May 1986

UILU-ENG-86-2214
ACT-70

COORDINATED SCIENCE LABORATORY
College of Engineering
Applied Computation Theory

MINIMUM POLYGONAL SEPARATION

H. Edelsbrunner
F. P. Preparata

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

[Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
None

2a. SECURITY CLASSIFICATION AUTHORITY
N/A

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release,
distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
UILU-ENG-86-2214 (ACT-70)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
N/A

6a. NAME OF PERFORMING ORGANIZATION
Coordinated Science
Lab., University of Illinois

6b. OFFICE SYMBOL *(If applicable)*
N/A

7a. NAME OF MONITORING ORGANIZATION
National Science Foundation & Amoco
Fnd. Fac. Dev. Comput. Sci.

6c. ADDRESS *(City, State and ZIP Code)*
1101 W. Springfield Avenue
Urbana, IL 61801

7b. ADDRESS *(City, State and ZIP Code)*
1800 G. Street, Washington, DC 20550
200 E. Randolph Drive, Chicago, IL 60601

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
NSF and Amoco
Fnd. Fac. Dev. Comput. Sci.

8b. OFFICE SYMBOL *(If applicable)*
N/A

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
1-6-44862 NSF Grant ECS 84-10902

8c. ADDRESS *(City, State and ZIP Code)*
1800 G. Street, Washington, DC 20550
200 E. Randolph Drive, Chicago, IL 60601

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
N/A	N/A	N/A	N/A

11. TITLE *(Include Security Classification)*
Minimum Polygonal Separation

12. PERSONAL AUTHOR(S)
H. Edelsbrunner & F.P. Preparata

13a. TYPE OF REPORT
final

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Yr., Mo., Day)*
May 1986

15. PAGE COUNT
16

16. SUPPLEMENTARY NOTATION
N/A

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
polygonal separation, convex polygon

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

In this paper we study the problem of polygonal separation in the plane, i.e., finding a convex polygon with minimum number k of sides separating two given finite point sets (k-separator), if it exists. We show that for $k=\Theta(n)$, $\Omega(n\log n)$ is a lower bound to the running time of any algorithm for this problem, and exhibit two algorithms of distinctly different flavors. The first relies on an $O(n\log n)$-time preprocessing task, which constructs the convex hull of the internal set and a nested star-shaped polygon determined by the external set; the k-separator is contained in the annulus between the boundaries of these two polygons and is constructed in additional linear time. The second algorithm adapts the prune-and-search approach, and constructs, in each iteration, one side of the separator; its running time is $O(nk)$, but the separator may have one more side than the minimum.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. □ DTIC USERS □

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE NUMBER *(Include Area Code)*

22c. OFFICE SYMBOL
N/A

Minimum Polygonal Separation[1]

H. Edelsbrunner [2], and F. P. Preparata [3]

Abstract:

In this paper we study the problem of polygonal separation in the plane, i.e., finding a convex polygon with minimum number k of sides separating two given finite point sets (k–separator), if it exists. We show that for $k=\Theta(n)$, $\Omega(n\log n)$ is a lower bound to the running time of any algorithm for this problem, and exhibit two algorithms of distinctly different flavors. The first relies on an $O(n\log n)$–time preprocessing task, which constructs the convex hull of the internal set and a nested star–shaped polygon determined by the external set; the k–separator is contained in the annulus between the boundaries of these two polygons and is constructed in additional linear time. The second algorithm adapts the prune–and–search approach, and constructs, in each iteration, one side of the separator; its running time is $O(nk)$, but the separator may have one more side than the minimum.

[1] Research of the first author is supported by Amoco Fnd. Fac. Dev. Comput. Sci. 1–6–44862; research of the second author is supported by NSF Grant ECS 84–10902.

[2] Department of Computer Science, University of Illinois at Urbana–Champaign, Urbana, Il 61801, USA.

[3] Coordinated Science Lab., Departments of Electrical & Computer Engineering and of Computer Science, University of Illinois at Urbana–Champaign, Urbana, Il 61801, USA.

1. Introduction

The separability of two finite sets of points in Euclidean space by means of a suitable separator of one less dimension is an interesting problem in a number of applications, typically in classification theory. Traditionally, the research interest has generally remained confined to linear separability [SW, MP, DK, D, M] or to spherical separability [OKM].

In this note we wish to extend the scope of these investigations as suggested in [BEHW]. Restricting ourselves to the Euclidean plane, we consider the set of separators represented by convex polygons. Note that if two finite sets of points are separated by a convex k–gon, k linear tests are sufficient to carry out the classification of a new sample point. We formalize this problem as follows:

> 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 (not necessarily disjoint) sets of points S_1 and S_2, construct a separating convex k–gon for the smallest possible integer k.

With this definition, linear separability becomes 1–gon separability. The solution of this problem implicitly solves the problem of determining k and the problem of deciding if there is a separating triangle. For this problem we exhibit an algorithm that runs in time $O(n\log n)$; this algorithm is optimal in the sense that for $k=\Theta(n)$, $\Omega(n\log n)$ is shown to be a lower bound to the running time.

For small k, it may be desirable to resort to a technique asymptotically superior to the preceding one. We exhibit one such algorithm to obtain an approximate solution of the given problem, which consists either of k or $k+1$ edges. The approximation is the price exacted by $O(kn)$ running time. The method is an adaptation of the approach proposed by Dyer [D] and Megiddo [M] to solve linear programming; we have been unable to formulate our problem in linear–programming terms, which suggests a perhaps inherently new application of the Dyer–Megiddo technique, called "prune–and–search" in [LP].

A related problem — the construction of a separating convex k–gon of two nested convex polygons, for minimum k — has been recently studied by Aggarwal et. al. [ABRSY]. We should point out that, in spite of the superficial similarity, the algorithmic techniques needed for the two problems appear to be inherently different.

This paper is organized as follows. In Section 2 we present the lower–bound argument and, in Section 3 we characterize the solution. In Section 4 we exhibit the main algorithm, with running time $O(n\log n)$. Finally Section 5 describes the approximation algorithm based on the prune-and-search approach. Some open problems are mentioned in Section 6.

2. Lower Bound

The lower bound argument is based on a linear-time transformation of sorting to "minimum polygonal separation".

Let $x_1,x_2,...,x_n$ be n real numbers, we wish to sort. We assume that n is even; otherwise, we add an arbitrary new number and remove it from the set after the sorting process.

The problem transformation is carried out as follows. We first construct the set of points $S_1=\{(x_i,x_i^2)$: $i=1,2,...,n\}$ (on the parabola $y=x^2$) and then let $S_2=S_1$. We then construct a minimum convex separator P of S_1 and S_2. Due to the definitions of S_1 and S_2, each point of S_1 belongs to an edge of P, or conversely, each edge of P intersects the parabola in two points of S_1. Therefore, by traversing the boundary of P in counterclockwise order beginning at the leftmost intersection of P and the parabola, and by computing the intersections of each edge with the parabola, in linear time we traverse the sequence of points of S_1 by increasing x_i, i.e., we retrieve the desired sorted sequence.

Since the transformation only takes time $O(n)$, the $\Omega(n\log n)$ lower bound for sorting becomes a lower bound for "minimum polygonal separation," and we have:

Theorem 2.1: The computation of the minimum polygonal separator of two sets of points S_1 and S_2 in the plane, with $\text{card}(S_1 \cup S_2)=n$, requires $\Omega(n\log n)$ operations, in the worst case.

3. Characterization of the Optimum Solution

The two sets of points S_1 and S_2 play asymmetric roles in the problem. Indeed, the k-gon referred to as the *separator* contains one set (internal), and the other set (external) belongs to the complement of the interior of the separator. We assume for the time being that the internal set has been determined. Let it be S_1.

Since any separator is a convex polygon, only the vertices of the convex hull of S_1 are relevant to the construction of the separator. Therefore let $\mathcal{C}_1$=conv(S_1), the convex hull of S_1.

For any line not intersecting the interior of $\mathcal{C}_1$ we call *positive* the open half-plane $h_+(l)$ containing the interior of $\mathcal{C}_1$, and negative the other, $h_-(l)$. Let p be an arbitrary point of S_2. If we trace from p the supporting lines l_1 and l_2 to $\mathcal{C}_1$, each of them defines two half-planes. The intersection $h_-(l_1) \cap h_-(l_2)$ is called the *remote wedge of* p, denoted $\mathcal{W}(p)$. We have

Lemma 3.1: For any $p \in S_2$ and any convex separator $\mathcal{P}$ of S_1 and S_2, $\mathcal{W}(p) \cap \mathcal{P} = \emptyset$.

Proof: Assume, for a contraction, that a point q in $\mathcal{W}(p)$ belongs to the separator. Since $\mathcal{W}(p)$ is defined as an open set, we can as well assume that q belongs to the interior of $\mathcal{P}$. Consider the straight line l passing by q and p, and let u be the intersection of l with the interior of $\mathcal{C}_1$. The segment u is contained in the interior of $\mathcal{P}$, but so is point q; since $\mathcal{P}$ is convex the entire segment conv($u \cup \{q\}$) is contained in the interior of $\mathcal{P}$, and therefore point $p \in S_2$ that lies on it (see Fig. 1). This contradicts the definition of separator. □

We can therefore define the region $\mathcal{F}$, of the plane whose interior must have void intersection with any convex separator of S_1 and S_2, that is:

$$\mathcal{F} = \bigcup_{p \in S_2} \mathcal{W}(p).$$

$\mathcal{F}$ is referred to as the *forbidden region* (see Fig. 2, for an illustration). The complement of $\mathcal{F}$, denoted $\mathcal{C}_2$, is a (possibly unbounded) star-shaped polygon, whose kernel [PS, p.18] contains $\mathcal{C}_1$. The nature of the boundary of $\mathcal{C}_2$ deserves some discussion. The reflex vertices of $\mathcal{C}_2$ are points of S_2, and no two reflex vertices are

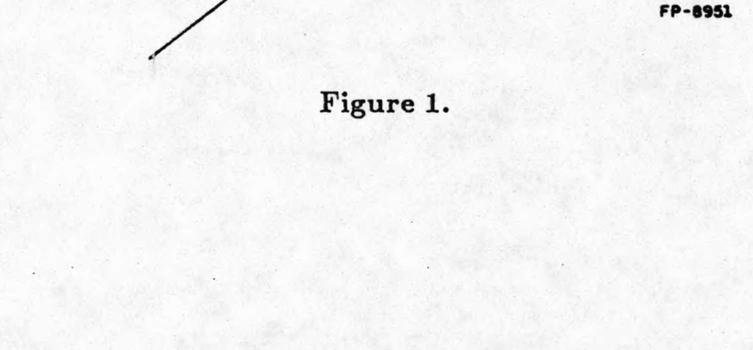


Figure 1.

adjacent. Edges incident to a reflex vertex are either bounded or unbounded. In the first case, the other extreme is a convex vertex of C_2, the intersection of the boundary of two adjacent remote wedges; in the second case, the other extreme is conventionally thought of at infinity. In both cases, the convex extreme of an edge is called a *niche*. Each edge of the boundary of C_2 is directed towards its reflex vertex extreme and called an *arc*. This orientation partitions the set of arcs into two equal-size subsets, called *clockwise set* (A_-) and *counterclockwise set* (A_+) defined as follows: an arc e belongs to A_- if a ray, sweeping the plane clockwise around a pole internal to the kernel of C_2, scans the points of e towards e's terminus. Set A_+ is defined with respect to a polar ray sweeping counterclockwise. The members of A_- are numbered in the order in which they are encountered by the sweeping ray; similarly for A_+. (Notice that this definition covers both the case when an arc of C_2 is bounded and the one when it is unbounded.)

For our purposes it is sufficient to consider only the counterclockwise set A_+. We extend an arc $e \in A_+$ beyond its terminus towards the interior of the star-shaped polygon up to the furthest intersection with C_2 if it exists, or to infinity, otherwise. This furthest intersection is where the extension leaves C_2, for the line which contains the arc intersects C_2 in a connected segment as it contains a point of the kernel of C_2. Notice that this intersection, if it exists, always occurs with another member of A_+. We call a thus constructed extension of an arc an *extended* arc, and assign to it the same direction as its defining arc. Fig. 2 shows the extensions of all counterclockwise arcs of C_2.

On the set of extended arcs we transfer the ordering relation of their corresponding arcs and naturally define the following predecessor/successor

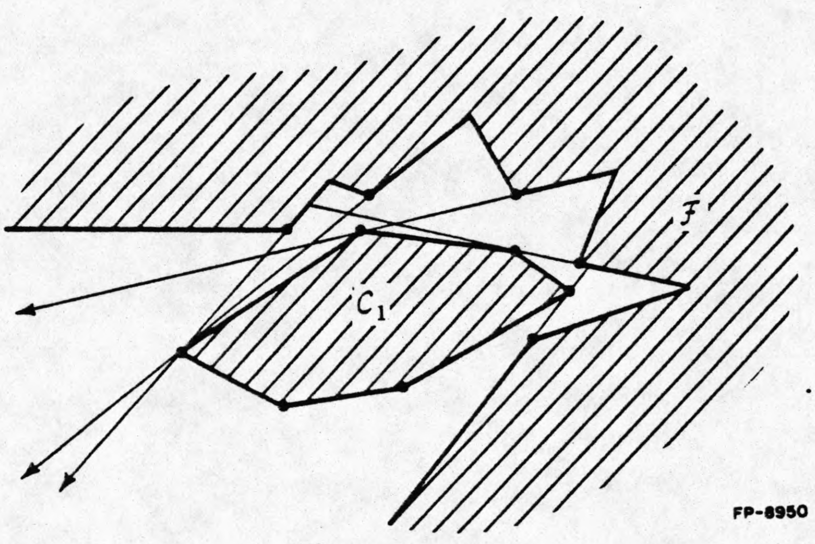


Figure 2.

relation:

> Two extended arcs e_1 and e_2 are in a *predecessor/successor relation* "$\longrightarrow$" (denoted $e_1 \longrightarrow e_2$) in either of these mutually exclusive cases: (i) if e_1 has a finite terminus which lies on e_2; (ii) if e_1 has no finite terminus, then e_2 has its niche at infinity, and, letting l_j be the line containing e_j (j=1,2), the region $h_+(l_1) \cap h_+(l_2)$ does not contain a connected component of $\mathcal{F}$.

Let t_1 and t_2 be two lines tangent to $\mathcal{C}_1$, and define the *wedge of* t_1 and t_2, denoted as $w(t_1,t_2)$, as the connected component of $(h_+(t_1) \cap h_+(t_2)) - \mathcal{C}_1$ that increases when line t_2 is rotated in counterclockwise direction. Note the non-symmetry of this definition. In fact, $w(t_2,t_1)$ is the other component such that

$$(h_+(t_1) \cap h_+(t_2)) - \mathcal{C}_1 = w(t_{1,} t_2) \cup w(t_2,t_1).$$

The significance of the predecessor/successor relation defined for the extended arcs of A_+ stems from the fact that $e_1 \longrightarrow e_2$ if and only if l_2, the line that supports e_2, is the unique line l which maximizes $w(l_1,l)$ under the constraint that it does not contain any point of S_2. We now demonstrate a crucial property of the solutions.

Lemma 3.2: If there is a k-separator of S_1 and S_2 with minimum k, then there is a k-separator each edge of which is contained in an extended arc of the counterclockwise set A_+ of $\mathcal{C}_2$.

Proof: Let $\mathcal{P}$ be a k-separator, with minimum k, having at least one edge e not contained in an extended arc. We now construct a new k-separator $\mathcal{P}'$ by a continuous transformation of $\mathcal{P}$:

1) If $e \cap \mathcal{C}_1 = \emptyset$, we translate e until it touches $\mathcal{C}_1$. The resulting polygon, which is contained in $\mathcal{P}$ (being the intersection of $\mathcal{P}$ with a half-plane) and contains $\mathcal{C}_1$ by construction, is a k-separator.

2) Let q be a point shared by e and $\mathcal{C}_1$. We rotate e in counterclockwise direction around q until it is contained in an extended arc or until it becomes aligned with an edge conv$\{q,q_1\}$ of $\mathcal{C}_1$. The resulting polygon $\mathcal{P}'$ is obtained by removing from $\mathcal{P}$ triangle $\mathcal{T}_1$ and by adding to it triangle $\mathcal{T}_2$ (see Fig. 3). Clearly, $\mathcal{T}_2$ contains no point of S_2 in its interior, otherwise, we would have passed an extended arc. If e belongs to no extended arc then it is aligned with the edge conv$\{q,q_1\}$, and we repeat the process with pivot in q_1.

By applying this construction to each edge of $\mathcal{P}$ not contained in an extended arc of $\mathcal{C}_2$, we obtain the desired result. □

The preceding lemma shows that the minimal separator may be sought in the (finite) set of convex polygons embedded in the union of the extended

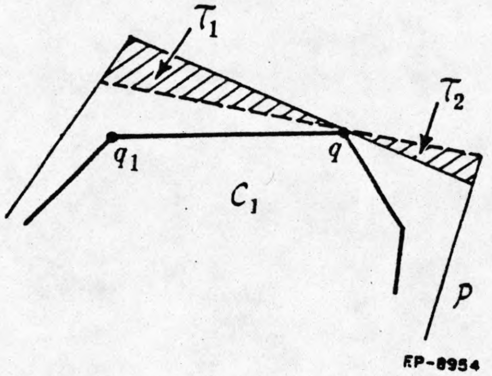


Figure 3.

arcs. We further reduce the set of possible candidates to the set of "greedy separators", obtained as follows.

If r is the number of the reflex vertices of C_2, there are r counterclockwise extended arcs. Number them $e_1, e_2, \ldots, e_r$, in the order previously defined. Select an extended arc, e_{i_1}, as *initial* arc and construct the sequence $e_{i_1}, e_{i_2}, e_{i_3}, \ldots$, where e_{i_j} and $e_{i_{j+1}}$ are a predecessor/successor pair. $(e_{i_1}, e_{i_2}, \ldots, e_{i_k})$ is a *cycle* if k is the smallest integer such that e_{i_1} and e_{i_k} intersect; this cycle identifies a k-separator, whose conventional first vertex is the intersection of e_{i_1} and e_{i_k}, and whose j-th vertex is the terminus of e_{i_j}, for $1 \leq j \leq k-1$. Due to the mechanism of the construction, we refer to this separator as "greedy"; clearly, there are only $r = O(\text{card}(S_2))$ greedy separators, and this set contains the minimal separator. By virtue of the following property, only a subset of this set needs to be inspected.

Lemma 3.3: There is an integer k such that each greedy separator has either k or $k+1$ edges.

Proof: The predecessor/successor relation "$\rightarrow$" on the set of extended arcs can be viewed as a function ϕ on the indices of the (ordered) set of extended arcs. Specifically, $\phi(i) = j$ if and only if $e_i \rightarrow e_j$.

Let e_i, $e_{i'}$, and $e_{\psi(i)}$ be respectively the initial, second, and last extended arc used in the construction of a greedy separator. Then, since $e_{\psi(i)}$ intersects e_i, we have that $i \leq \phi(\psi(i)) \leq \psi(i)$, $\phi(\psi(i)) \leq \psi(i) \leq i$, or $\psi(i) \leq i \leq \phi(\psi(i))$, depending on where we started indexing the extended arcs. This is illustrated in Fig. 4, where a greedy separator starting at i is shown as a path ending at $\phi(\psi(i))$. The greedy separator defines a natural partition of the extended arcs into intervals $[e_i, e_{i+1}, \ldots, e_{i'-1}]$, $[e_{i'}, e_{i'+1}, \ldots, e_{i''-1}]$, etc., where $e_{i''}$ is the third extended arc of the greedy separator. It is easy to recognize that the solid pointers of two paths corresponding to distinct greedy separators do not intersect, except possibly at their destinations. This shows that each greedy separator must use an extended arc in the interval $[e_i, e_{i+1}, \ldots, e_{i'-1}]$, and for that matter in any analogous interval. The fact that the "paths" corresponding to the r distinct greedy separators are interleaved implies that two greedy separators with initial extended arc in $[e_i, e_{i+1}, \ldots, e_{i'-1}]$ have numbers of arcs differing by at most one. □

By the same reasoning as in the above proof, we can show that a greedy separator using a fixed extended arc e^* has the same number of edges as the one having e^* as initial arc. It follows that it is sufficient to construct only the greedy separators whose initial extended arc is a member of $[e_i, e_{i+1}, \ldots, e_{i'-1}]$ or of another interval of the greedy separator defined by e_i. If the minimum member of edges of the separator is k, by the pigeonhole principle there is an interval with at most $\lfloor n/k \rfloor$ members.

4. A Simplified Algorithm to Construct a Separator

Two sets S_1 and S_2 of n_1 and n_2 points in the plane are given. Our first task is to decide the respective roles of the two sets, i.e., which of them is the internal set. The condition to be verified is that no point of the external set belongs to the interior of the convex hull of the internal set. Therefore, we construct the convex hull C_1 of S_1 and test whether each point of S_2 is outside the interior of C_1. If the test passes, then S_1 and S_2 are respectively internal and external. If it fails, we try again with reversed roles; if it fails again, no convex separator exists. This initial test is carried out in time $O((n_1+n_2)\log(n_1+n_2))$. Without loss of generality, we assume that S_1 and S_2 are polygon–separable and let S_1 be the internal set. After this initial test, our task consists of the following subtasks:

1. Construct the forbidden region $\mathcal{F}$.

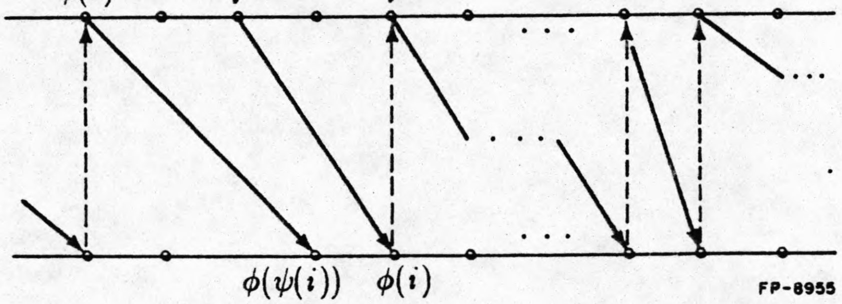


Figure 4.

2. Construct a greedy separator.

3. On the basis of the obtained greedy separator, select an interval I of arcs, and, for each arc e in I, construct the greedy separator having e as its initial arc and select among these separators an optimal one.

We now consider these three subtasks in detail.

1. For each $p \in S_2$ we construct $\mathcal{W}(p)$. If we arrange the vertices of $\mathcal{C}_1$ as a linear array, the two supporting lines of a point p to $\mathcal{C}_1$ can be determined in time $O(\log n_1)$ (see [PS]). Thus in time $O(n_2 \log n_1)$ the set $\{\mathcal{W}(p) | p \in S_2\}$ is available.

Next, we define the *left supporting line* $l(p)$ *of* a point $p \in \mathcal{P}_2$ as the line through p and tangent to $\mathcal{C}_1$ directed from p to the contact point on the boundary of $\mathcal{C}_1$ such that $\mathcal{C}_1$ lies to the right of $l(p)$ (see Fig. 5). Analogously, we define the *right supporting line* $r(p)$ *of* point p. By the *angle of* a directed line we mean the angle through which the positive x-axis has to be rotated before it is parallel and equally directed as the directed line. We order the points of S_2 in increasing angle of their left supporting lines. The vertices of $\mathcal{F}$ are clearly a not necessarily connected subsequence of the just constructed sequence, and are obtained by a scan of the sequence. The initial step consists of selecting the first point $p \in S_2$. At a generic step, we assume that the currently found subsequence is stored in a sequential list L and let p be the current point. We consider the remote wedge $\mathcal{W}(p)$ of p and scan L backwards until a point is found that lies outside the closure of $\mathcal{W}(p)$, and eliminate all points scanned before. This generic step is performed for each point of S_2 in turn. In the final step, we perform a generic step for the first point in the constructed list. The correctness of the method is provided by the following lemma.

Lemma 4.1: Let p_1 and p_2 be two points in the current list, ordered by

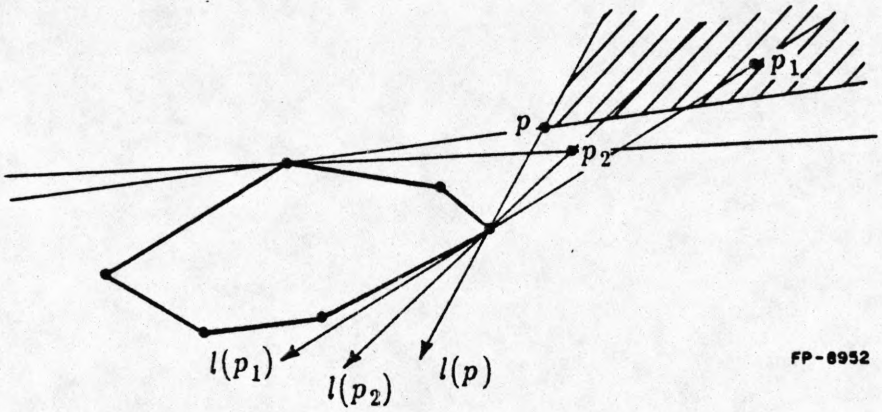


Figure 5.

increasing angle of their left supporting lines to $\mathcal{C}_1$, and let p be a new point. Then p_1 is contained in $\mathcal{W}(p)$ only if point p_2 is contained in $\mathcal{W}(p)$.

Proof: (Refer to Fig. 5.) Due to the convexity of $\mathcal{C}_1$ and to the chosen order on the set S_2, the intersections u_j of $r(p)$ with lines $l(p_j)$ $(j=1,2)$ are such that u_2 is between p and u_1. Now, assume for a contradiction that $p_1 \in \mathcal{W}(p)$ and $p_2 \notin \mathcal{W}(p)$. This implies that p_1 belongs to $h_-(r(p_2))$, and $p_1 \in \mathcal{W}(p_2)$ since $p_1 \in h_-(l(p_2))$ as noted above. This is a contradiction, because p_1 belongs by hypothesis to the current list. □

It is evident that the present subtask (very akin to the Graham scan for the convex hull) runs in time $O(n_2 \log n_2)$ for constructing the initial order, plus $O(n_2)$ time to actually construct $\mathcal{F}$.

2. $\mathcal{F}$ is available as the (counterclockwise) sequence of its reflex vertices. ($\mathcal{F}$ may consist of several disjoint connected components.) From this, we can construct in linear time the ordered sequence of the arcs in A_+ and arrange them in a linear list L_1.

The next step is the construction of the extended arcs and simultaneously, of the predecessor/successor relation on this set. In the initial step we arbitrarily select an arc $e \in L_1$, and denote by l the line containing e. We then scan L_1 starting from e, as long as the arc e' currently scanned forms an angle smaller than π with l we test for intersection of l with e'; if an intersection is found, the extended arc associated with e' is the successor of the extended arc associated with e. If no arc e' intersecting l is found, then the extended arc associated with the first arc that forms an angle larger or equal to π with l is the successor of the extended are of e.

After this initial construction, we establish two pointers, one at e and the other to e'. By the construction of $\mathcal{F}$ distinct predecessor/successor pairs are interleaved, so that as we step forward the predecessor pointer, the successor pointer cannot regress and the construction is therefore completed in linear time.

At this point, on the set A_+ we have a cyclic order and the relation "$\rightarrow$". To construct a greedy separator we proceed as follows. Select an arbitrary $e \in A_+$, and let $e_0 := e$. Construct a sequence $e_0, e_1, e_2, \ldots, e_s$ such that $e_i \rightarrow e_{i+1}$ $(i=0,1,..,s-1)$ and $e_{s-1} < e_0 \le e_s$ in the cyclic order. Then the polygon whose vertices are the intersections between consecutive extended arcs is a greedy separator. This construction is clearly completed in time $O(s)$.

3. The separator obtained above partitions the cyclic order of arcs in A_+ into disjoint intervals. If k is the size of the minimum separator, then either $s=k$ or $s=k+1$; in any case, there is one of these intervals which contains at

most n_2/k arcs. Let this be the set $\mathcal{A}$. Finally, we perform the greedy separator construction for each arc $e \in \mathcal{A}$. This subtask to completed in time $O(\frac{n_2}{k}k+1)=O(n_2)$. We conclude therefore with the following result:

Theorem 4.2: Given two finite sets S_1 and S_2 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 $O((n_1+n_2)\log(n_1+n_2))$ and this is optimal.

5. Constructing a Near–Optimal Separation

We have seen in Section 3 that a greedy construction which starts with an arbitrary extended arc of C_2 yields either a separating k–gon or $(k+1)$–gon, for minimum k. We will show that such a greedy construction can be performed algorithmically in $O(n)$ time per edge of the separator, where $n=n_1+n_2$ and $n_i=\text{card}(S_i)$, for $i=1,2$. In this construction, we do not assume that $C_1=\text{conv}(S_1)$ or C_2, the complement of the union of all remote wedges, are available.

The global construction is exactly the greedy construction outlined in Section 3. Initially, we determine an arbitrary line l_1 which contains an extended arc of C_2. Recall that extended arcs are now no longer available as a precomputed set, so we determine line l_1 from an arbitrary line l_0 supporting C_1 by a so–called general step described below. Let l_1 contain the first edge of the separator P. In a general step, we are given a sequence of lines $l_1,l_2,\ldots,l_j$ which contain the first j edges of P in this sequence. Each line l_i, $1 \leq i \leq j$, contains an extended arc e_{m_i}, and it is directed as e_{m_i}, that is, C_1 is to the left of l_i. Furthermore, the lines are such that $m_{i+1}=\phi(m_i)$. In one general step, we determine line l_{j+1}, which is the unique line that contains the extended arc $e_{\phi(m_j)}$. The general step is executed until l_{j+1} intersects e_{m_1}.

Below, we describe how the $(j+1)^{\text{st}}$ line l_{j+1} can be determined in $O(n)$ time. For convenience, we assume that l_j is vertical and downward directed (see Fig. 6). Let l be another directed line supporting C_1 such that C_1 is to its left. We define the *angle* $\alpha(l)$ *of* l as the angle through which l_j has to be rotated before it is parallel to l and equally directed.

As in Section 3, we define $w(l_j,l)$ as the connected component of

$$(h_+(l_j)\cap h_+(l))-C_1$$

whose area increases when l is rotated counterclockwise (see Fig. 6). Our objective is to find line l_{j+1}, which is the line l such that $w(l_j,l)$ is largest and contains no points of S_2. However, it is not enough to guarantee that all regions $w(l_i,l_{i+1})$ are empty; there is also the possibility that a point of S_2 belongs to the interior of the convex hull of S_1. To catch these cases, we let $\square(l_j,l)$ be the quadrilateral defined as follows:

> let c be an arbitrary but fixed point in the interior of C_1; $\square(l_j,l)$ is the quadrilateral defined by l_j, l, and the segments that connect c with the points where lines l_j and l touch the boundary of C_1.

For convenience, we let $\square(l_j,l)$ include the two bounding segments but not the pieces of its boundary that belongs to line l_j or l. Note that $\square(l_j,l)$ contains $w(l_j,l)$ which implies that $w(l_j,l)$ contains no point of S_2 if $\square(l_j,l)$ does so.

In our algorithm, we assume that $\square(l_j,l_{j+1})$ is bounded, which implies that $\alpha(l_{j+1})<\pi$. It is rather easy to decide when this is not the case: determine the line $\hat{l}_j$ with $\alpha(\hat{l}_j)=\pi$, that is, $\hat{l}_j$ is parallel to l_j and supports to C_1, and determine whether $\square(l_j,\hat{l}_j)$ is empty. If it is, then l_{j+1} is either the line with the largest angle which separates S_1 and $S_2\cap h_-(\hat{l}_j)$, or it is the line that we get when we replace l_j by $\hat{l}_j$, whichever has smaller angle. The separating line with largest angle can be found in O(n) time using linear programming, or by a straightforward modification of the general step described below.

To determine line l_{j+1}, we use a novel algorithmic paradigm due to Megiddo [M] and Dyer [D], called prune-and-search in [LP]. The central idea of this technique is to find a constant fraction of the data points to be redundant, and to recur for the remaining points. If this constant fraction can be determined in linear time, then the time-complexity $T(n)$ of the whole algorithm follows the

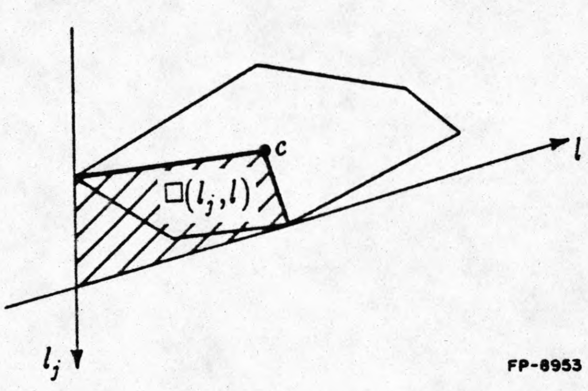


Figure 6.

recurrence relation

$$T(n)=T(cn)+O(n),$$

for some real number $0<c<1$. Therefore, $T(n)=O(n)$.

The prune-and-search algorithm combines several subtasks which are:

(i) *Determine an angle.* Select an angle α that a trial-line forms with the reference line l_j.

(ii) *Test an angle.* Determine whether l_{j+1} forms with l_j an angle smaller than, equal to, or larger than α selected in (i).

(iii) *Detect redundant points.* Given a trial-line, eliminate redundant points from S_1 and S_2.

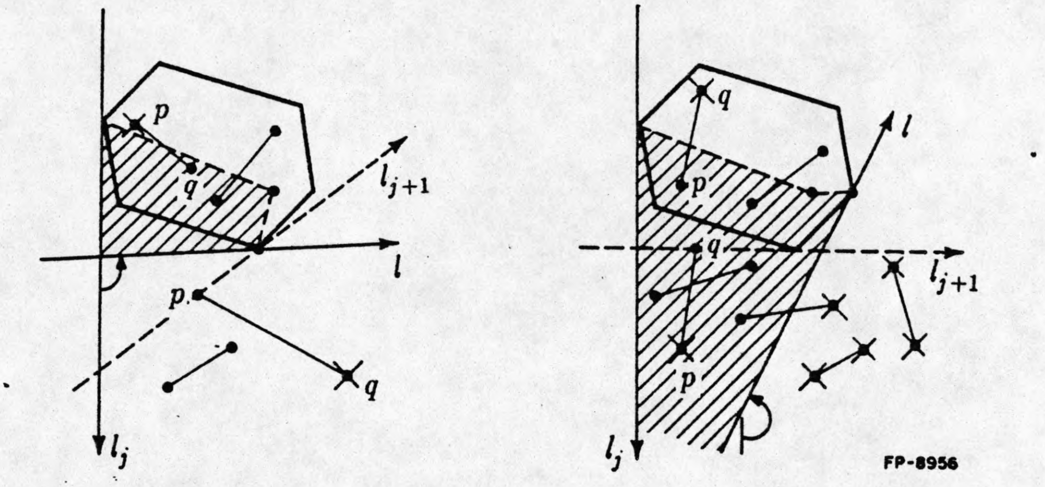
We will discuss the subtasks in the reverse order and will then put the pieces together to get a linear-time algorithm for finding l_{j+1}, if it exists.

Detect redundant points: Here we consider two cases. In the first case, we assume that the angle of the trial-line l is smaller than the angle of l_{j+1} (see Fig. 7(a)), in the second case, we assume the opposite (see Fig. 7(b)). For convenience, we assume that no two points lie on a common vertical line; if the x-coordinate of a point p is smaller than the one of a point q then we say that p is *to the left of* q. All arguments will concern pairs of points conveniently joined by segments, and their *angles*, which are the *angles* of their containing lines directed from left to right. Each pair will either have both points in S_1 or both points in S_2.

First, we assume $\alpha(l)<\alpha(l_{j+1})$, and we let $\{p,q\}$ be a pair of points with angle smaller or equal to $\alpha(l)$. If p is to the left of q and both belong to set S_1, then p is redundant since no line through p with angle larger than $\alpha(l)$ is tangent to the convex hull of S_1. If p and q belong to S_2, then q is redundant, since q is in $\square(l_j,l')$ only if p does, for every line l' through q such that $\alpha(l')>\alpha(l)$ (Fig. 7(a)).

Second, we assume $\alpha(l)>\alpha(l_{j+1})$, and we let $\{p,q\}$ be a pair with angle greater or equal to $\alpha(l)$. Again, let p be to the left of q. By the same reasoning as above, we know that q is redundant, if p and q belong to S_1, and that p is redundant, if p and q belong to S_2. Furthermore, all points of S_2 outside $\square(l_j,l)$ are redundant, since $\square(l_j,l)$ contains $\square(l_j,l_{j+1})$ (see Fig. 7(b)).

Test an angle: To test a given angle α, we construct the line l with $\alpha(l)=\alpha$ which supports the convex hull of S_1. Obviously, this can be done in $O(n_1)$ time. Next, we test whether or not $\square(l_j,l)$ contains points of S_2 which takes $O(n_2)$ time. If this quadrilateral contains at least one point of S_2 then α is too large and has to



(a) (b)

Figure 7.

be decreased; it is even possible that a point of S_2 belongs to the interior of $\mathcal{C}_1$. Otherwise, there are two cases to consider. If there is a point of S_2 on the edge of $\square(l_j,l)$ contained in l, then we are finished, that is, $l=l_{j+1}$; otherwise, α is too small and has to be increased.

Determine an angle: The angle α is used for a binary search like strategy which narrows, step by step, the interval of possible angles. The only problem with this approach is that the set of possible angles is not discrete. To overcome this difficulty, we choose the angles such that, with each tested angle, there are some points found to be redundant. The search is now finite since we can eliminate only a finite number of points. In order to obtain a search which takes time $O(n)$, we choose an angle which allows us to eliminate at least $(\text{card}(S_1)+\text{card}(S_2)-2)/4$ points where S_1 and S_2 are the current sets which contain the not yet eliminated points. This is done as follows: in a first step, construct an arbitrary pairing of points of S_1 and separately, of S_2. Each pair determines a segment. Consider the angles formed by these segments with the vertical line, and find the pair with median angle (in time $O(\text{card}(S_1)+\text{card}(S_2))$ using a linear time median finding algorithm). The angle of this segment is the sought angle α.

Below, we give a more formal description of the algorithm which finds the line l_{j+1}, if it exists; otherwise, it reports that there is no convex separation. Its input is the line l_j, which is assumed to be vertical, and the sets S_1 and S_2. We also assume that there are no points to the left of l_j; otherwise, we remove points of S_2 that violate this condition. Note that this does not influence the construction.

```
Algorithm (Find next edge):
    if card($S_1$)=card($S_2$)=1 then
        The line through the only point in $S_1$ and the only point in $S_2$ is $l_{j+1}$. If $\alpha(l_{j+1})$ does not belong to the interval of angles determined during earlier iterations of the algorithm, then $S_1$ and $S_2$ are not separable by a convex polygon such that $S_1$ is interior and $S_2$ is exterior. Otherwise, $l_{j+1}$ contains the $(j+1)^{st}$ edge of the separator to be constructed.
    else
        Step 1: Determine an angle $\alpha$ as described above.
        Step 2: Decide whether $\alpha=\alpha(l_{j+1})$, in which case we halt,
            $\alpha<\alpha(l_{j+1})$, or $\alpha>\alpha(l_{j+1})$.
        Step 3: Eliminate the redundant points of $S_1$ and $S_2$ using the
            observations described above.
    endif
```

The time-complexity of the algorithm is linear in n_1+n_2 because Step 1 guarantees that at least half of the segments formed by pairs of points have angle greater or equal to the chosen α, and that at least half of the pairs have an angle less or equal to α. At least one point of each pair in either collection is eliminated, which implies that at least (card($S_1 \cup S_2$)-2)/4 points are removed. (The "-2" gets into effect when both card(S_1) and card(S_2) are odd.) This implies the main result of this section.

Theorem 5.1: Let S_1 and S_2 be two sets with a total of n points in the plane. If k is the smallest integer such that there is a convex k-gon that contains S_1 and whose interior avoids S_2, then the above algorithm constructs a separating k- or $(k+1)$-gon in $O(kn)$ time. If no such separator exists then the algorithm reports this in $O(n^2)$ time.

6. Discussion

This paper presents two algorithms for constructing a convex polygon with the fewest edges that separates two sets of a total of n points in the plane, if it exists. The first algorithm takes $O(n\log n)$ time, and this is optimal in the worst case if $k=\Theta(n)$. The second algorithm takes $O(kn)$ time for construct a separating convex k-gon, where k is either optimal or one larger. These results raise a few interesting open problems:

1. Is $\Omega(n\log n)$ a lower bound for the construction of a separating convex k-gon, for smallest k, even if k is small? More specifically, is $\Omega(n\log n)$ time required

to decide whether or not there exists a separating triangle?

2. Is it possible to refine our O(kn) time algorithm so that it finds a separating k-gon in O($n\log k$) time, with k equal to the minimum or one larger?

3. Finally, can the presented techniques be extended to three dimensions?

7. References

[ABOSY] A. Aggarwal, H. Booth, J. O'Rourke, S. Suri, and C. K. Yap, Finding minimal convex nested polygons, In Proc. 1st ACM Sympos. Comput. Geom., Baltimore, MD, 1985, 296–304.

[BEHW] A. Blumer, A. Ehrenfeucht, D. Haussler and M. Warmuth, Classifying learnable geometric concepts with the Vapnik–Chervonenkis dimension. In Proc. 18th ACM Sympos. Theory Comput., Berkeley, CA, 1986.

[DK] D. P. Dobkin and D. G. Kirkpatrick, A linear algorithm for determining the separation of convex polyhedra, Manuscript, 1983.

[D] M. E. Dyer, Linear time algorithms for two- and three-variable linear programs, *SIAM J. Comput.* **13** (1984), 31–45.

[LP] D. T. Lee and F. P. Preparata, A survey of computational geometry, *IEEE Trans. Comput.* **C–33** (1984), 1072-1101.

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

[MP] D. E. Muller and F. P. Preparata, Finding the intersection of two convex polyhedra, *Theoret. Comput. Sci.* **7**, (1978) 217–236.

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

[PS] F. P. Preparata and M. I. Shamos, *Computational Geometry*, Springer–Verlag, New York, 1985.

[SW] J. Stoer and C. Witzgall, *Convexity and Optimization in Finite Dimensions.* Springer–Verlag, New York, 1970.