union, *J. Comput. Syst. Sci.* **30** (1985), 209–221.

[20] B. Gärtner, A subexponential algorithm for abstract optimization problems, *SIAM J. Comput.* **24** (1995), 1018–1035.

[21] B. Gärtner and E. Welzl, Linear programming — Randomization and abstract frameworks, *Proc. 13th Symp. Theo. Asp. Comp. Sci.*, Lecture Notes in Computer Science **1046**, Springer-verlag, 1996, pp. 669–687.

[22] A. Glozman, K. Kedem, and G. Shpitalnik, On some geometric selection and optimization problems via sorted matrices, *Proc. 4th Workshop Algorithms Data Struct.* (1995), 26–37.

[23] M. Golumbic, *Algorithmic Graph Theory*, Academic Press, New York, 1980.

[24] R.Z. Hwang, R.C.T. Lee and R.C. Chang, The slab dividing approach to solve the Euclidean p-center problem, *Algorithmica* **9** (1993), 1–22.

[25] O. Kariv and S. L. Hakimi, An algorithmic approach to network location problems. I: The p-centers, *SIAM J. Appl. Math.* **37** (1979), 513–538.

[26] M. Katchalski and D. Nashtir, On a conjecture of Danzer and Grünbaum, *Proc. AMS*, to appear.

[27] M. Katz and F. Nielsen, On piercing sets of objects, *Proc. 12th ACM Symp. Comput. Geom.* (1996), pp. 113–121.

[28] M. T. Ko and Y. T. Ching, Linear time algorithms for the weighted tailored 2-partition problem and the weighted 2-center problem under l_∞-distance, *Discrete Applied Math.* **40** (1992), 397–410.

[29] M. T. Ko and R. C. T. Lee, On weighted rectilinear 2-center and 3-center problems, *Inform. Sci.* **54** (1991), 169–190.

[30] M. T. Ko, R. C. T. Lee, and J. S. Chang, Rectilinear m-center problem, *Proc. National Computer Symp.*, Taipei, Taiwan, R.O.C. (1987), 325–329.

[31] M. T. Ko, R. C. T. Lee, and J. S. Chang, On optimal approximation for the rectilinear m-center problem, *Algorithmica* **5** (1990), 341–352.

[32] D.T. Lee and Y.F. Wu, Geometric complexity of some location problems, *Algorithmica* **1** (1986), 193–211.

[33] J. Matoušek, M. Sharir, and E. Welzl, A subexponential bound for linear programming and related problems, *Algorithmica* **16** (1996), 498–516.

[34] N. Megiddo, Linear time algorithms for linear time programming in R^3 and related problems, *SIAM J. Comput.* **12** (1983), 759–776.

[35] N. Megiddo, The weighted Euclidean 1-center problem, *Math. Oper. Res.* **8** (1983), 498–504.

[36] N. Megiddo, Applying parallel computation algorithms in the design of serial algorithms, *J. ACM* **30** (1983), 852–865.

[37] N. Megiddo and A. Tamir, New results on the complexity of p-center problems, *SIAM J. Comput.* **12** (1983), 751–758.

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

[39] M. Overmars and J. van Leeuwen, Maintenance of configurations in the plane, *J. Comp. System Sciences* **23** (1981), 166–204.

[40] M. Sharir, A near-linear algorithm for the planar 2-center problem, *Discrete Comput. Geom.*, in press.

[41] M. Sharir and E. Welzl, A combinatorial bound for linear programming and related problems, *Proc. 9th Symp. Theo. Asp. Comp. Sci.*, Lecture Notes in Computer Science **577**, Springer-verlag, 1992, pp. 569–579.