Separating Objects in the Plane by Wedges and Strips

Ferran Hurtado [a] Marc Noy [a] Pedro A. Ramos [b] Carlos Seara [a,2]

[a] *Departament de Matemàtica Aplicada II, Universitat Politècnica de Catalunya, Pau Gargallo 5, 08028 Barcelona, Spain*

[b] *Departamento de Matemáticas, Universidad de Alcalá, Aptdo. de Correos 20, 28871 Alcalá de Henares, Madrid, Spain*

Abstract

In this paper we study the separability of two disjoint sets of objects in the plane according to two criteria: *wedge separability* and *strip separability.* We give algorithms for computing all the separating wedges and strips, the wedges with the maximum and minimum angle, and the narrowest and the widest strip. The objects we consider are points, segments, polygons and circles. As applications, we improve the computation of all the largest circles separating two sets of line segments by a $\log n$ factor, and we generalize the algorithm for computing the minimum polygonal separator of two sets of points to two sets of line segments with the same running time.

Key words: Red-blue separability, wedges, strips, circular and polygonal separability.

1 Introduction

Let P and Q be two disjoint sets of objects in the plane classified as *blue* and *red* objects, respectively. The objects we consider are either points, segments, polygons or circles. If the objects are polygons, n and m represent the total number of segments of the polygons in P and in Q. In other cases n and

[1] Partially supported by Proyectos DGES-MEC PB96-0005-C02-01, DGES-MEC PB96-0005-C02-02, DGES-MEC PB98-0713-C02-01, DGES-MEC PB98-0933. Partially supported by Joint Commission USA-Spain for Scientific and Technological Cooperation Project 98191.

[2] Corresponding author. E-mail: seara@ma2.upc.es

m are the number of objects in P and Q respectively, and in any case $N = \max\{n, m\}$.

Let $\mathcal{C}$ be a family of curves in the plane. The sets P and Q are $\mathcal{C}$ *separable* if there exists a curve $S \in \mathcal{C}$ such that every connected component of $\mathbb{R}^2 - S$ contains objects only from P or from Q. If S is a straight line, the sets P and Q are *line separable*. P and Q are line separable if, and only if, their convex hulls do not intersect [16]. The decision problem of linear separability for any of the above object classes can be solved in $O(N)$ time [7,10]. The region of the plane formed by the points of the separating lines can be computed in $O(N \log N)$ time.

There are other separability criteria. For example, in [4] the authors consider the problem of finding the minimum (in the number of edges) convex polygon separating two point sets in the plane. In image processing, efficient algorithms to find circular separators for two sets of points can be used to recognize disks [1,5,11]. In this paper we present two criteria of separability: the *wedge separability* and the *strip separability*.

Definition 1 *Two disjoint object sets P and Q are wedge separable if there exists a wedge that contains only all the objects of one of the sets (Figure 1a).*

The vertex of the wedge is the common extreme of the half lines. If the angle of the wedge is exactly π, we have the linear separability. Given P and Q, we study the problem of deciding whether they are wedge separable computing the region of the plane formed by the vertices of separating wedges and additionally, the wedges with the maximum and minimum angle.

Definition 2 *Two disjoint object sets P and Q are strip separable if there exist two parallel straight lines (a strip) that contain only all the objects of one of the sets in between (Figure 1b).*

If P and Q are strip separable then they are also wedge separable: just move a little bit one of the parallel lines. Given P and Q, we study the problem of deciding whether they are strip separable computing the set of intervals of strip slopes and additionally, the narrowest and the widest strip.

In this paper we show algorithms that solve the wedge and strip separability problems in $O(N \log N)$ time. In Section 2 we study the wedge separability problem for different kinds of objects. As applications, we improve a result in [2] for the computation of all the largest circles separating two sets of line segments by a $\log n$ factor, and we generalize the algorithm in [4] for computing the minimum polygonal separator of two sets of points to two sets of line segments with the same $O(n \log n)$ running time. In Section 3 we study the strip separability problem. The objects are classified as *red* and *blue* objects, a common terminology used in many problems.

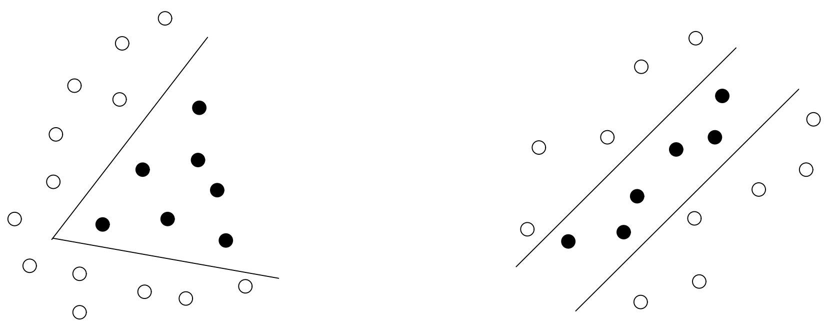


Figure 1. a) Wedge separability, b) strip separability.

2 Wedge separability

2.1 Separating points by wedges

Let $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, q_m\}$ be two disjoint point sets in the plane. Suppose that $n, m \geq 3$ and both P and Q have at least three non collinear points. If the convex hulls of P and Q, $CH(P)$ and $CH(Q)$, do not intersect, the point sets are line separable and therefore wedge separable. A necessary condition for the existence of wedge separability is that the convex hull of one set has to be monochromatic, i.e. it contains only points of one color. In $O(N \log N)$ time we compute the convex hull of each set and determine if any of them is monochromatic, otherwise they are not wedge separable. Suppose that the sets are not line separable and $CH(Q)$ is monochromatic. The problem is to compute the region of the plane formed by vertices of separating wedges.

Tracing the supporting lines from $p_i \in P$ to $CH(Q)$ we obtain two regions A_i and B_i (Figure 2) that do not contain vertices of separating wedges, otherwise the wedges would contain the point p_i. Reciprocally, for any point p in $\overline{A} \cap \overline{B}$, where $A = \cup_{i=1}^{n} A_i$ and $B = \cup_{i=1}^{n} B_i$, the wedge formed by the supporting lines from p to $CH(Q)$ is a separating wedge. Next, we describe how to compute $\overline{A} \cap \overline{B}$.

1) Computing $\overline{A}$

The region $\overline{A}$, as we will see, is a (possibly unbounded) star-shaped polygon whose kernel contains $CH(Q)$. The boundary of $\overline{A}$ is formed by a set of polygonals, $\mathcal{POL}$, around $CH(Q)$. In [4] the same idea is used to obtain a minimum (in the number of edges) convex polygon separating two point sets but our goal is to obtain the vertices of all the separating wedges. We will describe

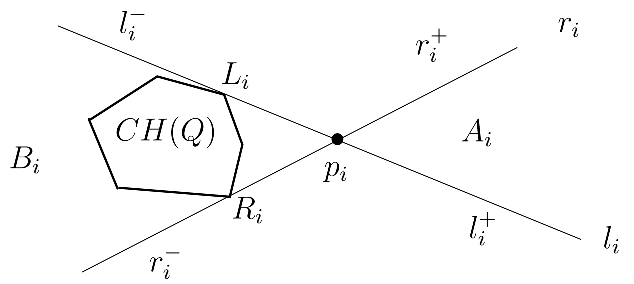


Figure 2. Regions A_i and B_i.

how to obtain $\mathcal{POL}$ because it is relevant for the rest of the paper. First, we eliminate unnecessary points of P. Consider the following *dominance relation with respect to* $CH(Q)$.

Definition 3 *p_i dominates p_j with respect to $CH(Q)$ ($p_j \prec_Q p_i$) if $A_i \subset A_j$.*

The relation $\prec_Q$ is a partial order in P. A point p_j is minimal with respect to $CH(Q)$ if there exists no i, $i \neq j$, such that $A_j \subset A_i$. It is clear that in the computation of A we only need the minimal points, the rest are not necessary. The dominance relation is a generalization of the *vectorial dominance relation* [8]. The problem of the computation of minimal points can be formulated in the language of partial orders [12]. First of all, we introduce the notation and we order the points of P with respect to $CH(Q)$.

Notation: Let l_i (r_i) be the supporting line from p_i to $CH(Q)$ such that $CH(Q)$ is on the left (right) side going from p_i to $CH(Q)$. We compute the supporting lines, their angles and the tangency points L_i (R_i) of l_i (r_i) with $CH(Q)$. By l_i^+ or l_i^- we denote the half line given by l_i and the tangency point L_i depending on whether it contains or not the point p_i. Analogously for r_i^+ or r_i^- (Figure 2).

Ordering points: Let the X axis be the line l_1 and let L_1 be the first point of $CH(Q)$. We order the points of P according to the order of the half lines l_i^+, which are ordered according to: first, the order of L_i in $CH(Q)$ and secondly, by decreasing angle. If there are several points of P sharing a half line l_i^+, the order is given by the proximity to L_i. Relabeling p_i and r_i^+ we have the ordering $< (l_1^+, p_1, r_1^+), (l_2^+, p_2, r_2^+), \ldots, (l_n^+, p_n, r_n^+) >$, that corresponds to a clockwise sweep line around $CH(Q)$. Let $L = \{p_1, p_2, \ldots, p_n\}$ be the ordered points of P. This process can be done in $O(n \log N)$ time.

Minimal points: A point p_i is not a minimal point if it dominates some point p_j. The procedure MINIMAL analyzes the points of P and deletes the non minimal points. Deciding if $p_j \prec_Q p_i$ requires constant time, the number of times the procedure analyzes a point is constant, so the running time of the procedure is $O(n)$. Note that the last points can dominate the first one.

MINIMAL

```
Input: list L (list of the ordered points of P)
Output: list MIN of minimal points
MIN := L, i := 1, j := 2
while j ≤ n do
   if p_j ≺_Q p_i then find the first p_k in MIN such that p_j ≺_Q p_k, k ≤ i;
                       delete {p_k, ..., p_i} of MIN
   i := j, j := j + 1
if p_1 ≺_Q p_i then find the first p_k de MIN such that p_1 ≺_Q p_k, k ≤ i;
                    delete {p_k, ..., p_i} of MIN.
```

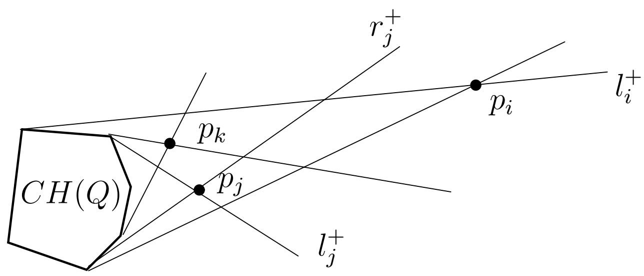


Figure 3. Minimal points.

Lemma 4 *The procedure* MINIMAL *obtains only all the minimal points of* P *with respect to* $CH(Q)$.

PROOF. The procedure only deletes points that dominate other points. Suppose we have the closed chain obtained by the procedure. If a point p_i is minimal it does not dominate any other point and therefore the procedure does not delete it. If p_i is not minimal then there exists a minimal point p_j such that $p_j \prec_Q p_i$ with $i < j$. Note that $p_j \prec_Q p_i$ if p_j follows p_i in clockwise order of l^+ half lines and precedes p_i in counterclockwise order of r^+ lines. A list of minimal points $p_1, p_2, \ldots, p_k$, ordered clockwisely according to l^+ half lines, is also ordered counterclockwisely according to r^+ half lines $p_1, p_2, \ldots, p_k$. Suppose the procedure p_j does not eliminate p_i, then there exist a point p_k of the chain such that $i < k < j$ and $p_j \not\prec_Q p_k$. Therefore, p_k belongs to the region delimited by l_i^+, r_j^+ and l_j^+ (Figure 3). Reasoning similarly, p_{i+1} belongs to this region and p_{i+1} will eliminate p_i. $\square$

Polygonals: By $l_i^+ p_i$ and $r_i^+ p_i$ we denote the half lines l_i^+ and r_i^+ from p_i. By t_{ij} we denote the intersection point of the half lines $l_i^+ p_i$ and $r_j^+ p_j$, and by $p_i t_{ij}$ or $t_{ij} p_j$ we denote the segments determined by their endpoints. The procedure POLYGONALS on MIN outputs $\mathcal{POL}$, which has at most $2n$ segments and half lines and it is the boundary between A and $\overline{A}$. The running time of the procedure is $O(n)$.

POLYGONALS
Input: MIN (list of minimal points)
Output: $\mathcal{POL}$ (list of segments and half lines)
$\mathcal{POL}$:=$\{r_1^+ p_1\}$, $i := 1$, $j := 2$
while $j \leq |MIN|$ **do**
    if $\exists\, t_{ij}$ **then** add $\{p_i t_{ij}, t_{ij} p_j\}$ to $\mathcal{POL}$, **else** add $\{l_i^+ p_i, r_j^+ p_j\}$ to $\mathcal{POL}$
    $i := j$, $j := j + 1$
if $\exists\, t_{i1}$ **then** delete $\{r_1^+ p_1\}$ of $\mathcal{POL}$, add $\{p_i t_{i1}, t_{i1} p_1\}$ to $\mathcal{POL}$,
        else add $\{l_i^+ p_i\}$ to $\mathcal{POL}$.

2) Computing $\overline{B}$

The region B_i is delimited by the half lines l_i^- and r_i^- and part of $CH(Q)$. If $p_i \prec_Q p_j$ then $B_j \subset B_i$, and again we only need the minimal points. Let p_i, p_{i+1} be consecutive points from MIM. We have the following cases:

i) p_i and p_{i+1} are linked by $\mathcal{POL}$. The region $B_i \cup B_{i+1}$ is delimited by the half line r_i^-, the segments in $CH(Q)$ between R_i and L_{i+1}, and the half line l_{i+1}^-, (Figure 4a).

ii) p_i and p_{i+1} are the last point and the first point of two consecutive polygonals of $\mathcal{POL}$. In this case we have two different situations:

(1) the half lines l_i^- and r_{i+1}^- are parallel and the region $B_i \cup B_{i+1}$ is as in i);
(2) the half lines l_i^- and r_{i+1}^- intersect at a point denoted by p_i', the region $B_i \cup B_{i+1}$ is as in i), but the wedge $\{l_i^- p_i', r_{i+1}^- p_i'\}$ belongs to $\overline{B}$, as it is showed in Figure 4b.

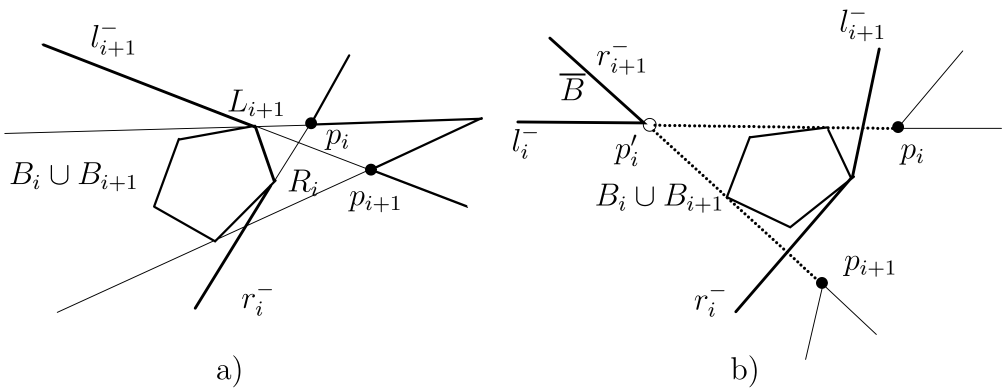


Figure 4. a) Regions of B, b) regions of $\overline{B}$.

Note that if P and Q are not line separable then $\overline{B}$ is formed only by disjoint wedge regions. Using $\mathcal{POL}$ in $O(n)$ time we can obtain the at most n disjoint wedges of $\overline{B}$. Let L' be the list of ordered points p_i' corresponding to the wedge regions $\{l_i^- p'_i, r_{i+1}^- p'_i\}$ of $\overline{B}$. We denote by $l_i'^+$ the half line r_{i+1}^- and by $r_i'^+$ the half line l_i^-.

3) Computing $\overline{A} \cap \overline{B}$

First, note that a wedge of $\overline{B}$ can be totally included in the region A, in such case the wedge is *useless*. This situation happens if the point p'_i defining a wedge of $\overline{B}$ dominates a point p_k of MIN.

Useless wedges: In $O(n \log n)$ time we order $MIN \cup L'$ as we did for the computation of minimal points. The procedure USELESS WEDGES deletes the points of L' dominated by some point of MIN in $O(n)$ time.

USELESS WEDGES
Input: list of ordered points of $MIN \cup L'$
Output: L' without points dominated by points of MIN
while there are points p'_i of L' **do**
  find the first p_k of MIN following p'_i, $(i < k)$,
  if $p_k \prec_Q p'_i$ **then** delete p'_i from L'.

Regions of $\overline{A} \cap \overline{B}$: The regions of $\overline{A} \cap \overline{B}$, denoted by F_i, are the intersections between the useful wedges of $\overline{B}$ and the (possibly unbounded) star-shaped polygon $\overline{A}$. Since a point p'_i of L' is within $\overline{A}$, the region F_i is delimited by the half lines of the useful wedge and the part of $\mathcal{POL}$ between the half lines. According to the order of $MIN \cup L'$, the half lines $l_i'^+$ and $r_i'^+$ of a useful wedge of $\overline{B}$ intersect $\mathcal{POL}$ in at most one point since the half line $l_i'^+$ is between some l_j^+ and l_{j+1}^+ (analogously for $r_i'^+$), or it contains some segment or half line of $\mathcal{POL}$. Therefore, the number of intersections between $\mathcal{POL}$ and the half lines of wedges of $\overline{B}$ is less than or equal to $2n$. The intersections can be computed in $O(N)$ time using the order of $MIN \cup L'$ and the order of $\mathcal{POL}$.

We say that the region F_i is a *fan* since it is still a (possibly unbounded) star-shaped polygon whose kernel contains the point p'_i, *the apex of the fan*, and the angle of the useful wedge at p'_i is less than π (Figure 5). The vertices of a fan are reflex vertices if they are points of P, otherwise they are convex vertices, *the tips of the fan.* The procedure REGIONS outputs the fans of $\overline{A} \cap \overline{B}$ in $O(n)$ time. The number of segments and half lines of the fans is at most $4n$.

REGIONS
Input: L', $\mathcal{POL}$
Output: segments and half lines of $\overline{A} \cap \overline{B}$
while there are points p'_i of L' **do**
  compute F_i: region delimited by the wedge $\{l_i'^+ p'_i, r_i'^+ p'_i\}$ and the part
    of $\mathcal{POL}$ between $l_i'^+$ and $r_i'^+$.

From all the above discussion we have the following theorem.

Theorem 5 *Let P and Q be two disjoint sets of n and m points in the plane respectively. To decide whether they are wedge separable and to compute the region of vertices of separating wedges can be done in $O(N \log N)$ time.*

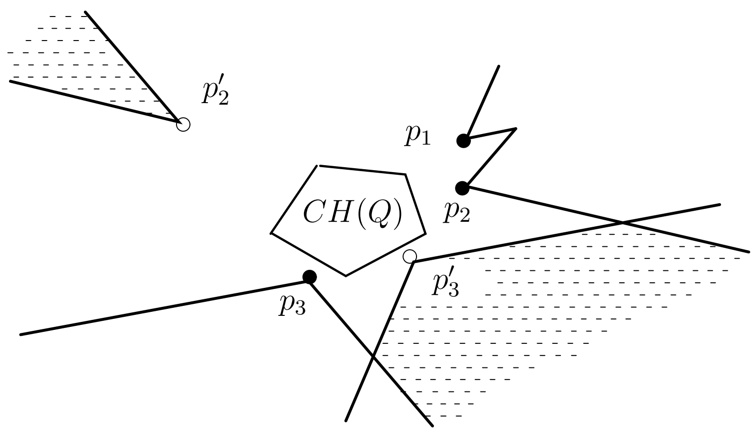


Figure 5. Regions of vertices of separating wedges.

Wedges with maximum and minimum angle

Suppose we have computed the fans of vertices of separating wedges and we want to determine the wedges with maximum and minimum angle.

Lemma 6 *In a fan F_i, the vertex of the separating wedge with maximum angle is the apex of the fan. If the fan is bounded, the vertex of the separating wedge with minimum angle is one of the convex vertices,* the tips of the fan. *If the fan is unbounded the minimum angle is zero.*

PROOF. Let p be any point of the fan F_i with apex p'_i. Note that the smallest separating wedge with vertex p is formed by the supporting lines from p to $CH(Q)$ and the largest separating wedge with vertex in p is formed by the lines pp_j and pp_k, where p_j and p_k are the two points of P which give rise to the useful wedge with apex p'_i.

The supporting lines from p to $CH(Q)$ intersect the half lines l'^{+}_i and r'^{+}_i at points u_1 and u_2, respectively (Figure 6). If the vertex of a separating wedge moves from p to u_1 and from u_1 to p'_i, the angle of the separating wedge increases continuously; analogously when the vertex moves from p to u_2 and from u_2 to p'_i. Therefore, the vertex of the separating wedge with maximum angle is p'_i, the apex of the fan.

If the fan is bounded, the supporting lines from p intersect $\mathcal{POL}$ at points v_1 and v_2. Moving the vertex of the separating wedge from p to v_1 and from v_1 through the segment of the $\mathcal{POL}$ that contains v_1 to the right endpoint of the segment, the angle of the separating wedge decreases continuously. Similarly, when the vertex moves from p to v_2 and from v_2 to the left endpoint of the segment that contains v_2. Therefore, the vertex of the separating wedge with the minimum angle is in some tip of the fan. If the fan is unbounded, the minimum angle is zero for a separating wedge with the vertex at infinity. The total number of tips of fans is at most n. Doing a tour in $\mathcal{POL}$ we determine the wedge with minimum angle in $O(n)$ time. $\square$

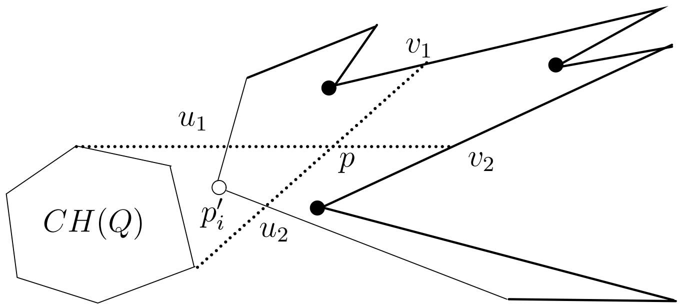


Figure 6. Maximum and minimum angle.

Proposition 7 *Let P and Q be two disjoint wedge separable sets of n and m points in the plane, respectively. If we have pre-computed the region of vertices of separating wedges, the wedges with maximum and minimum angle can be found in $O(N)$ time.*

2.2 Separating segments by wedges

Let $P = \{p_1p'_1, p_2p'_2, \ldots, p_np'_n\}$ and $Q = \{q_1q'_1, q_2q'_2, \ldots, q_mq'_m\}$ be two disjoint sets of segments. In $O(N \log N)$ time we compute the convex hull of each set of segments, determine if any of them is monochromatic, i.e. it does not intersect segments or does not contain endpoints of segments of the other set; otherwise they are not wedge separable.

Suppose that the sets are not line separable and that $CH(Q)$ is monochromatic. We want to compute the region of vertices of wedges separating Q from P. First, we consider the restriction produced by a segment $s_i = p_ip'_i$ of P. The supporting lines between s_i and $CH(Q)$ define the regions A_i and B_i that do not contain vertices of separating wedges, otherwise the wedge contains totally or partially the segment s_i (Figure 7a). The region of vertices of separating wedges is $\overline{A} \cap \overline{B}$, where $A = \cup A_i$ and $B = \cup B_i$.

1) **Computing $\overline{A}$**

The region $\overline{A}$, as we are going to see, is again a (possibly unbounded) star-shaped polygon whose kernel contains $CH(Q)$. The boundary of $\overline{A}$ is formed by a set of polygonals, *the final polygonals* or $\mathcal{POL}$ for short, around $CH(Q)$. We will describe how to obtain the final polygonals.

Ordering segments: We order the segments of P with two orders with respect to $CH(Q)$: clockwise and counterclockwise. For the clockwise order we compute the lines supporting $CH(Q)$ from all the endpoints of segments of P. Let l_i^+ ($l_i'^+$) be the half line tangent to $CH(Q)$ going through p_i (p_i') such that $CH(Q)$ is on the left side going from p_i (p_i') to $CH(Q)$ and starting at the tangency point (Figure 7b). We order the half lines in an angular clockwise way and we order the segments according to the first endpoint that appears in the order given by the half lines. If there are several segments with the first endpoints sharing the same half line, the order is given by the proximity of the endpoint to the tangency point of the half line. If a segment is included in the A region of one of its endpoints it can be replaced by this endpoint alone. If some of these segments have the same first endpoint, the order is given by the order of the second endpoint. Once we have ordered the segments we denote by p_i the first endpoint of the segment and by p_i' the second one. Let $s_1 = p_1p_1'$ be the first segment of P in this order. If a segment $s_i = p_ip_i'$ intersects l_1^+ at a point p_i'' we divide it in two segments p_ip_i'' and $p_i''p_i'$ and delete the segment s_i. The number of segments increases in at most $n-1$. Reordering the segments with the new endpoints according to the criteria above, we are sure that doing a clockwise angular sweep starting at l_1^+ we find all the segments after just one round.

For the counterclockwise ordering we proceed in a similar way using the order of r_i^+ ($r_i'^+$), the half line tangent to $CH(Q)$ going through p_i (p_i') such that $CH(Q)$ is on the right side going from p_i (p_i') to $CH(Q)$ and starting at the tangency point (Figure 7c). We can take s_1 as the first segment of P and use $r_1'^+$ as the starting half line and proceed as above. The ordering process requires $O(n \log N)$ time.

Final polygonals: From a point of a polygonal of $\mathcal{POL}$ we *see* $CH(Q)$ without any obstacle, i.e. the region determined by the half lines tangent to $CH(Q)$ from that point and the part of $CH(Q)$ between the tangency points does not contain or intersect segments of P. Starting with s_1, each segment has an angular projection on the interval $[0, 2\pi)$ sweeping clockwise (counterclockwise) with half lines tangent to $CH(Q)$: each segment s_i has a clockwise angular interval α_i (Figure 7b) and a counterclockwise angular interval β_i (Figure 7c). Then α_i (β_i) is the clockwise (counterclockwise) angle defined by the half lines l_i^+ and $l_i'^+$ ($r_i'^+$ and r_i^+). With the above ordering, we are sure that all the angular intervals are within $[0, 2\pi)$. Now we describe how to compute $\mathcal{POL}$:

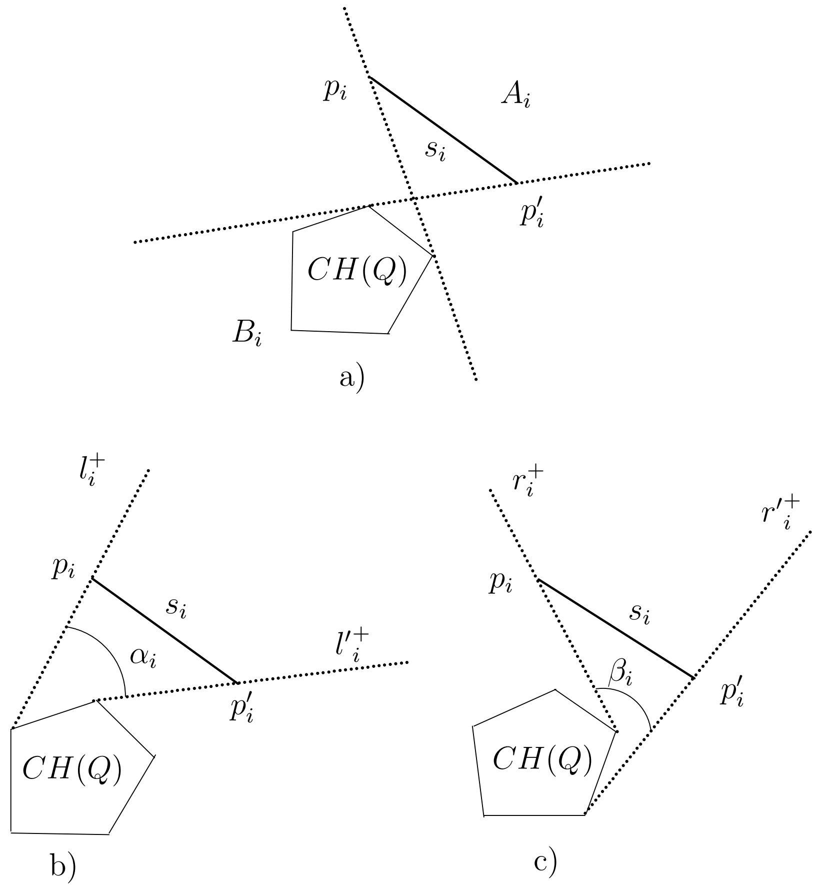


Figure 7. a) Restriction of a segment, b) clockwise angular interval, c) counterclockwise angular interval.

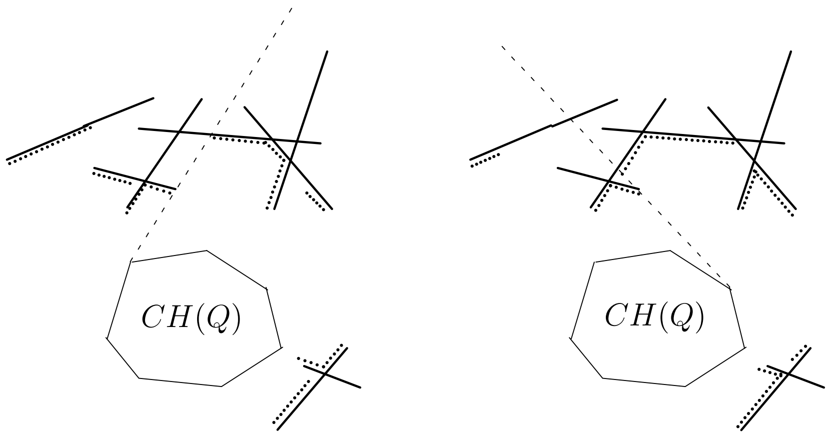
1) compute the set of polygonals of segments that can be seen from $CH(Q)$ (without obstacles) doing a clockwise rotating sweep with a half line tangent to $CH(Q)$, such that $CH(Q)$ is on the left. We call this set *the clockwise polygonals* (Figure 8a);

2) compute the set of polygonals of segments that can be seen from $CH(Q)$ (without obstacles) doing a counterclockwise rotating sweep with a half line tangent to $CH(Q)$, such that $CH(Q)$ is on the right. We call this set *the counterclockwise polygonals* (Figure 8b);

3) compute the set of polygonals corresponding to the endpoints of segments of P. We call this set *the extreme polygonals* (Figure 9a). It can be computed as the polygonals constructed in the case of points;

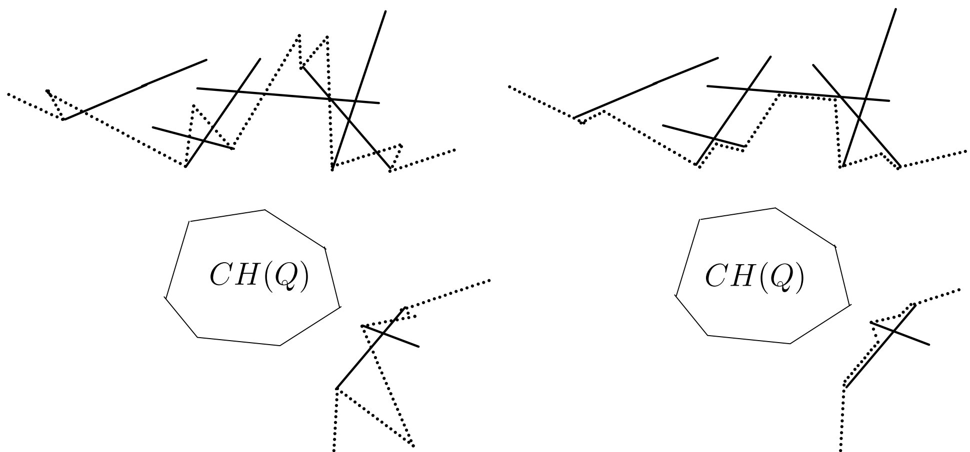
4) from the three sets of polygonals compute the final polygonals or $\mathcal{POL}$ (Figure 9b).

The clockwise polygonals are a variation of the *lower envelope of a set of* *segments* , we call it *clockwise angular lower envelope*. We modify the Hershberger algorithm for the lower envelope of a set of x-monotone Jordan arcs with at most s intersections between any pair of arcs [6,15] in the following way: for each segment point p we define the x-coordinate as the angle of the line tangent to $CH(Q)$ from p that has $CH(Q)$ on the left as going from p to $CH(Q)$, and we define $f(x)$ as the geodesic distance between p and the tangency point of the supporting line through p_1, the first endpoint of the first segment. The segments so defined are x-monotone and with at most one intersection between any pair of them. If there are segments with two common points, *partially coincident segments*, in $O(n \log n)$ time we can construct an equivalent set of segments without partially coincident segments. Therefore, the clockwise polygonals can be computed in $O(n \log N)$ time.



a) b)

Figure 8. a) Clockwise polygonals, b) counterclockwise polygonals.



a) b)

Figure 9. a) Extreme polygonals, b) final polygonals.

In the same way, we compute the counterclockwise polygonals starting from the endpoint p'_1 of s_1 and using the counterclockwise ordering. The extreme polygonals can be computed in $O(n)$ time using the order of the segment endpoints as we did for point sets. The extreme polygonals are formed by at most $4n$ segments and half lines.

For the construction of the polygonals of $\mathcal{POL}$, we merge the above three sets of polygonals taking into account that the number of transition points of the clockwise and counterclockwise polygonals is $O(n\alpha(n))$, where $\alpha(n)$ is the inverse of the Ackermann function. Making a clockwise rotating angular sweeping advancing in discrete steps over the angular intervals determined by the $O(n\alpha(n))$ transition points of the three sets of polygonals, we can determine in constant time which of the three polygonals we take in each angular interval between transition points. The total time needed in the construction of $\mathcal{POL}$ is $O(n \log N) + O(n\alpha(n)) = O(n \log N)$. Note that the number of segments and half lines of $\mathcal{POL}$ is bounded by $O(n\alpha(n))$, and by construction, $\mathcal{POL}$ is clockwise and counterclockwise angularly monotone.

2) **Computing** $\overline{B}$

Two consecutive polygonals of $\mathcal{POL}$ give a wedge region of $\overline{B}$ defined by the half lines l_i^- from the end of the first polygonal and r_{i+1}^- from the beginning of the second polygonal if l_i^- and r_{i+1}^- intersect at a point denoted by p_i'''. Looking up $\mathcal{POL}$ we determine all the wedge regions of $\overline{B}$ in $O(n\alpha(n))$ time. The number of these regions is at most n.

3) **Computing** $\overline{A} \cap \overline{B}$

A wedge region of $\overline{B}$ is *useless* if the point p_i''' is within A, since the wedge region has no intersection with $\overline{A}$. To eliminate all the useless wedge regions of $\overline{B}$ we analyze the points p_i''' checking if they lie in A or in $\overline{A}$. After that, we compute the part of $\mathcal{POL}$ between the half lines that define the useful wedge regions. By the angular monotonicity, each of these half lines intersects $\mathcal{POL}$ in at most one point (or it contains a segment or a half line of $\mathcal{POL}$). In $O(n \log N)$ time we compute the regions of $\overline{A} \cap \overline{B}$. These regions are (possibly unbounded) star-shaped polygons or fans whose kernel contains the point p_i'''. As a consequence of all the above discussion we have the following theorem.

Theorem 8 *Let P and Q be two disjoint sets of n and m segments in the plane respectively. To decide whether they are wedge separable and to compute the region of vertices of separating wedges can be done in $O(N \log N)$ time.*

The vertex of the separating wedge with maximum angle is in the apex of a fan region of $\overline{A} \cap \overline{B}$. It can be determined computing in constant time the angle of the wedge with vertex in p_i''' for each of the at most n fans. If there is an unbounded fan, the minimum angle is zero. Otherwise, the wedge with minimum angle is in the convex extremes of the fan, *the tips*. An upper bound of the total number of tips is $O(n\alpha(n))$ and for each tip we compute its angle in constant time (Figure 10). Hence we have the following result.

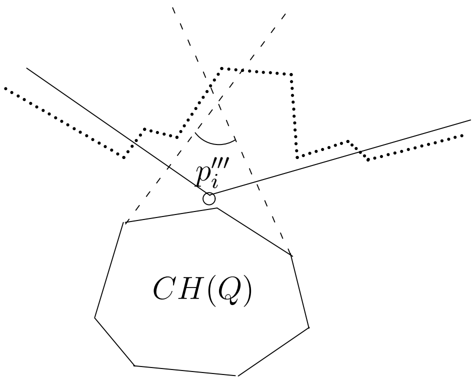


Figure 10. Wedges with maximum and minimum angle.

Proposition 9 *Let P and Q be two disjoint wedge separable sets of n and m segments in the plane respectively. If we have pre-computed the regions of vertices of separating wedges, the wedges with maximum and minimum angle can be found in $O(N\alpha(N))$ time.*

The wedge separability of polygons depends on the total number of segments of the polygons. Let P and Q be two sets of polygons with n and m total segments, respectively. In time $O(n \log m)$ we can check if all the polygons of P are exterior to $CH(Q)$, i.e. if there are no segments of polygons of P intersecting $CH(Q)$ and the polygons of P are not contained in $CH(Q)$. Otherwise Q is not wedge separable from P. If it is so, consider the set of segments of polygons of P and run the above segment algorithm.

Corollary 10 *Let P and Q be two disjoint sets of polygons in the plane with n and m total segments respectively. To decide whether they are wedge separable and to compute the regions of vertices of separating wedges and the wedges with maximum and minimum angle can be done in $O(N \log N)$ time.*

Let P and Q be two disjoint sets of circles with n and m circles classified as blue and red circles, respectively. In $O(N \log m)$ time we compute $CH(Q)$, formed by at most $2m-1$ circular arcs and $2m-1$ segments [3,13], and we check that the circles of P do not intersect or contain $CH(Q)$ and that all of them are exterior to $CH(Q)$. In this case, in $O(n \log N)$ time we compute the interior tangents between each circle of P and $CH(Q)$, delete possible circle duplicities of P, and determine the circular arc of each circle of P that can take part in the computation of the vertices of separating wedges. The interior tangents between each circular arc s_i and $CH(Q)$ define two regions A_i and B_i that do not contain vertices of separating wedges, otherwise the wedge contains s_i totally or partially. The region of vertices of separating wedges is $\overline{A} \cap \overline{B}$, where $A = \cup A_i$ and $B = \cup B_i$.

1) **Computing $\overline{A}$**

We compute the *final polygonals* or $\mathcal{POL}$, formed by circular arcs, segments and half lines that, as we are going to see, separate A from $\overline{A}$ (Figure 11).

Ordering circular arcs: We order the circular arcs $s_i = p_i p'_i$ of P' (p_i and p'_i are the endpoints) with two orders with respect to $CH(Q)$: clockwise and counterclockwise. Let l_i^+ ($l_i'^+$) be the half line tangent to $CH(Q)$ going through p_i (p'_i) such that $CH(Q)$ is on the left side going from p_i (p'_i) to $CH(Q)$ and starting at the tangency point. We order the half lines in an angular clockwise way and we order the circular arcs according to the first endpoint that appears in the order given by the half lines. If there are several circular arcs with the first endpoint sharing the same half line, the order is given by the proximity of the first endpoint to the tangency point of the half line with $CH(Q)$. Once we have ordered the circular arcs we denote by p_i the first endpoint of the circular arc and by p'_i the second one. Let $s_1 = p_1 p'_1$ be the first circular arc. If a circular arc $s_i = p_i p'_i$ intersects l_1^+ in a point p''_i we divide it into two circular arcs $p_i p''_i$ and $p''_i p'_i$ and delete s_i. The number of circular arcs increases in at most $n-1$. Let P' be the set of these circulars arcs. Reordering P' according to the criteria above we are sure that performing a clockwise angular sweep starting in l_1^+ we find all the circular arcs after just one round.

For the counterclockwise ordering we proceed in a similar way using the order of r_i^+ ($r_i'^+$), half line tangent to $CH(Q)$ containing the point p_i (p'_i) and starting at the tangency point that has $CH(Q)$ on its right going from p_i (p'_i) to the tangency point. The ordering process requires $O(n \log N)$ time.

Final polygonals: The polygonals of $\mathcal{POL}$ are formed by circular arcs, segments and half lines. From any point of one polygonal of $\mathcal{POL}$ we *see* $CH(Q)$

without any obstacle. The circular arcs define a clockwise (counterclockwise) angular intervals and they are angular-monotone.

Because two circles can intersect in at most two points, the computation of the clockwise angular lower envelope with the modification of the Hershberger algorithm [6,15] requires $O(\lambda_3(n) \log n)$ time, where $\lambda_3(n) = O(n\alpha(n))$. Nevertheless, the complexity of the clockwise angular lower envelope of blue circular arcs is linear since it is less than or equal to the combinatorial complexity of a *single face* in a arrangement of n closed Jordan curves (blue circles) in the plane, which is $\lambda_2(n) = 2n - 1$ (see theorem 5.7 of [15]). The same applies to the counterclockwise angular lower envelope and to the merging of both envelopes. Taking into account this complexity, we compute the clockwise angular lower envelope using the algorithm given by the corollary 6.2 of [15], that consists of making a partition of P' in two subsets, each of size at most $\lceil \frac{n}{2} \rceil$, compute the clockwise angular lower envelope of each one recursively, and then merge these subenvelopes to obtain the overall envelope. For merging the two clockwise angular lower subenvelopes we sweep clockwise angularly a half line tangent to $CH(Q)$ advancing in discrete steps over the clockwise angular intervals determined by the transition points of the subenvelopes and spending $O(1)$ time in each step. The merge step requires $O(\lambda_2(n))$ time. If $T(n)$ is the maximum running time of the algorithm, we obtain the recurrence $T(n) = 2T(\frac{n}{2}) + O(\lambda_2(n))$. The solution to this recurrence is $O(n \log n)$.

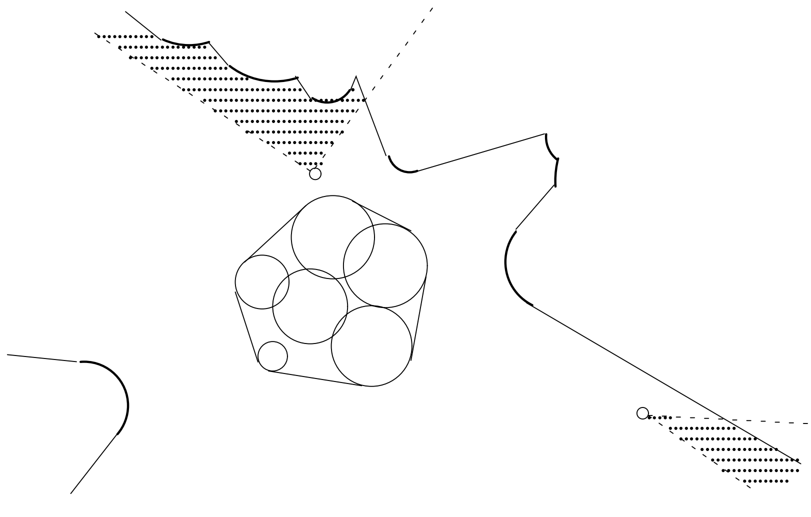


Figure 11. Regions of vertices of separating wedges.

In the same way, we compute the counterclockwise angular lower envelope. The merging of the two envelopes, *the circle polygonals*, is obtained as the above merging in $O(\lambda_2(n))$ time, which is the number of transition points. With the order of the endpoints of the circular arcs we compute in $O(n)$ time the *extreme polygonals* formed by segments and half lines. To obtain $\mathcal{POL}$ we merge the circle polygonals and the extreme polygonals in $O(n)$ time making a clockwise angular sweep with a half line tangent to $CH(Q)$ advancing in discrete steps over the clockwise angular intervals of the circular arcs and segments. The complexity of the final polygonals is $O(n)$. Note that $\overline{A}$ is a (possibly unbounded) star-shaped polygon (with also circular arcs as edges) whose kernel contains $CH(Q)$.

2) **Computing** $\overline{B}$

Two consecutive polygonals of $\mathcal{POL}$ give a wedge region of $\overline{B}$ defined by the half lines l_i^- from the end of the first polygonal and r_{i+1}^- from the beginning of the second polygonal if l_i^- and r_{i+1}^- intersect at a point denoted by p_i'''. Using $\mathcal{POL}$ we determine all the at most n wedge regions of $\overline{B}$ in $O(n)$ time.

3) **Computing** $\overline{A} \cap \overline{B}$

A wedge region of $\overline{B}$ is useless if the point p_i''' lies in A. We eliminate the useless wedge region of $\overline{B}$ and compute the part of $\mathcal{POL}$ between the half lines of the useful wedge regions. By the angular monotonicity, each half line intersects $\mathcal{POL}$ in at most one point (or it contains a half line of $\mathcal{POL}$). In $O(n \log N)$ time we compute the regions of $\overline{A} \cap \overline{B}$. These regions are (possibly unbounded) star-shaped polygons (with also circular arcs as edges) or fans whose kernel contains the point p_i''' (Figure 11). As a consequence of the above discussion we have the following theorem.

Theorem 11 *Let P and Q two disjoint sets of n and m circles in the plane respectively. To decide whether they are wedge separable and to compute the region of vertices of separating wedges can be done in $O(N \log N)$ time.*

Wedges with maximum and minimum angle

The vertex of the separating wedge with maximum angle is in the apex of a fan of $\overline{A} \cap \overline{B}$. It can be determined computing in constant time the angle of the wedge with vertex in p_i''' for each of the at most n fans. If there is an unbounded fan the minimum angle is zero. Otherwise, the wedge with minimum angle is in the convex extremes of the fan, *the tips*. The number of tips is $O(n)$ and for each tip we compute its angle in constant time (Figure 11).

Proposition 12 *Let P and Q be two disjoint wedge separable sets of n and m circles in the plane, respectively. If we have pre-computed the region of vertices of separating wedges, the wedges with maximum and minimum angle can be found in $O(N)$ time.*

2.4 Applications to circular and polygonal separability

Circular separability

In [2] it is shown that for two sets of line segments P and Q with a total of N segments meeting only at their endpoints, it is possible to compute in $O(N \log N)$ time and $O(N)$ space all the largest circles separating P and Q. For the general case the following theorem is shown.

Theorem 13 *[2] For two sets of line segments P and Q containing a total of N segments, it is possible to compute in $O(N\alpha(N)\log^2 N)$ deterministic time, or in $O(N\alpha(N)\log N)$ randomized time and $O(N\alpha(N))$ space all locally largest circles separating P and Q.*

As pointed out in [2]: *to prove the theorem it is sufficient to take into account separation of the external cell of the arrangement of line segments of set P, and containing it, a single cell of the arrangement of line segments of Q.*

It is clear that if there exists a circle separating Q from P, a largest circle C has to contain $CH(Q)$ and has to be tangent to segments of P in some points. The tangency points of C with segments of P satisfy the property that these points see $CH(Q)$: we say that a point p *sees* $CH(Q)$ if the region defined by the half lines from p tangent to $CH(Q)$ and the part of $CH(Q)$ between the half lines does not contain any obstacle. Then, any circle separating Q from P separates also $CH(Q)$ from the set of segments of the final polygonals obtained in this section for the wedge separability of Q from P. The final polygonals are formed by $O(N\alpha(N))$ segments meeting only at their endpoints and we have computed the final polygonals in $O(N \log N)$ time. Then applying the above theorem to the at most N segments of $CH(Q)$ and $O(N\alpha(N))$ segments of the final polygonals, it is possible to compute all the largest circles separating Q from P in time $O(N\alpha(N)\log(N\alpha(N))) = O(N\alpha(N)\log N)$.

Theorem 14 *For two sets of line segments P and Q containing a total of N segments, it is possible to compute in $O(N\alpha(N)\log N)$ deterministic time and $O(N\alpha(N))$ space all locally largest circles separating P and Q.*

Polygonal separability

The final polygonals of the segments can also be used to generalize the next theorem of Edelsbrunner and Preparata about the minimum (in the number of edges) polygonal separator of two sets of points P and Q, $card(P \cup Q) = N$.

Definition 15 *[4] A convex k-gon is the intersection of k but no fewer closed half planes and a convex k-gon is said to separate two point sets if it contains* *one and its interior avoids the other. This k-gon is also referred to as a k-separator of the two sets.*

Given two finite sets of points P and Q, construct a separating convex k-gon for the smallest possible integer k.

Theorem 16 *[4] Given two finite sets P and Q of points in the plane, the construction of the minimum polygonal separator (or the decision that no such separator exists) can be done in time $O(N \log N)$ and this is optimal.*

Let P and Q be two sets of segments, $card(P \cup Q) = N$. We compute the $O(N\alpha(N))$ segments of the final polygonals in $O(N \log N)$ time. Then, we substitute Lemma 3.2 in [4] by the following one:

Lemma 17 *If there is a k-separator of P and Q with minimum k, then there is a k-separator with at least one edge having one endpoint in one of the reflex vertices of the final polygonals.*

PROOF. The part of a polygonal between two consecutive reflex vertices is convex. Let R be a k separator, with minimum k, such that no vertex of R is a reflex vertex of the final polygonals. We can construct a new k-separator R' by a continuous transformation of R. Let e be an edge of R, translate e until it touches $CH(Q)$ in a point q of $CH(Q)$, and extend e until the endpoints touch the final polygonals. If one of the endpoints is not a reflex vertex, rotate the extended e in a counterclockwise direction (the endpoints moving on convex parts of the final polygonals) until it becomes aligned with an edge $\{q, q_1\}$ of $CH(Q)$ or one endpoint of the extended e is a reflex vertex. In the first case repeat the process with pivot in q_1. □

The proof of the lemma is similar to the proof of the mentioned Lemma 3.2 with some changes. The final polygonals have at most $2N$ reflex vertices since these are endpoints of the original segments. It is easy to define a predecessor/successor relation between the segments of the final polygonals and construct it in $O(N\alpha(N))$ time. A greedy separator starts in a reflex vertex and an edge is formed with the line tangent to $CH(Q)$ until it intersects the successor segment in the final polygonals. A greedy separator defines a natural partition of the reflex vertices. By simply following the rest of the proof of the above theorem in [4] we prove the following one.

Theorem 18 *Given two finite sets P and Q of N segments in the plane, the construction of the minimum polygonal separator (or the decision that no such separator exists) can be done in time $O(N \log N)$ and this is optimal.*

We observe that the above results can also be proved for two sets of circles.

3 Strip separability

We study now the problem of separability of two disjoint sets of objects in the plane by a strip. Suppose that P and Q are the sets of *blue* and *red* objects, respectively. It is clear that if P and Q are strip separable they are wedge separable, just move a little bit one of the parallel lines that defines the strip. The difference is that before we looked for wedge vertices (two parameters) and now it is enough to determine the slope of the strip (one parameter), which is the slope of the parallel lines and it is given by the counterclockwise angle in $[0, \pi)$ formed by the horizontal with one of the parallel lines.

A necessary condition for the strip separability is that the convex hull of one of the sets is monochromatic. In $O(N \log N)$ time we compute $CH(P)$, $CH(Q)$ and determine if there is a monochromatic set. We admit that the parallel lines of the strip can *lean on* the objects of P and Q. The solution of the problem is to determine the slope intervals of the separating strips. We study the strip separability problem for different kinds of objects.

3.1 Separating points by strips

Let $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, q_m\}$ two disjoint point sets in the plane. Suppose that they are not line separable and $CH(Q)$ is monochromatic. We want to determine the slope intervals of the separating strips. First, we note the restriction of a point p_i: the half lines l_i^+ and r_i^+ have counterclockwise angles α_{l_i} and α_{r_i} with the horizontal. It is clear that the interval $(\alpha_{l_i}, \alpha_{r_i})$ contains no solutions, otherwise the strip would contain the point p_i. We call this interval the *negative interval* of p_i (Figure 12).

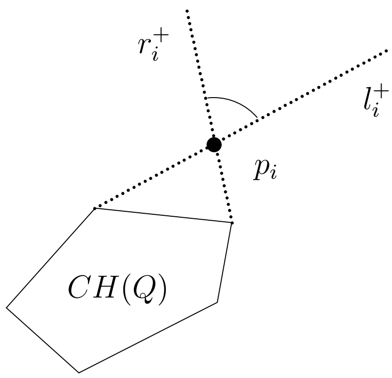


Figure 12. Negative interval.

Moreover, if $p_i \prec_Q p_j$, the negative interval of p_j is contained in the negative interval of p_i. Therefore, the final polygonals constructed in the previous section correspond to the points of P that can take part in the computation of the strips. In fact, we only need the first and the last points $\{p_i, p_k\}$ in each polygonal, giving the negative interval $(\alpha_{l_k}, \alpha_{r_i})$. Using the final polygonals and in $O(n)$ time, we obtain the at most n disjoint negative intervals within $[0, 2\pi)$. In $O(n)$ time we reduce the negative intervals in $[\pi, 2\pi]$ to intervals in $[0, \pi)$ and we compute their union. The complement of this union with respect to $[0, \pi)$ are the slope intervals of the separating strips, *strip intervals* for short. If the complement is the empty set, the sets are not strip separable. Note that the negative intervals are open and therefore the strip intervals are closed.

Proposition 19 *Let P and Q be two disjoint sets of n and m points in the plane respectively. To decide whether they are strip separable computing the strip intervals can be done in $O(N \log N)$ time.*

A natural problem in the case that the sets are strip separable is to determine the narrowest and the widest strip. Suppose that Q is strip separable from P and we have computed the at most n strip intervals.

The narrowest strip

The narrowest strip is defined by two parallel lines tangent to $CH(Q)$ in a pair of antipodal points and its computation depends on $CH(Q)$ and on the strip intervals. Note that it is similar to the computation of the width of a convex polygon, i.e. the smallest distance between parallel supporting lines of the polygon. In our case we compute the width of $CH(Q)$ restricted to the strip intervals.

In $O(m)$ time we generate all the pairs of antipodal points of $CH(Q)$ in the counterclockwise order [14,17]. For each pair we compute in constant time the interval (or two intervals) in $[0, \pi)$ of common slopes of the supporting lines in these points. In the intersection of this interval with the strip intervals, the variation of the distance between supporting lines is monotone or upwards unimodal. Therefore, the minimum distance corresponds to the extremes of the intersection. In constant time we compute the extremes, the minimum width of the strips corresponding to a pair of antipodal points and update the minimum. Note that it is possible to have strips with minimum width in different antipodal points.

Proposition 20 *Let P and Q be two disjoint strip separable sets of n and m points in the plane respectively. If we have pre-computed the strip intervals, the narrowest strip can be found in $O(N)$ time.*

The widest strip

We exclude here the situation in which P and Q are line separable since in this case the widest strip is infinite. The widest strip only depends on the points of P in the final polygonals. We consider the problem in the dual plane by means of the transformation: $L : y = 2ax - b \longleftrightarrow p : (a, b), \quad D(L) = p, \;\; D(p) = L.$

We transform the points of $CH(Q)$ and P so that $D(CH(Q))$ are red lines and $D(P)$ are blue lines. Previously, we make a change of coordinates for simplifying the situation in the dual plane. If in the primal plane all the strip intervals are reduced to points, in $O(n)$ time we compute the widest strip. Otherwise we choose a strip interval $[\alpha_1, \alpha_2]$, $\alpha_1 \neq \alpha_2$ and we make a change of coordinates taking as Y axis a line with slope within (α_1, α_2) and origin such that the points of $P \cup Q$ have different abscissas. In this way, the blue lines in the dual plane have different slopes out of the slope interval (a_{q_1}, a_{q_m}), where q_1 and q_m are the red points with minimum and maximum abscissa, respectively (Figure 13).

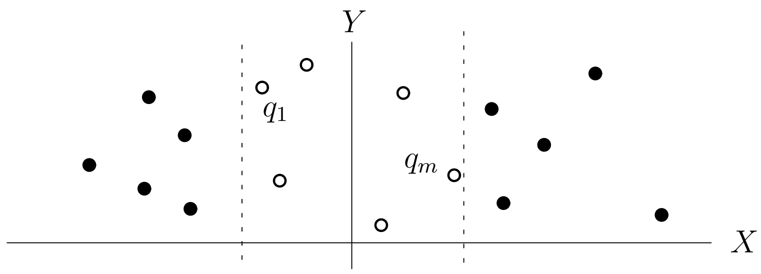


Figure 13. Change of coordinates.

We order the lines according to increasing slope (counterclockwise) departing from the vertical (there are no vertical lines). The situation in the dual plane is as shown in Figure 14. The convex hull of $D(CH(Q))$ determines two regions: an upper region U and a lower region L, limited by E_U and E_L, respectively. All the red lines lie between E_U and E_L, but no blue line, since in the primal plane there are no blue points inside the red convex hull. A blue line intersects E_U in one point and intersects E_L in other point, determining a *null vertical slab* defined by the two parallel vertical lines that pass through the intersection points of the blue line with E_U and E_L (Figure 14).

In the arrangement of the blue lines we consider the zone of E_U (the zone of E_L) as the set of cells of the arrangement intersected by E_U (E_L). Then we define the *upper cells* (in U) and the *lower cells* (in L) as follows:

Definition 21 *The upper cells (lower cells) are the intersection of the zone of E_U (E_L) with the region U (L) and the complementary of the null vertical slabs.*

Note that the cells are convex regions limited by segments of red lines, segments of blue lines or vertical segments from the null vertical slabs. We call *opposite cells* two cells (an upper cell and a lower cell) that have intersection with some common vertical line. Two blue segments of opposite cells that have intersections with some common vertical line, (*opposite blue segments*), determine in the primal plane two blue points. These blue points have the property that there exist two parallel lines passing through the blue points and defining a strip interval, since in the dual plane *between the blue segments* there are only red lines and no blue lines. The maximum width of the strips in this interval can be determined in constant time. To compute the widest strip we make a plane-sweep with a vertical line halting at the opposite blue segments. At each stop we compute in constant time the maximum width of the strip interval and update the maximum.

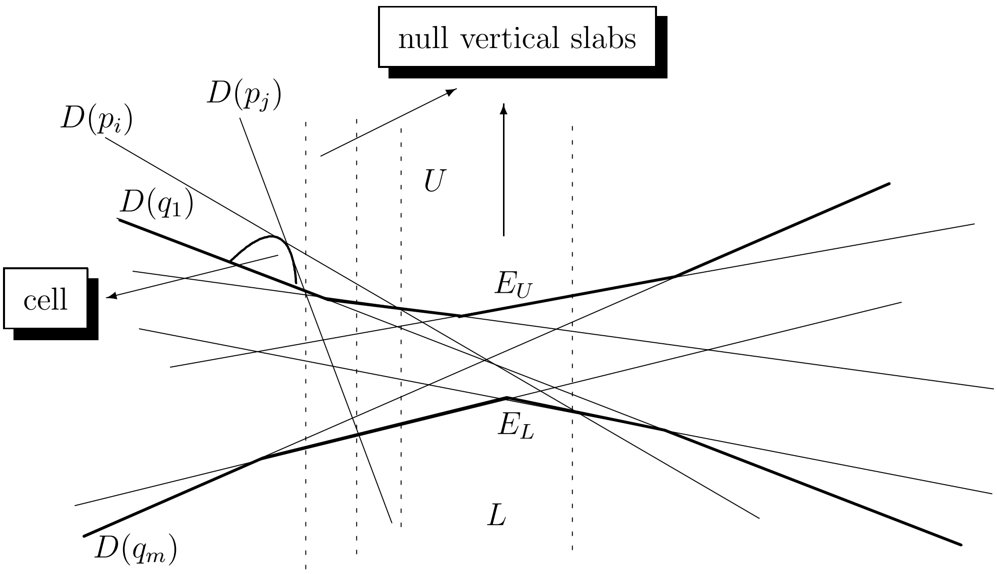


Figure 14. Dual arrangement.

Let $A_U(n,m)$ ($A_L(n,m)$) be the ordered set of segments that define the upper (lower) cells generated by the arrangement of n blue lines and m red lines. Let $c_U(n,m)$ ($c_L(n,m)$) be the respective cellular complexity, i.e. the number of segments in all the upper (lower) cells. We define $A(n,m) = A_U(n,m) \cup A_L(n,m)$ and $c(n,m) = c_U(n,m) + c_L(n,m)$.

Lemma 22 *The number of upper and lower cells generated by the arrangement of m red lines and n blue lines is less than or equal to $2n+2$.*

PROOF. We count the number of upper cells. Initially, the region U is a single cell. The first blue line divides the cell into two cells. Therefore, if d_n represents the number of upper cells after the introduction of n blue lines, we have that $d_1 = 2$. Suppose we have introduced $n-1$ blue lines by increasing slope starting from the vertical. The *null vertical slab* of the next blue line can divide at most an upper cell in two cells. In any other case the blue line only modifies or deletes upper cells. Therefore, $d_n \leq d_{n-1} + 1$, and then $d_n \leq n + 1$. □

Lemma 23 *The cellular complexity is* $c(n, m) = O(N)$.

PROOF. In the cellular complexity we consider all the segments that define upper and lower cells: vertical segments, red segments and blue segments. We count separately each kind of segments and only for the upper cells. For the lower cells we proceed analogously.

i) The number of vertical segments is less than or equal to n, since they are generated by the blue lines and each blue line can produce at most one vertical segment in U (null vertical slab).

ii) Initially, there are at most m red segments. A blue line can divide a red segment into two segments, increasing the number by one. Therefore the number of red segments is less than or equal to $m + n$.

iii) Now, we count the blue segments. In the arrangement of the upper cells, a blue line contributes either with at most one segment or with two or more segments. Note that in an upper cell there are not two segments from the same line. We say that two blue lines *share* segments in an upper cell if there is a segment from each line in the cell. It is easy to proof that two blue lines can share segments in at most one upper cell. By the above lemma, the number of upper cells is less than or equal to $n + 1$. The lines that contribute with only one segment, give an overall contribution of at most n segments. The lines that contribute with two or more segments are counted in the following way:

1) Lines with slope between a_{q_1} and the vertical, ordered by decreasing slope in this interval: let $k_1, k_2, \ldots, k_s$ be the number of segments contributed by each line. It is clear that a line can share a segment with a previous line in at most one cell, since the segment contribution is from the last intersection with the previous lines and therefore, only with the last previous line can share a segment in a cell (Figure 14). If we count cells, in the worst case we will have that $k_1 + k_2 - 1 + \ldots + k_s - 1 \leq n + 1$, and then $k_1 + k_2 + \ldots + k_s \leq n + s$.

2) Lines with slope between the vertical and a_{q_m}, ordered by increasing slope in this interval. Let $k'_1, k'_2, \ldots, k'_t$ be the number of segments contributed by each line. With a similar reasoning we obtain $k'_1 + k'_2 + \ldots + k'_t \leq n + t$. Taking into account that $s + t \leq n$, we can deduce that the total number of segments in the upper cells is less than or equal to $6n + m$. □

Lemma 24 $A(n, m)$ *can be computed in* $O(N \log N)$ *time.*

PROOF. $A_U(n,m)$ is computed by divide and conquer in time $T(n) = 2T(n/2) + f(n,m)$, where $f(n,m) = O(N)$ is the time needed to merge the cells of the two subproblems, since the number of segments of the cells of the subproblems is $O(n+m)$. Making a plane sweep we get the resulting cells. □

Theorem 25 *Let P and Q be two disjoint strip separable sets of n and m points in the plane respectively. The widest strip can be found in $O(N \log N)$ time.*

PROOF. By the above discussion and using the previous lemmas, we can obtain $A(n,m)$ in $O(N \log n)$ time. Then we make a plane sweep with a vertical line halting in the $O(n)$ opposite blue segments [9]. □

3.2 Separating segments by strips

In this case the strip intervals can be obtained from the final polygonals constructed for the wedge separability of segments. Each polygonal defines a *negative interval* in $[0, 2\pi)$ where there are no slopes of strips. In $O(n\alpha(n))$ time we obtain the at most n disjoint negative intervals. We compute their union and reduce it to the interval $[0, \pi)$. The strip intervals are the complement of the union with respect to the interval $[0, \pi)$. There are at most n strip intervals and they are closed intervals.

Proposition 26 *Let P and Q be two disjoint sets of n and m segments in the plane respectively. To decide whether they are strip separable and to compute the strip intervals can be done in $O(N \log N)$ time.*

The narrowest strip

The narrowest strip depends on $CH(Q)$ and on the strip intervals. We have to compute the width of the convex polygon $CH(Q)$ [14,17] with the restriction to the strip intervals.

Proposition 27 *Let P and Q be two disjoint strip separable sets of n and m segments in the plane respectively. If we have pre-computed the strip intervals, the narrowest strip can be found in $O(N)$ time.*

The widest strip

The widest strip depends on the segments of P and on the strip intervals. If all the strip intervals are points, in time $O(n)$ we analyze each one in constant time and determine the widest strip. Otherwise we choose a strip interval and we make a change of coordinates taking as Y axis a line with slope within the strip interval and such that the segment endpoints have different abscissas. We consider the problem in the dual plane by doing the same transformation as for the point sets. We transform the points of $CH(Q)$ and the endpoints of segments of P in red and blue lines. The lines in the dual plane have different slopes.

With the red lines we compute the upper envelope, E_U, and the lower envelope, E_L, in $O(m \log m)$ time. The dual of a segment of P located on the left (right) side of the Y axis gives in the dual plane a double wedge defined by the two lines with negative (positive) slope that are the dual of the segment endpoints. The dual arrangement is similar to the case of point sets, with the restriction of the double wedge regions that can be handled by *null vertical slabs* defined by the intersection points of the two lines of the double wedge with E_U and E_L (Figure 15). Therefore, the number of upper (lower) cells and the cellular complexity are less than or equal to the corresponding values obtained in the widest strip for point sets. It is clear that we can apply a similar algorithm to determine the widest strip in $O(N \log N)$ time.

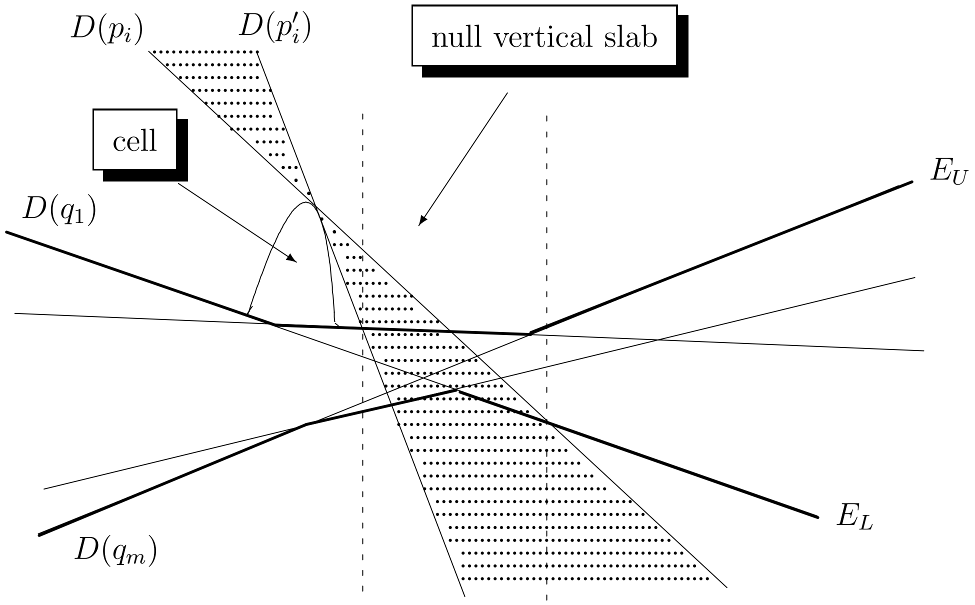


Figure 15. Dual of segments.

Theorem 28 *Let P and Q be two disjoint strip separable sets of n and m segments in the plane respectively. The widest separating strip can be found in $O(N \log N)$ time.*

The problem of the strip separability for polygons is easily reduced to the strip separability problem for segments. Hence we have:

Corollary 29 *Let P and Q be two disjoint sets of polygons in the plane with n and m total segments respectively. To decide whether they are strip separable and to compute the strip intervals and the narrowest and the widest strip can be done in $O(N \log N)$ time.*

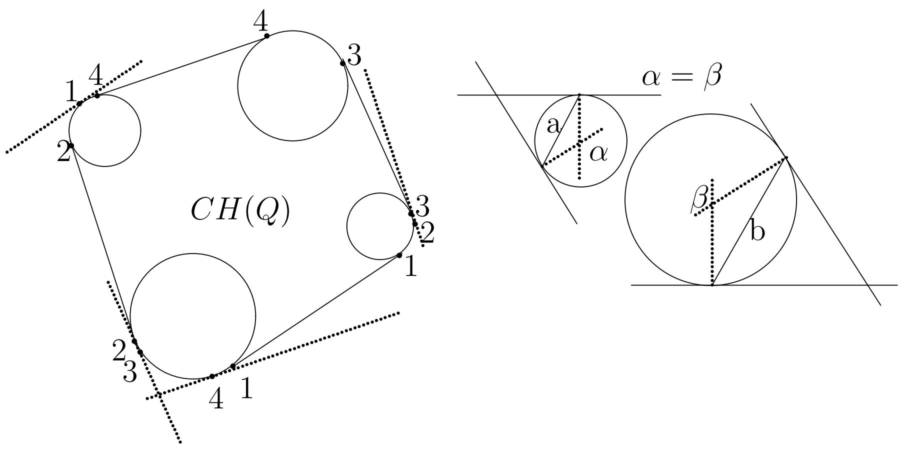
3.3 Separating circles by strips

In this case, using the final polygonals of the wedge separability, we compute the union of the negative intervals. The complement of this union with respect to $[0, \pi)$ is the set of at most n closed strip intervals or points.

Proposition 30 *Let P and Q be two disjoints set of n and m circles in the plane respectively. To decide whether they are strip separable and to compute the strip intervals can be done in $O(N \log N)$ time.*

The narrowest strip

The narrowest strip depends on $CH(Q)$ and on the strip intervals. For computing the strip with the minimum width we use calipers [14,17]. Starting on a segment of $CH(Q)$ we move the calipers (two parallel supporting lines) counterclockwise with the minimum angle, taking into account the restriction of the strip intervals and determining pairs of antipodal circular arcs (Figure 16a). This process can be done in $O(m+n)$ time [13]. For a pair of antipodal circular arcs the minimum width (distance between the supporting lines) corresponds to the endpoints of the circular arcs, since their chords are parallel and the variation of the width is upwards unimodal (Figure 16b).



a) b)

Figure 16. a) Computing antipodal arcs, b) antipodal arcs.

Proposition 31 *Let P and Q be two disjoint strip separable sets of n and m circles in the plane respectively. If we have pre-computed the strip intervals, the narrowest strip can be found in $O(N)$ time.*

The widest strip

The widest strip depends on the circular arcs of the final polygonals. If all the strip intervals are points, we analyze each one in constant time and determine the widest strip in $O(n)$ overall time. Otherwise we choose a strip interval and we make a change of coordinates taking as Y axis a line with slope within the strip interval and such that the endpoints of the blue circular arcs of the final polygonals and the endpoints of the red circular arcs of $CH(Q)$ have different abscissas (Figure 17a). We consider a correlation of the projective plane in the dual such that the tangents to a circle are transformed into the points of a hyperbola. Any vertical line in the dual plane intersects the hyperbola in two points, so we can consider the upper branch and the lower branch of the hyperbola. The dual of the points of a circle are the lines tangent to a hyperbola. The dual of the points of a circular arc correspond to lines tangent to a hyperbolic arc (Figure 17b). We proceed in the following way:

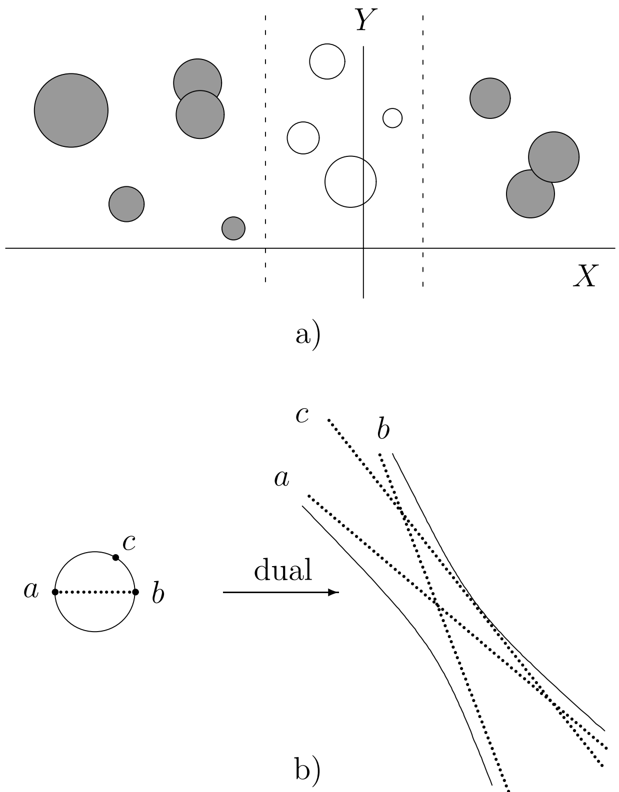


Figure 17. a) Change of coordinates, b) dual of a circle.

Dual of $CH(Q)$: each of the m red circles gives in the dual plane two hyperbolic branches. Each branch is an unbounded x-monotone Jordan curve and any pair of them intersects at most in two points. We compute the upper (lower) envelope E_U (E_L) of the at most $2m$ hyperbolic branches using Theorem 6.1 of [15] in $O(\lambda_2(m) \log m) = O(m \log m)$ time. The complexity of E_U and E_L is less than or equal to the complexity of a single face of $2m$ unbounded Jordan curves, which is $O(\lambda_2(m)) = O(m)$ (see Theorem 5.7 of [15]).

Dual of blue circles: first, we dualize the circles of P which contribute with at least one circular arc to the final polygonals. In the dual plane we compute the intersections of each blue hyperbolic branch with E_U and with E_L, obtaining the null vertical slabs formed by the vertical parallel lines that pass through the intersection points. We compute the union of the intervals defined by the null vertical slabs in order to get the disjoint set of the null vertical slabs. In these slabs there are no upper or lower cells. All these computations can be done in $O(n \log N)$ time. Next, we dualize only the circular arcs of the final polygonals obtaining lines tangent to hyperbolic arcs. From the hyperbolic arcs we take only the part of the arcs that is over E_U, obtaining an arrangement HA_U, and the part of the arcs that is under E_L obtaining a second arrangement HA_L. The complexity of the hyperbolic arcs is $O(n)$, since the complexity of the final polygonals of circular arcs is also $O(n)$.

The arrangement of the upper (lower) cells can be obtained computing the lower (upper) envelope of HA_U (HA_L) restricted to the union of null vertical slabs. The intersection point of two hyperbolic arcs in the dual plane corresponds in the primal plane to a common tangent to the circular arcs.

Given a circle R and a circle C exterior to R, we say that a point p of C *sees* R if the region defined by the half lines from p tangent to R and the part of R between the half lines does not contain any obstacle. A (connected) circular arc a of C *sees* R if all the points of a see R.

Lemma 32 *Let C_1 and C_2 be blue circles exterior to a red circle R, and let a_1 and a_2 be circular arcs of C_1 and C_2, respectively, that see R. At most one of the tangents between C_1 and C_2 touches points of both a_1 and a_2.*

PROOF. The arc a_1 (a_2) is contained in the arc obtained with the interior tangents between C_1 (C_2) and R (Figure 18a). Any tangent line to C_1 (C_2) at a point of the arc a_1 (a_2) separates C_1 (C_2) from R.

The interior tangents of C_1 and C_2 do not pass through points of a_1 and a_2: if C_1 does not intersect C_2 and p_1 and p_2 are the tangency points of an interior tangent, then p_1 and p_2 are not visible from R simultaneously, since their *visibility regions* are disjoint (Figure 18a). If C_1 intersects C_2, there are no interior tangents. If C_1 is tangent to C_2, the tangency point has null visibility.

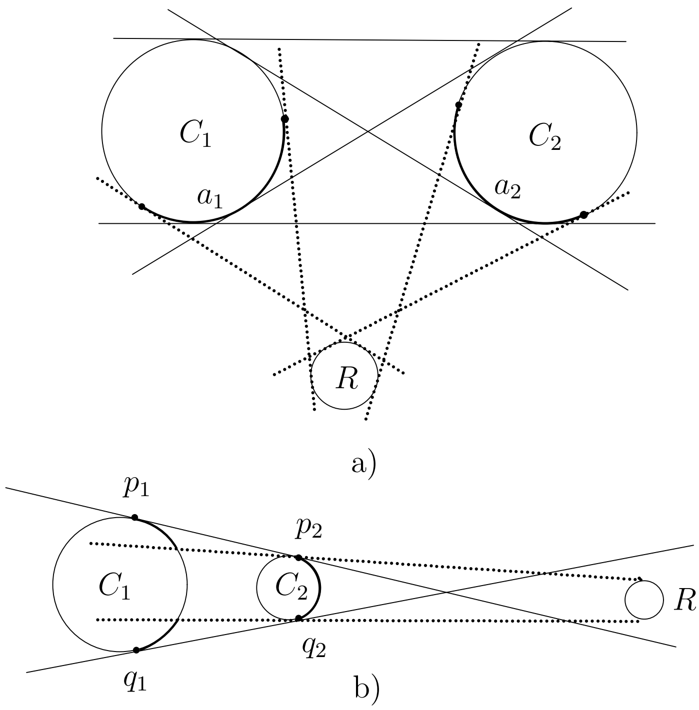


Figure 18. Tangents between two circles.

At most one exterior tangent of C_1 and C_2 goes through points of a_1 and a_2 simultaneously. Let p_1, p_2 (q_1, q_2) be the tangency points of one (the other) exterior tangent between C_1 and C_2: if the exterior tangents are parallel, the visibility regions of p_1 and p_2, and of q_1 and q_2 are disjoint. If the exterior tangents intersect, the visibility region of p_1, p_2, q_1 and q_2 is the wedge formed by the exterior tangents. If R is within this region, there are three different situations: 1) the exterior tangents are tangent to R, then one of the blue circles is not visible from R; 2) one exterior tangent is tangent to R and the other one is not, then a_1 and a_2 share only the other exterior tangent; 3) the exterior tangents are not tangent to R, then the farthest blue circle from R has two disjoint visibility arcs, each one sharing an exterior tangent with the visibility arc of the nearest blue circle to R (Figure 18b). □

The lemma is also true for convex closed curves, each pair intersecting in at most two points. The lemma above ensures that each pair of hyperbolic arcs intersect in at most one point and therefore, we can compute the lower envelope of HA_U (upper cells) and the upper envelope of HA_L (lower cells) in $O(\lambda_{1+1}(n) \log n) = O(n \log n)$ time.

The complexity of the upper and lower envelopes is $O(n\alpha(n))$. Then, making a plane sweep with a vertical line on the cells, we determine the at most $O(n\alpha(n))$ pairs of circular arcs that define strips. In constant time we compute the maximum width of the strip interval of each pair of circular arcs. The maximum width corresponds to the intersection points of the circles with the line through both centers of the circles in the case that this intersection points belong to the pair of circular arcs. Otherwise, it corresponds to one common endpoint of the pair of circular arcs.

Theorem 33 *Let P and Q be two disjoint strip separable sets of n and m circles in the plane respectively. The widest strip can be found in $O(N \log N)$ time.*

4 Conclusions

We have solved the problems of separating two sets of points, segments, polygons or circles in the plane with wedges and strips, obtaining also the wedges with the maximum and minimum angle and the narrowest and the widest strip. Our algorithms run in $O(N \log N)$ time. As applications, we improve the computation of all the largest circles separating two sets of line segments by a $\log n$ factor, and we generalize the algorithm for computing the minimum polygonal separator of two sets of points to two sets of line segments with the same running time. In case that there are no separating wedges or strips we are currently studying the separability problems from other points of view, either by weakening the separability condition, allowing some points to be missclassified, or by using more than one separator.

Acknowledgements

We are grateful to John Hershberger and Joseph S. B. Mitchell for useful comments. An anonymous referee deserves our thanks for a careful reading of the manuscript and for many valuable suggestions and comments.

References

[1] B. K. Bhathacharya, *Circular Separability of Planar Point Sets*, Computational Morphology, G. T. Toussaint ed. North Holland, Amsterdam, 1988.

[2] J.-D. Boissonnat, J. Czyzowicz, O. Devillers, J. Urrutia, M. Yvinec, *Computing Largest Circles Separating Two Sets of Segments*, Proceedings of the Eighth Canadian Conference on Computational Geometry, 1996, pp. 173-178.

[3] O. Devillers, M. J. Golin, *Incremental Algorithms for Finding the Convex Hulls of Circles and the Lower Envelopes of Parabolas*, Proceedings of the Sixth Canadian Conference on Computational Geometry, 1994, pp. 153-158.

[4] H. Edelsbrunner, F. P. Preparata, *Minimum Polygonal Separation*, Information and Computation, 77, 1988, pp. 218-232.

[5] S. Fish, *Separating Point Sets by Circles and the Recognition of Digital Disks*, Pattern Analysis and Machine Intelligence 8 (4), 1986.

[6] J. Hershberger, *Finding the Upper Envelop of* n *Line Segments in* $O(n \log n)$ *Time*, Information Processing Letters, 33, 1989, pp. 169-174.

[7] M. E. Houle, *Weak Separation of Sets*, Ph. D. Thesis, McGill University, 1989.

[8] F. Hurtado, M. Noy, C. Seara, *Dominación Geométrica respecto de un Convexo*, (in Spanish) Manuscript.

[9] F. Hurtado, M. Noy, C. Seara, *Separación de Puntos mediante Cuñas y Bandas*, (in Spanish) VII Encuentros de Geometría Computational, Madrid, 1997, pp. 45-55.

[10] N. Megiddo, *Linear-time Algorithms for Linear Programming in* $\mathbb{R}^3$ *and Related Problems*, SIAM J. Computer, Vol 12, No 4, 1983, pp. 759-776.

[11] J. O'Rourke, S. R. Kosaraju, N. Megiddo, *Computing Circular Separability*, Discrete Comput. Geom. 1 (1), 1986, pp. 105-113.

[12] D. Rotem, J. Urrutia, *Circular Permutations Graphs*, Networks, 11, 1982, pp. 429-437.

[13] D. Rappaport, *A Convex Hull Algorithm for Discs, and Applications*, Computational Geometry: Theory and Applications, 1, 1992, pp. 171-187.

[14] M. I. Shamos, *Computational Geometry*, Ph. D. Thesis, Yale University, 1978.

[15] M. Sharir, P. K. Agarwal, *Davenport-Schinzel Sequences and their Geometric Applications*, Cambridge University Press. 1995.

[16] J. Stoer, C. Witzgall, *Convexity and Optimization in Finite Dimension I*, Springer-Verlag, New York, 1970.

[17] G. T. Toussaint, *Solving Geometric Problems with Calipers*, Proceedings of IEE MELECON'83, 1983.