Computing containment relations between Rectilinear Convex Hulls

Carlos Alegría-Galicia∗ Carlos Seara † Jorge Urrutia‡

Abstract

Let \( P \) be a \( k \)-colored set of \( n \) points in the plane, where \( k \leq n \). After a pre-processing step, we solve queries about the containment relations between the Rectilinear Convex Hulls of any pair of chromatic classes of \( P \), computed after rotating the plane by a specified angle. When \( k = 2 \), we pre-process in optimal \( O(n \log n) \) time and \( O(n) \) space, and solve queries in \( O(\log n) \) time. For \( k > 2 \), the pre-processing complexity is increased to \( O(kn + dn \log n) \) time and \( O(dn) \) space, where \( 1 \leq d \leq k - 1 \). The value of \( d \) depends on the containment relations between the convex hulls of the chromatic classes of \( P \).

1 Introduction

Given two points \( p, q \in \mathbb{R}^2 \) with coordinates \((a, b)\) and \((c, d)\) respectively, we say that \( p \) dominates \( q \) \((p \succ q)\) if and only if \( a \geq c \) and \( b \geq d \). We call \( \succ \) the vector dominance relation between \( p \) and \( q \).

Let \( P \) be a set of \( n \) points in the plane colored using \( k \leq n \) colors into \( k \) chromatic classes. The dominance region of \( P \) is the set \( D(P) = \{ x \in \mathbb{R}^2 : p \succ x, \forall p \in P \} \) of points of the plane dominated by at least one point in \( P \). Let \( D_\theta(P) \) be the dominance region of \( P \) computed after counter clockwise rotating the coordinate axes by an angle \( \theta \). We consider that \( D_0(P) = D(P) \). We say that \( P \) dominates a point (or a set of points) \( \) with respect to the \( i \)th quadrant of the coordinate system, if it is contained in \( D_{(i-1)\pi/2}(P) \). See Figure 1.

![Figure 1: The dominance regions: (a) \( D(P) \) and (b) \( D_{\pi/2}(P) \). The point \( x \) is dominated by \( P \) with respect to the first and the second quadrants of the coordinate system.](image)

**Definition 1 (Ottmann et al. [5])** The Rectilinear Convex Hull \( RH(P) \) of \( P \) (or rectilinear hull of \( P \) for short) is the set of points in the plane dominated by \( P \) with respect to the four quadrants of the coordinate system.

The rectilinear hull of a set of points is depicted in Figure 2 along with three of its fundamental properties. In contrast with the convex hull (with which has been compared to, looking for similarities [5, 6]), the rectilinear hull is orientation-dependent and it is not always connected. Moreover, the rectilinear

---

∗Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, alegria_c@uxmcc2.iimas.unam.mx
†Dept. Matemàtica Aplicada II, Universitat Politècnica de Catalunya, Spain, carlos.seara@upc.edu
‡Instituto de Matemáticas, Universidad Nacional Autónoma de México, urrutia@matem.unam.mx
hull of \( P \) is contained in the region bounded by the convex hull of \( P \), regarding of the orientation of the plane [6].

![Figure 2: A sample rectilinear hull for two different orientations of the plane.](image)

Let \( P_i \) be the \( i \)th chromatic class of \( P \), and \( \mathcal{RH}_\theta(P) \) the rectilinear hull of \( P \) computed after rotating the coordinate axes by an angle \( \theta \). We describe algorithms that, after a pre-processing step, can solve queries about the containment relations between \( \mathcal{RH}_\theta(P_i) \) and \( \mathcal{RH}_\theta(P_j) \), for \( 1 \leq i, j \leq k \) and all \( \theta \in [0, 2\pi) \). When \( k = 2 \), our algorithms pre-process in optimal \( O(n \log n) \) time and \( O(n) \) space, and solve queries in \( O(\log n) \) time. For \( k > 2 \), the pre-processing bounds are increased to \( O(kn + dn \log n) \) time and \( O(dn) \) space, where \( 1 \leq d \leq k - 1 \). The parameter \( d \) denotes the maximum out-degree of the containment digraph [4] of the set of convex hulls of the chromatic classes of \( P \).

We based our work mainly on results from Avis et al. [2] and Díaz-Báñez et al. [3]. As far as it is known by the authors, there are no previous results pertaining containment between rectilinear hulls.

2 The dominance interval table

The algorithms we present in this paper are based on a fundamental tool described in this section. This tool will allow us to compute containment relations regarding of the orientation of the plane, and establishes a lower bound that will be used in Section 4, where we deal with the case \( k = 2 \).

Let \( L_x \) and \( R_x \) be two rays leaving the point \( x \in \mathbb{R}^2 \) and \( \theta \) an angle such that, after counter clockwise rotating \( R_x \) by \( \theta \) around \( x \), we obtain \( L_x \). The polygonal chain \( L_x \cup R_x \) divides the plane in two regions called wedges. We say that both wedges have vertex \( x \) and sizes \( \theta \) and \( 2\pi - \theta \), respectively. A ray leaving a point \( p \in P \) is maximal, if it contains another point in \( P \). A region of the plane is \( P \)-free if it has an empty intersection with \( P \). We say that \( p \) is an un-oriented \( \Theta \)-maximal of \( P \), if there exists two maximal rays from \( p \) bounding a \( P \)-free wedge with size at least \( \Theta \) (Figure 3(a)).

A point \( p \in P \) is a maximal element of \( P \), if there is no \( q \in P \) such that \( q \neq p \) and \( q \succ p \). As the vector dominance relation establishes a partial order in the elements of \( P \), more than one maximal element can be found in \( P \). Un-oriented \( \Theta \)-maximality generalizes the concept of maximal element over vector dominance. Indeed, a point \( p \in P \) is a maximal element of \( P \) if and only if, it is the vertex of an isothetic \( P \)-free wedge with size \( \pi/2 \), otherwise it will be dominated by at least one point (Figure 3(b)). Thus, every maximal element in \( P \) is an un-oriented \( \pi/2 \)-maximal.

**Lemma 1 (Avis et al. [2])** All un-oriented \( \pi/2 \)-maximal elements in \( P \) can be found in optimal \( O(n \log n) \) time and \( O(n) \) space.

The output of the algorithm of Lemma 1 is the list of all un-oriented \( \pi/2 \)-maximal points in \( P \). For every such a point \( p \), the output also contains the maximal rays \( L_p \) and \( R_p \) that bound the corresponding \( P \)-free wedge with vertex \( p \) and size at least \( \pi/2 \). Let \( \theta \)-wedge be a wedge of size \( \theta \). Let \( W_p \) be the wedge bounded by \( L_p \) and \( R_p \), and \( w_p \) be a \( \pi/2 \)-wedge with vertex \( p \). In constant time we can compute the set of orientations in which \( w_p \) fits in \( W_p \). Figure 4(a) shows how such a computation can
be done, by finding the angle between the bisectors of \( w_p \) when it is placed over the bounding rays of \( W_p \). The same computation is possible using any fixed vector inside \( w_p \).

Performing a second constant-time operation, we can transform the previous set of orientations to \( S^1 \). By doing this, we have effectively computed an angular interval \([\theta_1, \theta_2]\) in which \( p \) is the vertex of a \( P \)-free \( \pi/2 \)-wedge. Clearly, \( p \) is a maximal element in \( P \) when the coordinate axes are rotated by any angle \( \theta \in [\theta_1, \theta_2] \). We say that \([\theta_1, \theta_2]\) is a \textit{dominance interval} of \( p \), and that \( \theta_1, \theta_2 \) are respectively, its \textit{in-} and \textit{out-} events. As the size of \( W_p \) is at least \( \pi/2 \), we can compute at most three dominance intervals for every point in \( P \).

The \textit{dominance interval table} is the set of dominance intervals corresponding to all points in \( P \), ordered by its appearance when increasing \( \theta \) from 0 to \( 2\pi \) (Figure 4(b)). Considering Lemma 1 and the fact that there is a linear number of dominance intervals, we obtain the following lemma.

**Lemma 2 (Díaz-Báñez et al. [3])** The dominance interval table can be computed in \( O(n \log n) \) time and \( O(n) \) space.

### 3 The containment relation

Let \( A, B \) denote two sets of points in the plane, and \( \text{CH}(A), \text{CH}(B) \) denote respectively, the convex hulls of \( A \) and \( B \). It is clear that \( \text{CH}(A) \subset \text{CH}(B) \) if and only if, the respective convex polygons satisfy the same contained relation; i.e., the polygon to \( A \) is nested within the polygon to \( B \). To detect this
condition we don’t need to operate on all the elements on \( A \) (or \( B \)), but only in the vertices of its corresponding convex hulls.

The containment relation for rectilinear hulls is not that straightforward, due the fact that the rectilinear hull might be a disconnected region. Nevertheless, it is not hard to see that Definition 1 provides a way to describe this nesting property, in terms the following necessary and sufficient condition.

**Proposition 3** \( \mathcal{RH}(A) \subseteq \mathcal{RH}(B) \) if and only if, \( B \) dominates \( A \) with respect to the four quadrants of the coordinate system (Figure 5).

![Figure 5: \( \mathcal{RH}(A) \) is contained in \( \mathcal{RH}(B) \). Note that, regarding of the configurations of the disconnected components of \( \mathcal{RH}(A) \) and \( \mathcal{RH}(B) \), Proposition 3 holds.](image)

The points of \( A \) over the boundary of \( \mathcal{RH}(A) \) form the set \( \mathcal{M}(A) \) of maximal elements of \( A \), computed after counter clockwise rotating the coordinate axes by \( i \cdot \frac{\pi}{2} \), \( 0 \leq i \leq 3 \). By the transitivity of vector dominance, we can clearly express Proposition 3 as follows.

**Lemma 4** (containment) \( \mathcal{RH}(A) \subseteq \mathcal{RH}(B) \) if and only if, \( \mathcal{M}(B) \) dominates \( \mathcal{M}(A) \) with respect to the four quadrants of the coordinate system.

4 **Containment relations for** \( k = 2 \)

Consider the dominance interval table computed on \( P \). All the intervals intersected by a vertical line on \( \theta = 0 \), correspond to maximal points in \( P \) when no rotation of the plane has been made. Thus, the elements in \( \mathcal{M}(P) \) correspond to the intervals intersected by vertical lines on \( \theta = 0, \pi/2, \pi, \) and \( 3/2\pi \), as shown in Figure 6.

![Figure 6: The intervals intersected by the vertical lines correspond to points in \( \mathcal{M}(P) \).](image)
Remember from Section 1 that \( P_i \) is the \( i \)th chromatic class of \( P \). We then have two disjoint point sets \( P_1 \) and \( P_2 \) such that \( P = P_1 \cup P_2 \). Let \( m_1 \) (\( m_2 \)) be the number of dominance intervals corresponding to points in \( P_1 \) (\( P_2 \)) intersected by the four vertical lines. Note that \( m_1 \) (\( m_2 \)) represents the number of elements of \( M(P_1) \) (\( M(P_2) \)) contained in \( M(P) \). When \( m_2 = 0 \), all the intervals intersected by the vertical lines correspond to points in \( P_1 \) and thus, \( M(P) = M(P_1) \). This means that \( M(P_1) \) dominates \( M(P_2) \) and from Lemma 4, that \( \mathcal{RH}(P_1) \) contains \( \mathcal{RH}(P_2) \). The same is true for the opposite case, when \( m_1 = 0 \).

The basic idea of the algorithm is to maintain the values of \( m_1 \) and \( m_2 \) during a complete rotation of the plane. The rotation is simulated by sweeping the dominance table from 0 to \( \pi/2 \) using the four vertical lines at the same time. All the changes on \( m_1 \) and \( m_2 \) will be organized in a data structure that allow us to efficiently retrieve its values for a particular angle. Using this data structure we will answer queries about the containment relations by verifying the condition on Lemma 4 as we explained before. The algorithm is outlined as follows:

1. Compute the dominance interval table on \( P \).
2. Initialize an ordered list \( L \). The \( i \)th entry in this list will be a tuple of the form \( \langle m_1^i, m_2^i, \alpha^i \rangle \).
   The elements \( m_1^i, m_2^i \) contain the values of \( m_1, m_2 \) that remain unchanged during the interval \([\alpha^i, \alpha^{i+1})\). The entries will be sorted in \( L \) by \( \alpha^i \). We will store \( L \) in a balanced search tree.
3. Compute \( m_1 \) and \( m_2 \) using vertical lines at 0, \( \pi/2 \), \( \pi \), and 3/2\( \pi \). Add the entry \( \langle m_1, m_2, 0 \rangle \) to \( L \).
4. Sweep the dominance table from 0 to \( \pi/2 \) using the four vertical lines at the same time.
   (a) When any of the lines reaches an event \( \alpha \) corresponding to a point in \( P_1 \) (\( P_2 \)), update \( m_1 \) (\( m_2 \)) increasing by 1 if an \( m \)-event, and decreasing by 1 if an \( o \)-event.
   (b) Before resuming the sweep, add the entry \( \langle m_1, m_2, \alpha \rangle \) to \( L \).
5. Answer queries about the containment relations for a given value of \( \theta \in [0, 2\pi) \) as follows. Search in \( L \) the entry \( \langle m_1^0, m_2^0, \alpha^0 \rangle \) such that \( \theta \in [\alpha^0, \alpha^{0+1}) \). Report the result to specify that there is no containment \((m_1^0, m_2^0 > 0)\), \( \mathcal{RH}_\theta(P_1) \) contains \( \mathcal{RH}_\theta(P_2) \) (\( m_2^0 = 0 \)), or viceversa.

From Lemma 2, item 1 requires \( O(n \log n) \) time and \( O(n) \) space. The construction of \( L \) in item 2 consumes constant time. The initialization step on item 3 takes \( O(n) \) time, as any vertical line intersects at most \( n \) intervals for any value of \( \theta \). The line sweep of item 4 requires \( O(n \log n) \) time: there is a linear number of events and we require logarithmic time to process each event (the time to update \( m_1 \) and \( m_2 \), plus the time to add a new entry on \( L \)). In total, the algorithm creates and fills up \( L \) using \( O(n \log n) \) time and \( O(n) \) space. As \( L \) is stored in a balanced search tree, the queries of item 5 are solved in logarithmic time.

By considering Lemma 1 and the previous complexity analysis, we obtain the following theorem.

**Theorem 5** After pre-processing \( P \) in optimal \( O(n \log n) \) time and \( O(n) \) space, queries about the containment relations between \( \mathcal{RH}_\theta(P_1) \) and \( \mathcal{RH}_\theta(P_2) \) for any \( \theta \in [0, 2\pi) \) can be answered in \( O(\log n) \) time.

## 5 Containment relations for \( k > 2 \)

By directly using the results from the previous section, it is easy to come up with an upper bound when dealing with the case \( k > 2 \). In \( O(k^2 n \log n) \) time and \( O(k^2 n) \) space, we can construct a dominance interval table for every one of the \( \binom{k}{2} \) pairs of chromatic classes of \( P \). From this tables we can compute the counting variables for every pair, and answer queries about the containment relations. Note that the query complexity remains the same.

We will lower this brute-force upper bound by attacking the problem from two fronts. First, we will reduce the number of pairs of chromatic classes to compare for inclusion, by using a fundamental property we mentioned in Section 1: \( \mathcal{RH}_\theta(P) \) is contained in \( \mathcal{CH}(P) \) regarding the value of \( \theta \).
Lemma 6 If $\mathcal{R}_\theta(P_i) \subset \mathcal{R}_\theta(P_j)$ for some $\theta \in [0, 2\pi)$, then $\mathcal{C}(P_i) \subset \mathcal{C}(P_j)$.

The containment digraph of the set $\{\mathcal{C}(P_1), \ldots, \mathcal{C}(P_k)\}$ of convex hulls of the chromatic classes of $P$, is an acyclic digraph $G = (V, E)$ with vertex set $V = \{1, \ldots, k\}$ and arc set $E = \{(i, j) : \mathcal{C}(P_i) \subset \mathcal{C}(P_j)\}$. By computing $G$, we will be able to provide an output sensitive algorithm whose complexity is expressed in terms of $k$ and the maximum out-degree $d$ of $G$. In the worst case, $d$ could be equal to $k - 1$ and thus, there might be $O(k^2)$ pairs that need to be compared (see Figure 7).

![Figure 7: A case where $O(k^2)$ pairs need to be compared ($d = k - 1$).](image)

Second, we will modify the algorithm of Lemma 1 to efficiently construct what we call a chromatic dominance interval table. In this data structure, a vertical line in a specific value of $\theta$ intersects at most $d$ dominance intervals for every $p \in P_i$. The $j$th intersected interval denotes that, when the coordinate axes are rotated by $\theta$, $p$ is a maximal element of $P_i$ with respect to $P_j$; that is, there is no $q \in P_j$ such that $q \succ p$. The containment condition established by Lemma 4 can be verified in a similar way as we did for $k = 2$. Let $m_{i,j}$ be the number of elements in $\mathcal{M}(P_i)$ computed with respect to $P_j$. The value of $m_{i,j}$ is the number of the corresponding dominance intervals intersected by the four vertical sweep lines. Again, $\mathcal{R}(P_i)$ contains $\mathcal{R}(P_j)$ if and only if $\mathcal{M}(P_i)$ dominates $\mathcal{M}(P_j)$ with respect to the four quadrants of the coordinate system. In this case, this is equivalent to say that $m_{j,i} = 0$.

5.1 Extending the bichromatic case

Let $h_i$ denote the number of points in $P_i$ over the boundary of $\mathcal{C}(P_i)$. The containment relation between $\mathcal{C}(P_i)$ and $\mathcal{C}(P_j)$ can be obtained by merging both convex hulls in $O(h_i + h_j)$ time and space [8]: compute $\mathcal{C}(P_i \cup P_j)$ and check if all its vertices belong to $P_j$ ($\mathcal{C}(P_i) \subset \mathcal{C}(P_j)$) or viceversa. If we make such verification with every pair of convex hulls, we end up realizing $(h_1 + \cdots + h_k)(k - 1) = O(kn)$ operations. An adjacency-list representation of $G$ can be constructed during this process. After verifying that $\mathcal{C}(P_i) \subset \mathcal{C}(P_j)$, in constant time we add $j$ to the adjacency list of node $i$. Clearly, the adjacency list requires $O(|E|) = O(kd)$ space.

Lemma 7 An adjacency-list representation of the containment digraph of the set $\{\mathcal{C}(P_1), \ldots, \mathcal{C}(P_k)\}$, can be computed in $O(kn)$ time and $O(kd)$ space.

To construct the chromatic dominance interval table, we will modify the restricted un-oriented maximum approach from [2], used in Lemma 1 to compute the $\pi/2$-maximal elements of a point set. Also, we will use $G$ to discriminate the pairs of chromatic classes to be processed.

A ray leaving a point $p \in P_i$ is maximal with respect to $P_j$, if it also contains a point in $P_j$. We say that $p$ is an un-oriented $\Theta$-maximal of $P_i$ with respect to $P_j$, if there exists two maximal rays with respect to $P_j$ leaving $p$ and bounding a $P_j$-free wedge with size at least $\Theta$. Let $M_{i,j}(P)$ be the set of $\pi/2$-maximal elements in $P_i$ with respect to $P_j$. Our goal is to compute $M_{i,j}(P)$ for all $(i, j) \in E$. The algorithm is based on the following simple property.

Observation 1 For each point $p \in M_{i,j}(P)$, the (at most three) $P_j$-free wedges with vertex on $p$ and size at least $\pi/2$, contains one of the $X^+, X^-, Y^+$, and $Y^-$ coordinate semi-axes.
The problem of computing $M_{i,j}(P)$ is then reduced to report, for each one of the four semi-axes, the un-oriented maxima in $P_i$ with respect to $P_j$ whose corresponding wedge contains the semi-axis.

Consider the semi-axis $Y^+$. The following is also to be done with $Y^-, X^-$, and $X^+$. We first sort the elements in $P$ according to the direction orthogonal to $Y^+$ (the $X$ axis). We then perform two linear passes on the points in $P$. In the first pass, we scan the points from left to right, using an on-line algorithm to construct the $k$ convex hulls of the visited points in each chromatic class, one point at a time. Suppose the current element in the scan is a point $p$ that belongs to $P_i$. Before processing $p$, we compute the empty angle $\alpha_j$ between the tangent from $p$ to $\CH(P_j)$ and $Y^+$, for every $j$ adjacent to node $i$ in $G$. In the second pass we perform a similar scan from right to left, computing the empty angles $\beta_j$ at $p$. A simple geometric argument shows that, with respect to the selected semi-axis, $p \in M_{i,j}(P)$ if and only if $\alpha_j + \beta_j \geq \pi/2$ for every $j$ adjacent to node $i$ in $G$. This situation is depicted in Figure 8.

![Figure 8: Computing the chromatic dominance interval table.](image)

After performing the four sweeps on the points in $P$, we can translate the wedge-based output of this algorithm to angular intervals in $S^1$. The dominance table can then be constructed sorting these intervals, in a similar way as we did in Section 4.

We require $O(d \log n)$ time to process a point while performing each sweep, as we consume constant time to compute the corresponding angles, and logarithmic time to insert the point into the convex hull of its respective chromatic class [6]. Thus, to sort the points and perform the sweeps we require $O(dn \log n)$ time. As we now have $O(dn)$ dominance intervals, we require $O(dn \log n)$ time to sort them into the dominance table. Considering the additional time required to compute the convex hulls of the chromatic classes, and the time consumed to compute the containment digraph as reported on Lemma 7, we obtain an overall complexity of $O(kn + n \log n + dn \log n) = O(kn + dn \log n)$ time and $O(n + dk + dn) = O(dn)$ space.

**Lemma 8** The chromatic dominance interval table can be computed using $O(kn + dn \log n)$ time and $O(dn)$ space.

### 5.2 Computing the containment relations

If we consider the algorithm on Section 4, the usage of the chromatic table is straightforward. Instead of a single relation list as the one computed at item 2, we will have $|E|$ lists, one for every pair of chromatic classes sharing an arc in $G$. The $l$th element of the list corresponding to $(i,j) \in E$ is a tuple of the form $(m_{i,j}^l, \alpha^l)$. For any $\theta \in [\alpha^l, \alpha^{l+1})$, we can say if there is no containment relation ($m_{i,j} > 0$) or
\(\mathcal{RH}_\theta(P_i) \subset \mathcal{RH}_\theta(P_j)\) \((m_{i,j} = 0)\). To answer queries proceed as follows. If the specified pair of chromatic classes is such that \((i,j) \notin E\), then there is not containment relation between \(\mathcal{RH}_\theta(P_i)\) and \(\mathcal{RH}_\theta(P_j)\) regarding the value of \(\theta\). Otherwise, search for the tuple \((m_{i,j}^b,\alpha^b)\) such that the given angle belongs to \([\alpha^b,\alpha^{b+1})\). Report the containment relation between the rectilinear hulls according to \(m_{i,j}^b\).

Every vertical line intersects at most \(dn\) intervals on the dominance table, so the computation of the counting variables on item 3 takes \(O(dn)\) time. A new entry is inserted in one of the lists for every of the \(O(dn)\) events, so there can be at most \(O(dn)\) entries in each list and there are \(O(dn)\) entries in total in all the lists. Thus, the sweep on item 4 requires \(O(dn \log n)\) time. From the algorithm explained above, using these lists we can answer queries in \(O(\log(dn)) = O(\log n)\) time. Considering the Lemma 8, we obtain the following theorem.

**Theorem 9** After pre-processing the chromatic classes of \(P\) in \(O(kn + dn \log n)\) time and \(O(dn)\) space, queries about the containment relations between \(\mathcal{RH}_\theta(P_1),\ldots,\mathcal{RH}_\theta(P_k)\) for any \(\theta \in [0, 2\pi]\) can be answered in \(O(\log n)\) time.

### 6 Concluding remarks

We presented algorithms to compute the containment relations between rectilinear hulls of a family of \(k\) disjoint point sets on the plane. After a pre-processing step, we are able to answer queries about the containment of rectilinear hulls, computed after rotating the plane by a specified angle. Our algorithms are time-optimal when \(k = 2\). To our knowledge, these are the first results pertaining the containment of rectilinear hulls.

We based our results in the use of dominance interval tables. The table for a single point set was developed by Diaz-Báñez et al. [3]. We constructed the chromatic version of this data structure extending techniques from Avis et al. [2]. The dominance table was used in [3] to fit a two-joint alternating orthogonal polygonal chain to a point set. In [1] it was used to compute the orientation of the plane, in which the rectilinear hull has minimum area. It was also used in [7] to separate the chromatic classes of a bichromatic point set using two wedges.

Dominance interval tables can be used to solve problems that can be modeled using the number of dominance intervals intersected by sweep lines. When computing single set or bichromatic dominance tables, Theorem 5 says that, as long as the problems are solved in no more than \(O(n \log n)\), the algorithms are time-optimal. Consider for example, for all \(1 \leq i, j \leq k\), the problem of computing the value(s) of \(\theta \in [0, 2\pi]\) in which:

1. \(\mathcal{RH}_\theta(P_i)\) contains the minimum (maximum) number of points in \(P_j\).
2. \(\mathcal{RH}_\theta(P)\) contains the minimum (maximum) number of points of \(P_i\) over its border (or its interior).

The first problem can be solved making a sweep on the chromatic dominance table. Note that, if \(m_{i,j} = |P_i|\), then we have computed an orientation of the plane in which there is a monochromatic rectilinear hull in the bichromatic point set \(P_i \cup P_j\). By slightly modifying the algorithm on Section 5 we can compute the values of \(m_{i,j}\), and we could also optimize the number of vertex in the monochromatic rectilinear hull. The second problem can be solved building a single-set dominance table on \(P\), as we did in Section 4. We then track the number \(m_i\) of intersected dominance intervals corresponding to elements in \(P_i\), and report the number of points of \(P_i\) over the border of \(\mathcal{RH}_\theta(P)\) \((m_i)\), or in its interior \((|P_i| - m_i)\).

Finally, note that we can use the algorithms we presented to report containment relations for angular intervals. Using a threaded binary tree [6] to store the list(s) containing the counting variables, we can report in linear time successive containment relations starting from a given angle.

**Acknowledgments.** Jorge Urrutia was partially supported by projects MTM2006-03909 (Spain), and SEPCONACYT of México, Proyect 80268. Carlos Seara was partially supported by the ESF EUROCORES programme EuroGIGA-ComPoSe IP04-MICINN Project EUI-EURC-2011-4306.
References


