MINIMUM DECOMPOSITIONS OF POLYGONAL OBJECTS

J. Mark Keil* and Jorg-R. Sack**

SCS-TR-42

March 1984

* J. Mark Keil
Department of Computational Science
University of Saskatchewan
Saskatoon, Saskatchewan
S7N OWO

** Jorg-R. Sack
School of Computer Science
Carleton University
Ottawa, Ontario
K1S 5B6

MINIMUM DECOMPOSITIONS OF POLYGONAL OBJECTS

J. Mark Keil* and Jorg-R. Sack**

ABSTRACT

Geometrical problems occur frequently in such areas as pattern recognition, image processing, computer graphics and VLSI.
Given a geometrical problem an efficient solution is often dependent on the nature of the objects we are dealing with. Objects are often described by polygons. The usual strategy for solving many of these problems on general polygons is to decompose the polygon into simple component parts, solve the problem on each component using a specialized algorithm, and then combine the partial solutions.
We survey the existing literature on minimum number and minimum edge-length decompositions.
We also give an $O(n^4)$ algorithm to decompose a rectilinear polygon into the minimum numbers of convex quadrilaterals.

* Department of Computational Science, University of Saskatchewan, Saskatoon, Saskatchewan

** Jorg-R. Sack
School of Computer Science, Carleton University,
Ottawa, Ontario

Minimum Decompositions of Polygonal Objects

J. Mark KEIL

Department of Computational Science,
University of Saskatchewan,
Saskatoon,Canada, S7N 0W0

Jörg-R. SACK

School of Computer Science,
Carleton University,
Ottawa, Canada, K1S 5B6

1. Introduction

In artificial intelligence, image processing, and pattern recognition, objects are frequently described by polygons. These polygons and their properties are in the center of attention of the rapidly growing field of computational geometry. Using computational geometry techniques one is often able to obtain elegant and efficient algorithms to solve certain geometrical problems. In some cases lower bounds have been derived giving us ideas about the intrinsic complexity of problems. Geometrical problems occur frequently in such areas as pattern recognition, image processing, computer graphics and VLSI. The large quantities of data produced by todays devices require algorithms handling these data to be designed efficiently.

Given a geometrical problem an efficient solution is often dependent on the nature of the objects we are dealing with. We illustrate this point by considering the problem of determining the convex hull of an arbitrary set of points versus that of a simple polygon connecting these points. A simple polygon is *convex* if all its internal angles are less than $180°$ as in figure 1.1. The *convex hull* of a set points is defined as the minimum area convex polygon enclosing the points. The vertices of the convex hull are points of the set. The convex hull of a polygon is defined analogously. The convex hull of a set of n points can be determined in O(nlogn) time and this is optimal [Gra 72,Sha 78,Yao 79]. However, linear time algorithms for computing the convex hull of a simple polygon exist [MA 79,Lee 80,Sac 83,GY 84].

If the objects are given as polygons their specific polygonal properties may also lead to more efficient solutions. Before proceeding we define several such properties. Refer to figure 1.2. A *polygon* P is specified by its vertices $(p_1,\ldots,p_n)$ listed in clockwise order. A *polygonal chain* $C_{i,j}$ is a sequence of consecutive

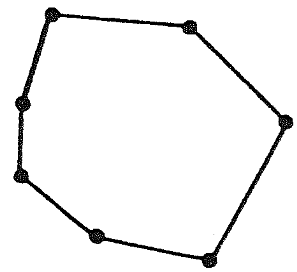


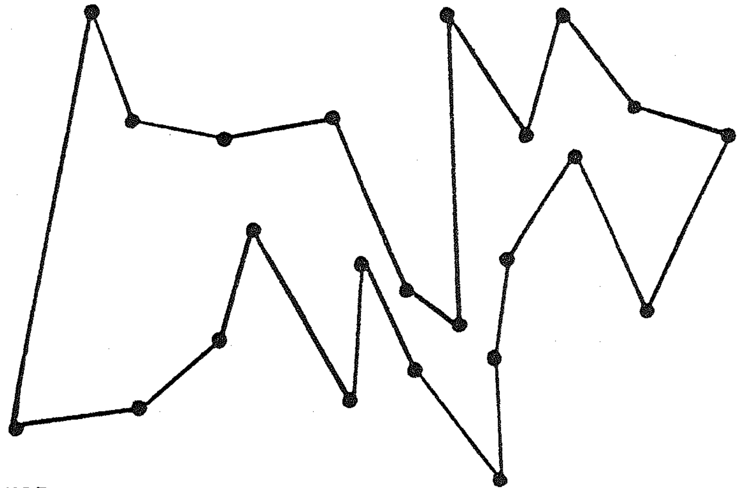
Figure 1.1 A Convex Polygon

vertices $p_i,\ldots,p_j$ of P. A polygonal chain is *monotone* with respect to a line l if the projections of p_k, k=i,...,j on l are ordered in exactly the same way as the vertices in $C_{i,j}$. A polygon is *monotone* if there exists a line l such that the boundary of P can be partitioned into two polygonal chains $C_{i,j}$ and $C_{j,i}$ which are monotone with respect to l. A point x in P is *visible* from a point y in P, if the open line segment joining x and y lies entirely inside P. A polygon P is *star-shaped* if there exists at least one point x inside P from which the entire polygon is visible. A polygon is *weakly visible* from an edge pq of P if for every point x in P there exists a point y in pq that is visible from x. A polygon is *edge-visible* if there exists an edge from which the polygon is weakly visible. A *spiral polygon* is a simple polygon whose boundary chain contains precisely one concave subchain.

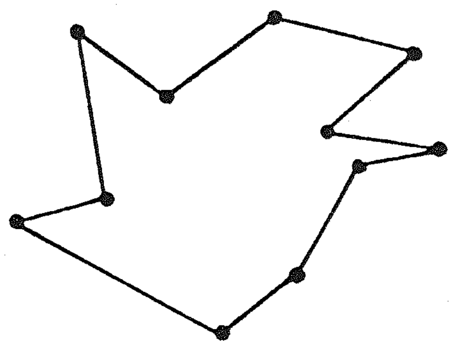
It appears that algorithms for such structured polygons are easier to write, thus easier to prove correct, and often exhibit lower space/time complexities. For example, the *triangulation problem* is to partition an n-vertex polygon into non-overlapping triangles. Linear-time algorithms for triangulating star-shaped, monotone, edge-visible and convex polygons exist [GJPT 78,SvL 80,EAT 81]. An arbitrary simple polygon can be triangulated in O(nlog(n)) time [GJPT 78,Cha 82,CI 83]. However, this has not yet been shown to be optimal, i.e. no non-trivial lower bound on this problem exists. Other examples of problems for which fast algorithms exist on restricted polygons include point inclusion in convex or star-shaped polygons [Sha 78], medial axis of a convex polygon [Pre 77], and intersections involving convex or star-shaped polygons [Cha 80,MF 82].

The usual strategy for solving many of these problems on general polygons is to decompose the polygon into simple component parts, solve the problem on each component using a specialized algorithm, and then combine the partial solutions [GJPT 78,SvL 80,ACYB 80,MF 80]. This strategy of decomposing polygonal objects into simpler components is one motivation for the development of efficient *decomposition techniques.*

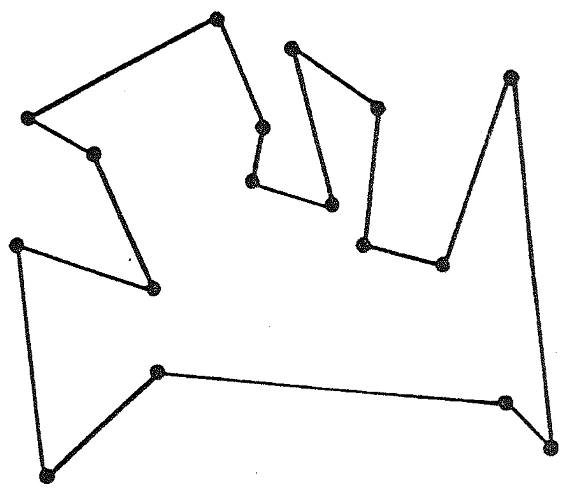
These decomposition techniques have also received considerable attention in pattern recognition and image analysis [FP 75,Pav 77]. Pattern recognition



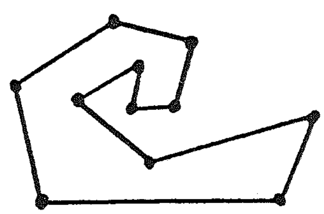
A Monotone Polygon



A Star-shaped Polygon



An Edge-Visible Polygon



A Spiral Polygon

Figure 1.2

extracts information from an object in order to identify or classify it. A usual strategy for recognizing a general polygonal object is to decompose it into simpler components, then identify the components and their interrelationships and use this information to determine the shape of the object [Pav 77]. The types of simpler components that have been considered include triangles[AAO 83], rectangles [FSS 80],trapezoids [AA 83], convex polygons [FP 75,Sch 78], spiral polygons [FP 75], star-shaped polygons [AT 81] and monotone polygons [Tou 80].

We can distinguish between two kinds of decomposition techniques: procedure-directed and component-directed. In a *procedure-directed* decomposition technique, a procedure is specified that decomposes the given object into smaller components. The specific nature of the components may vary. In a *component-directed* approach the object is decomposed into well-defined smaller components. Component-directed techniques are the most common and we will consider only them in this paper.

A decomposition is called a *partitioning*, if the object is decomposed into non-overlapping pieces. Algorithms to partition arbitrary simple polygons into convex, star-shaped, monotone, edge-visible components have been developed. If overlapping pieces are allowed we call the decomposition a *covering*.

Decomposing a polygonal object into simpler component parts can be done with or without introducing additional vertices which are commonly called *Steiner points*. Whether or not Steiner points are allowed often makes a big difference in the solution of a problem. Figure 1.3 shows decompositions of simple polygons into the minimum number of convex polygons with and without Steiner points.

If the interior angle at a vertex is reflex the vertex is called a *notch*. Experimental observation from graphics [NS 79] and pattern recognition [Pav 68] shows that, in practice, the number of notches in a polygon is much smaller than the number of vertices. We let N denote the number of notches in a polygon and we describe and analyze decomposition algorithms with respect to both n and N.

In this paper we are primarily concerned with techniques which provide decompositions which are minimal in some sense. We may be interested in decomposing a polygonal object into the minimum number of some component type. In some applications a decomposition that minimizes the total length of the internal edges used to form the decomposition is useful. We measure length in the usual Euclidean metric until section 4 when for technical reasons we switch to the discretized Euclidean metric. A minimum edge length decomposition can be quite different from a minimum number decomposition for the same component type.

Polygonal objects can be further classified according to certain other characteristics. A *rectilinear polygon* is a simple polygon whose edges are parallel to one of two given orthogonal directions. Section 2 considers decompositions of

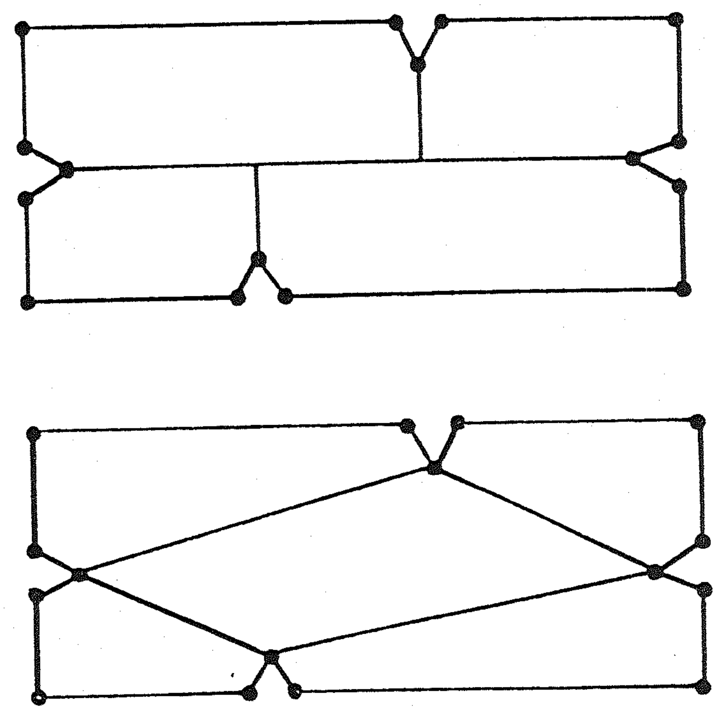


Figure 1.3 Convex Decompositions With and Without Steiner Points

rectilinear polygons. We may encounter polygons that contain holes. Holes are nonoverlapping "islands", simple polygons, inside the main polygon. The complexity of a decomposition problem usually increases if we allow the polygon to contain holes. We consider decompositions of polygons with holes in section 4.

2. Decomposition of Rectilinear Polygons

In many applications the objects encountered are rectilinear polygons. In image processing the boundaries of objects are often stored on a grid which usually implies that digitized images are rectilinear polygons. VLSI designs are also often stored on a grid and typically contain many rectilinear polygons. In this section we consider decompositions of rectilinear polygons, first when Steiner points are allowed and then when Steiner points are disallowed.

2.1. Decompositions allowing Steiner Points

Rectangles are the natural component type when decomposing rectilinear polygons allowing Steiner points. Ferrari, Sankar and Sklansky [FSS 81] have developed an algorithm for decomposing a rectilinear polygon into the minimum number of rectangles. Their algorithm is based on the following theorem.

Theorem

A rectilinear polygon can be partitioned into N - L + 1 rectangles, where N is the number of notches and L is the maximum number of nonintersecting chords that can be drawn either vertically or horizontally between two notches.

The theorem allows them to reduce the decomposition problem to that of finding the maximum number of independent vertices in the intersection graph of the vertical or horizontal chords between notches. The graph's matching properties are then used to solve the decomposition problem in $O(N^{5/2})$ time. Lipski [Lip 79] relates this decomposition problem to that of finding a Manhatten path and is able to develop an $O(n^{3/2}$ lognloglogn) time solution.

Lingas et al [LPRS 81] mention process control (stock cutting), automatic layout systems for VLSI (channel definition) and architecture (partitioning into offices) as applications for the problem of decomposing a rectilinear polygon into rectangles while minimizing the total length of the edges used to form the decomposition. To solve this minimum "ink" problem they proceed by inducing a grid over the polygon by extending the edges of the polygon into lines. They then build up partial solutions and use dynamic programming to achieve an $O(n^4)$ time algorithm. If the rectilinear polygon is restricted to the shape of a histogram they are able to provide an $O(n^3)$ time algorithm.

2.2. Decompositions without Steiner Points

Without Steiner points quadrilaterals rather than rectangles become the natural component type for the decomposition of rectilinear polygons. Even though arbitrary simple polygons do not necessarily admit decomposition into convex quadrilaterals Kahn,Klawe and Kleitman [KKK 80], in what they refer to as *quadrilaterization*, proved that it is always possible to decompose a rectilinear polygon into convex quadrilaterals without using Steiner points. In the remainder of this section quadrilaterizations are understood to be convex. Sack [Sac 82] has recently developed an O(nlogn) time algorithm for quadrilaterizing an arbitary simple rectilinear polygon. To find a quadrilaterization of an arbitrary simple polygon he begins by partitioning the rectilinear polygon into specific monotone polygons. See figure 2.1. Although arbitrary monotone polygons do not admit quadrilaterizations, these specific monotone polygons do. These monotone polygons, like arbitary monotone rectilinear polygons, starshaped and edge-visible rectilinear polygons, can be quadrilaterized in linear time [ST 81]. It is an open problem whether an arbitrary rectilinear polygon can be quadrilaterized in less than O(nlogn) time. Since any quadrilaterization of a rectilinear polygon results in the same number of quadrilaterals it is not meaningful to speak of a decomposition into the minimum number of quadrilaterals.

We instead turn to our other minimality criterion and present an $O(n^4)$ time algorithm for finding the minimum edge length quadrilaterization of a rectilinear

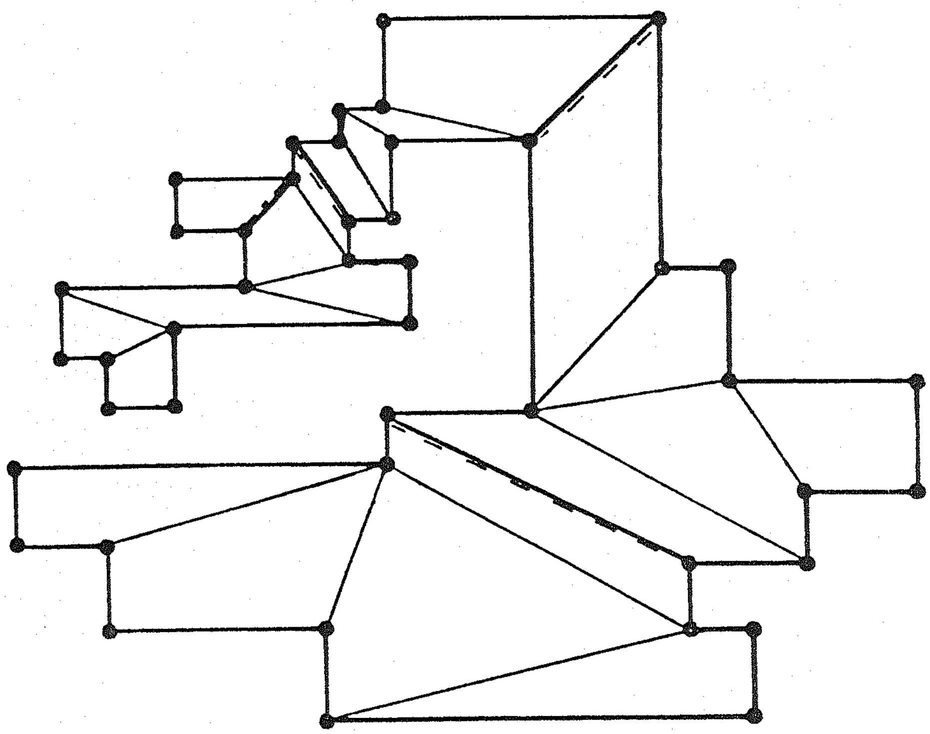


Figure 2.1 A Quadrilaterization. The dashed segments partition the polygon into monotone components.

polygon. Note that a brute force approach would require examining in worst case an exponential number of possible quadrilaterizations for a given rectilinear polygon. See figure 2.2.

The main tool used for solving minimum edge length polygonal decomposition problems is dynamic programming [Kei 83,Kli 80,LPRS 81] and we demonstate the use of it here. We let W(i,j) be the weight of the minimum weight quadrilaterization of the subpolygon P_{ij} of P cut off from p_i to p_j (i.e. $p_i,p_{i+1},...,p_j$). The idea is to build up minimum weight quadrilaterizations of larger and larger subpolygons and solve the problem using dynamic programming (see Figure 2.3). Let Q(i,j) store the vertices of the quadrilaterization that contains the diagonal p_ip_j in the minimum weight quadrilaterization of P_{ij}. Define d_{ij} to be the Euclidean distance from p_i to p_j if p_i and p_j are visible from each other and $+\infty$ otherwise.

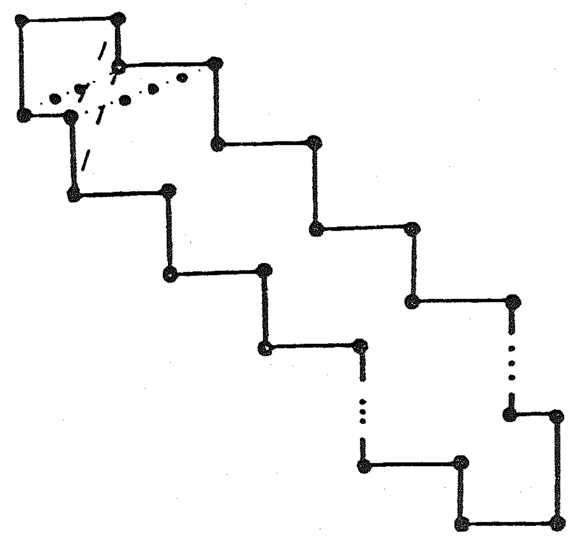


Figure 2.2 There can be an exponential number of quadrilateralizations of a rectilinear polygon.

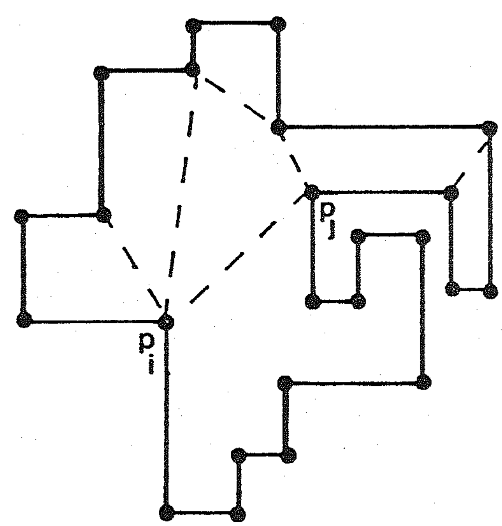


Figure 2.3 A Quadrilaterization of a subpolygon P_{ij}.

Algorithm Minimum Weight Quadrilaterization
Input: **A rectilinear polygon P**
Output: **The weight of the minimum weight quadrilaterizition and the quadrilaterization itself are stored in W(1,n) and Q respec-**

tively.

```
begin
for k := 1 to n-1 do
        for i := 1 to n-k do
        begin
                j := i + k
                begin
                        if k = 1 then W(i,j) := 0
                        else
                        if k is even then W(i,j) := +oo
                        else
                        begin
                                W(i,j) := d_ij + min(W(i,l) + W(l,m) + W(m,j))
                                (* where the minimum is taken over all l and m
                                such that i<l<m<j, and p_i, p_l, p_m, p_j form a con-
                                vex quadrilateral *)
                                Q(i,j) := (l,m)
                                (* indices of vertices exhibiting the minimum
                                are stored in Q *)
                        end
                end
        end
end.
```

To print out all the diagonals of the minimum weight quadrilaterization call the procedure PRINT(1,n).

```
Procedure PRINT(i,j)
begin
if p_i p_j is not an edge of P then output segment p_i p_j
where Q(i,j) = (l,m) do
        call PRINT(i,l); call PRINT(l,m); call PRINT (m,j)
end.
```

Complexity Analysis

The values of d_{ij} can be computed in a preprocessing step in $O(n^2)$ time. There are $O(n^2)$ W(i,j) to be calculated. Each of these requires examining $O(n^2)$ pairs of candidates for p_l and p_m. The total run time of the algorithm is therefore $O(n^4)$.

3. Decomposition of Simple Polygons

In this section we consider the decomposition of arbitrary simple polygons, first when Steiner points are allowed and then when Steiner points are disallowed.

3.1. Decompositions allowing Steiner Points

The first result in this section concerns the partitioning of a polygonal region into a minimum number of trapezoids with two horizontal sides. Triangles with a horizontal side are considered to be trapezoids with two horizontal sides one of which is degenerate. This problem is closely related to VLSI artwork data processing systems of Electron-Beam Lithography for VLSI microfabrication. In such systems the layout is stored as a set of polygonal regions per layer which should be partitioned into fundamental figures since the aperture of a pattern generator is restricted. Trapezoids have been used as fundamental figures [AA 83].

Asano and Asano [AA 83] have developed an $O(n^3)$ time algorithm for the problem of partitioning a polygonal region into the minimum number of trapezoids with two horizontal sides. They solve the problem by introducing the concept of minimally effective chords. A chord s of a polygon P is minimally effective if either (i) s joins two edges of P collinear with s, or (ii) s is a horizontal chord joining two vertices of P and containing no other vertices of P. The following theorem provides the basis for their algorithm.

Theorem

The minimum number of trapezoids needed to partition a polygon P is n - h - d - 1, where h is the number of horizontal edges of P and d is the cardinality of a maximum independent set of the intersection graph of the minimally effective chords of P.

They then show that the class of intersection graphs of minimally effective chords of polygons without holes is the class of circle graphs, and use an efficient algorithm for finding a maximum independent set of a circle graph to solve the decomposition problem.

A more difficult problem is to decompose a polygon into the minimum number of convex components. Chazelle and Dobkin [CD 79][Cha 80] exhibit an $O(n + N^3)$ time algorithm for this problem. They introduce the concept of an X-pattern from which minimal decompositions may be generated. An Xk-pattern is a particular interconnection of k notches which removes all reflex angles at the k notches and creates no new notches. The idea is to use as many X-patterns as possible. It turns out that determining whether some given notches can be combined to form an X-pattern seems too involved and they introduce a more constained type of interconnection which they call a Y-pattern. Using dynamic programming they are able to construct Y-patterns in polynomial time and with some further refinements achieve the $O(n + N^3)$ time bound.

Decomposition problems allowing Steiner points seem quite difficult. There are no futher results known concerning the decomposition of simple polygons into the minimum number of some component type. Nor are there any results known concerning the decomposition of simple polygons into components while minimizing the total internal edge length when Steiner points are allowed.

3.2. Decompositions without Steiner points

We first consider the decomposition of polygons into the minimum number of components and then the decomposition of polygons while minimizing the total internal edge length.

3.2.1. Minimum Number Decompositions

This section considers four types of fundamental components: convex, spiral, star-shaped and monotone polygons.

3.2.1.1. Convex Decompositions

Of all the polygon decomposition problems, the problem of decomposing a polygon into the minimum number of convex components has received the most attention. However, until recently, no polynomial time exact solution was known to the problem thus motivating the development of approximation algorithms. Feng and Pavlidis [FP 75] describe an O(N^3n) time algorithm which does not generally yield a minimum decomposition. Schachter's [Sch 78] O(Nn) time decomposition algorithm, based on the Delaunay triangulation, also fails to guarantee a minimum decomposition. Chazelle [Cha 82] develops an O(nlogn) time divide and conquer algorithm that finds a decomposition that contains less than 4.333.. times the optimum number of components. Recently, Greene [Gre 82a] has developed an O(nlogn) time algorithm that is guaranteed to find a decomposition that contains less than or equal to 4 times the optimum number of components. Note however that any convex decomposition that does not contain unnecessary edges will be less than of equal to 4 times the minimum decomposition. This is true since each edge can eliminate at most two notches and each notch requires at most two edges to eliminate it in any convex decomposition that does not contain unnecessary edges.

Recently, Greene [Gre 82a] discovered an O(N^2n^2) time exact algorithm for the convex decomposition problem. Independently, Keil [Kei 83] develops an algorithm for the problem that runs in O(N^2nlogn) time. This result employs a general technique for reducing the size of the state space in a dynamic programming formulation in order to achieve the polynomial time bound. The idea is to build up minimum decompositions of larger and larger subpolygons $P_{i,j}$. In general there can be an exponential number of minimum decompositions of a subpolygon. It turns out that the only edges of a minimum decomposition of a subpolygon P_{ij} that affect the decomposition's ability to form part of the global solution are the edges adjacent to the base p_ip_j. The minimum decompositions of a subpolygon can be placed in equivalence classes based on the angles the base p_ip_j forms with the two adjacent edges in the decomposition. By storing only one representative for each of these equivalence classes the complexity is reduced so that a dynamic programming approach can achieve the O(N^2nlogn) time bound.

3.2.1.2. Spiral Decompositions

Recall that a spiral polygon is a simple polygon whose boundary chain contains precisely one concave subchain. That is, a spiral polygon has precisely one set of adjacent notches. Feng and Pavlidis [FP 75] give an algorithm for decomposing a simple polygon into spiral polygons. Their algorithm does not generally yield a decomposition into the minimum number of spiral components. Keil [Kei 83] develops a dynamic programming algorithm to this problem that is similar to his algorithm for the convex decomposition problem. The algorithm may require O(n^3logn) time to find the minimum spiral decomposition.

3.2.1.3. Star-shaped Decompositions

A star-shaped polygon is a simple polygon in which the entire polygon is visible from at least one (possibly interior) fixed point of the polygon. Avis and Toussaint [AT 81] give an O(nlogn) algorithm that finds a decomposition into at most n/3 components. Their algorithm first triangulates the polygon P and then 3-colors it. A 3-coloring is an assignment of one of three colors to each vertex, such that so two vertices with the same color are adjacent. Consider the set S = $\{s_1,s_2,\ldots,s_j\}$ of the vertices with a given color. P can be decomposed into j nonoverlapping starshaped polygons $P_1,P_2,\ldots,P_j$ where P_k = {x|x is a vertex of P and x is adjacent to s_k in the triangulation of P} ∪ $\{s_k\}$, $1 \leq k \leq j$. This algorithm does not generally yield a minimum decomposition.

Keil [Kei 83] develops an O(n^5N^2logn) time algorithm for decomposing a simple polygon into the minimum number of star-shaped components. He employs the same general dynamic programming technique that he used for the convex decomposition problem. Again the idea is to build up solutions to larger and larger subpolygons P_{ij}. In general, there can be an exponential number of minimum star-shaped decompositions of a subpolygon. It turns out that there are situations where no minimum decomposition of a subpolygon can be developed into a global minimum decomposition. The solution is to introduce the concept of pseudo star-shaped polygons. A pseudo star-shaped subpolygon P_{ij} has the property that there exists a point x in P but not in P_{ij} so that every point of P_{ij} can be seen from x through the edge v_iv_j. The algorithm proceeds by dynamic programming while keeping only one star or pseudo star-shaped minimum decomposition of each of a number of equivalence classes of decompositions at each subpolygon.

3.2.1.4. Monotone Decompositions

Recall that a monotone polygon contains two extreme vertices in a prefered direction such that they are connected by two polygonal chains monotonic in the prefered direction. Lee and Preparata [LP 77] give an O(nlogn) time algorithm for decomposing a simple polygon into monotone polygons that does not generally yield a minimum decomposition. Keil [Kei 83] develops an $O(Nn^4)$ time algorithm that uses the same dynamic programming approach he uses for the convex decomposition algorithm. The algorithm relies on the fact that there are only a polynomial number of prefered directions with repect to which a subpolygon can be monotone.

3.2.2. Minimum Edge Length Decompositions

A decomposition of a simple polygon into triangles while minimizing the total internal edge length of the decomposition is known as the minimum weight triangulation of the polygon. If instead of a triangulating a polygon we triangulate an arbitrary set of points in the plane, no polynomial time algorithm is known for the minimum weight triangulation problem [GJ 79]. The minimum weight triangulation of a simple polygon can be found in $O(n^3)$ time [Kli 80]. The algorithm, due to Klincsek, uses standard dynamic programming.

Keil [Kei 83] uses dynamic programming to develop an $O(N^2n^7 \log n)$ time algorithm for the problem of decomposing a polygon into convex polygons while minimizing the total internal edge length. Figure 3.1 illustrates the difference between a decomposition into the minimum number of convex pieces and a decomposition into convex pieces that minimizes the total internal edge length. Solutions are built up for larger and larger subpolygons. It is not sufficient to keep only minimum solutions for each subpolygon but rather decompositions are kept that are minimum given that they include certain edges that affect their ability to merge into larger decompositions. Independently, Greene [Gre 82b] has noticed that his algorithm for the convex minimum number problem [Gre 82a] can be adapted to yield an $O(N^2n^2)$ time algorithm for the convex minimum edge length problem.

Keil [Kei 83] is able is use his dynamic programming approach to decomposition problems to develop an $O(n^4 \log n)$ time algorithm for finding a minimum edge length decomposition into spiral polygons, an $O(N^2n^6 \log n)$ time algorithm for finding a minimum edge length decomposition into star-shaped polygons and an $O(Nn^4)$ time algorithm for finding a minimum edge length decomposition into

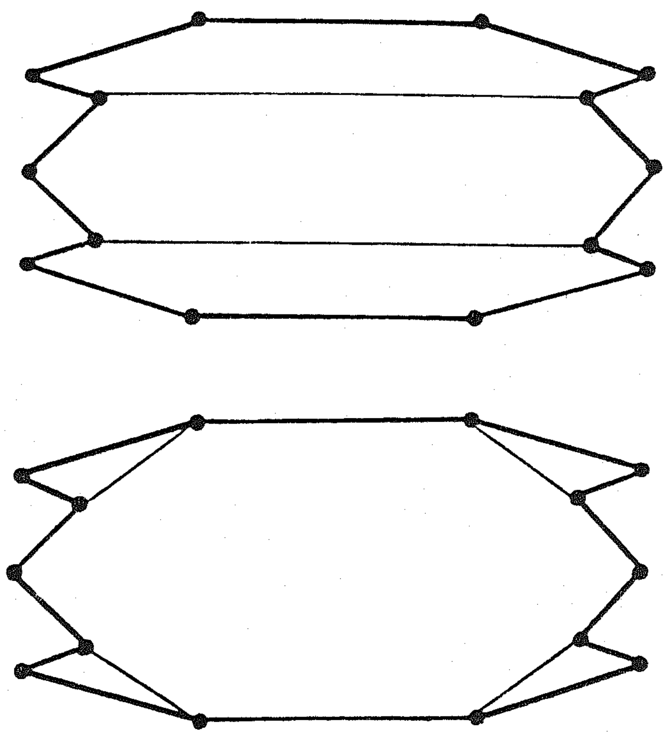


Figure 3.1 Minimum Number and Minimum "Ink" Convex Decompositions

monotone polygons.

4. Decomposition of Polygons with Holes

When we allow a polygon to contain holes the problem of decomposing the polygon in some minimal way seems to become more difficult. Recently, several of these problems have been shown to be NP-hard [GJ 79] and thus unlikely to admit efficient algorithms.

Let F be a boolean formula in conjuctive normal form with at most 3 variables per clause where $V = \{v_1,...,v_l\}$ are the variables and $C = \{c_1,...,c_m\}$ are the clauses. 3SAT is the problem of determining whether a given formula F is satisfiable. A formula is said to be planar if the bipartite graph $G(F) = (V \cup C,E)$ with $E = \{(v_i,c_j) | v_i \text{ or } \bar{v}_i \text{ is a literal in } c_j\}$ is planar. P3SAT is the problem of determining whether a given planar formula is satisfiable. Both 3SAT [GJ 79] and P3SAT [Lic 82] are known to be NP-complete. Many problems in this section are shown to be NP-hard with transformations from either 3SAT or P3SAT.

In the introduction we distinguished between decompositions that partition a polygonal object into a given type of component and decompositions that cover a polygonal object with a given type of component. In this section the difference between partitioning and covering is important.

4.1. Rectilinear Polygons with Holes

The results in this section all concern the decomposition of rectilinear polygons with holes into rectangles allowing Steiner points. The first problem is that of covering a rectilinear polygon with holes with the minimum number of rectangles. Masek [Mas 79] has shown that this problem is NP-complete with a transformation from 3SAT.

If we consider a partition, rather than a covering, then the problem of decomposing a rectilinear polygon into the minimum number of rectangles becomes polynomial. Lipski et al [LLLMP 79] have developed an $O(n^3)$ time algorithm for the problem that makes use of bipartite graph matching techniques. Ohtsuki [Oht 82] and Ohtsuki et al [OSTT 83] have improved this time and exhibit an $O(n^{5/2})$ algorithm.

If the holes are allowed to be points rather than rectilinear polygons, the nature of the problem changes again. Lingas [Lin 82a] uses a transformation from P3SAT to show that the problem of partitioning a rectilinear polygon with rectilinear polygon or point holes into the minimum number of rectangles is NP-complete.

The minimum edge length minimality criterion has also been considered. Lingas et al [LPRS 81] use transformations from P3SAT to show that the problems of decomposing a rectilinear polygon, with either rectilinear holes or rectilinear and point holes, into rectangles while minimizing total edge length are NP-complete. Lingas [Lin 82b] has developed an $O(n^4)$ heuristic for this problem that yields a partition whose length is not greater that 41 times the length of an optimal partition.

4.2. Arbitrary Polygons with Holes

A number of problems involving decomposing arbitrary polygons with holes into a given component type have been studied and found to be NP-hard. We first consider decompositions allowing Steiner points and then decompositions without Steiner points.

4.2.1. Decompositions allowing Steiner Points

The first problem is that of partitioning a polygon with polygon holes into the minimum number of triangles. Lingas [Lin 82a] uses a transformation from P3SAT to show that this problem is NP-hard. This problem arises in applications such as manipulation of LSI artwork data and image processing thus motivating the search for good approximation algorithms. Asano, Asano and Ohsuga [AAO 83] present an O(nlogn) time approximation algorithm that is guaranteed to find a decomposition that contains at most 4 times the number of triangles in an optimal decomposition.

Another problem that arises in the manipulation of VSLI artwork data processing is that of partitioning a polygon with holes into a minimum number of trapezoids with two horizontal sides. As before triangles with a horizontal side are considered to be trapezoids with two horizontal sides one of which is degenerate. Asano and Asano [AA 83] study this problem and show it to be NP-complete using a transformation from the problem of finding a maximum independent set of a straight-lines-in-the-plane graph. They also develop an O(nlogn) approximation algorithm which uses only horizontal chords to partition a polygonal region. This algorithm finds a decomposition which contains not more than 3 times the number of trapezoids in a minimum decomposition.

Both the covering and partitioning versions of the problem of decomposing a polygon with holes into a minimum number of convex components are NP-hard. O'Rourke and Supowit [OS 83] use a transformation from 3SAT to show the covering problem is NP-hard while Lingas [Lin 82a] uses a transformation from P3SAT to show the partitioning problem is NP-hard. While neither problem is known to be in NP, O'Rourke [ORo 82] has proven that the covering problem is decidable.

The problems of covering a polygon with holes with the minimum number of spiral or star-shaped components are also NP-hard. O'Rourke and Supowit [OS 83] use a transformation from 3SAT to prove this.

Use of the minimum edge length minimality crition does not make the convex partitioning problem easier. Lingas et al [LPRS 81] use a transformation from P3SAT to show the problem is NP-hard.

4.2.2. Decompositions without Steiner Points

When Steiner points are disallowed both the covering and partitioning versions of the problems of decomposing a polygon with holes into the minimum number of convex, spiral and star-shaped components are NP-complete. O'Rourke and Supowit [OS 83] show that the covering problems are NP-complete. The convex partitioning problem is shown to be NP-complete by Lingas [Lin 82a]. Keil [Kei 83] modifies the proofs of O'Rourke and Supowit to show that the spiral and star-shaped partitioning problems are NP-complete. Using a similar proof he also shows that the monotone partitioning problem is NP-complete.

The only minimum edge length result concerns partitioning into convex polygons. Keil [Kei 83] uses a complex transformation from P3SAT to show this problem to be NP-complete.

5. Other Decompositions

So far all of the decomposition methods have used only set union to relate the decomposition to the original polygonal object. If we also allow set difference we open an entire class of problems which have been called sum/difference decompositions [Tou 80]. Are there polynomial time algorithms for representing a polygonal object by the sum and difference of the minimum number of some givem type of component? Batchelor [Bat 80] describes a decomposition into a tree of convex polygons. The root is the convex hull of the polygon, the second level nodes represent the convex hulls of concavities in the polygon, the third the convex hulls of concavities of concavities, and so on.

There are a number of results concerning procedure-directed decomposition techniques. In most of these there is no obvious minimality criterion being used so we consider these techniques to be outside the scope of this paper.

6. REFERENCES

[ACYB 80] N. Ahuya, R.T. Chien, R. Yen and N. Birdwell, *Interference detection and collision avoidance among three dimensional objects*, Proc. First Annual National Conference on Artificial Intelligence, Stanford, California, 1980, 44-48.

[AA 83] T. Asano and T. Asano, *Minimum Partition of Polygonal Regions into Trapezoids*, Proc. 24th Annual Symposium on Foundations of Computer Science, 1983, 233-241.

[AAO 83] T. Asano, T. Asano and Y. Ohsuga, *Partitioning polygonal regions into triangles*, Papers of Technical Groups of IECE, CAS83-98, 31-36.

[AT 81] D. Avis and G.T. Toussaint, *An efficient algorithm for decomposing a polygon into star-shaped polygons*, Pattern Recognition, 13, 6, (1981), 395-398.

[Bat 80] B.G. Batchelor, *Hierarchical shape description based upon convex hulls of concavities*, Journal of Cybernetics,10 (1980),205-210.

[CD 79] B. Chazelle and D. Dobkin, *Decomposing a polygon into its convex parts*, Proceedings of the 11th Annual ACM Symposium on Theory of Computing, 1979, 38-48.

[Cha 80] B. Chazelle, *Computational Geometry and Convexity*, Ph.D. Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania, July 1980.

[Cha 82] B. Chazelle, *A theorem on polygon cutting with applications*, Proc. 23rd Annual Symposium on Foundations of Computer Science,Chicago, 1982,339-349.

[CI 83] B. Chazelle and J. Incerpi, *Triangulating a polygon by Divide-and-conquer*, Proc. 21st Annual Allerton Conference on Communication, Control and Computing, 1983.

[EAT 81] H. El Gindy, D. Avis and G.T. Toussaint, *Appications of a two dimensional hidden-line algorithm to other geometric problems*, Technical Report No. SOCS-81.13, School of Computer Science, McGill University, April 1981.

[FP 75] H. Feng and T. Pavlidis, *Decomposition of Polygons Into Simpler Components: Feature Generation for Syntactic Pattern Recognition*, IEEE Trans. on Computers C-24 (1975), 636-650.

[FSS 81] L. Ferrari, P.V. Sankar and J. Sklansky, *Minimal rectangular partitions of digitized blobs*, Proc. 5th International Conference on Pattern Recognition, Miami Beach, 1981, 1040-1043.

[GJPT 78] M.R. Garey, D.S. Johnson, F.P. Preparata and R.E. Tarjan, *Triangulating a simple polygon*, Information Processing Letters,7 (1978), 175-179.

[GJ 79] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco, CA, Freeman, 1979.

[Gra 72] R. Graham, *An efficient algorithm for determining the convex hull of a planar set*, Information Processing Letters, 1 (1972), 132-133.

[GY 84] R.L. Graham and F.F. Yao, *Finding the convex hull of a simple polygon*, Journal of Algorithms, in press.

[Gre 82a] D. Greene, *The Decomposition of Polygons into Convex Parts*, Manuscript, Xerox PARC, 1982.

[Gre 82b] D. Greene, Private Communication, May 1982.

[KKK 80] J. Kahn, M. Klawe, and D. Kleitman, *Traditional galleries require fewer watchmen*, SIAM J. Alg. Disc. Meth., Vol. 4, 2, June 1983, 194-206.

[Kei 83] J.M. Keil, *Decomposing Polygons into Simpler Components*, Ph.D. thesis, Department of Computer Science, University of Toronto, 1983.

[Kli 80] G.T. Klincsek, *Minimal Triangulations of Polygonal Domains*, Annals of Discrete Math., 9, 1980, pp. 121-123.

[Lee 80] D.T. Lee, *On finding the convex hull of a simple polygon*, Tech. Rep. No. 80-03-FC-01, Dept. Elec. Engr. and Comp. Sci., Northwestern University, 1980.

[LP 77] D.T. Lee and F.P. Preparata, *Location of a point in a planar subdivision and its applications*, SIAM J. on Comput., 6 (1977), 594-606.

[Lic 82] D. Lichtenstein, *Planar Formulae and their Uses*, SIAM J. on Comput., 11 (1982), 329-343.

[Lin 82a] A. Lingas, *The power of non-rectilinear holes*, Proc. 9th Colloquium on Automata, Languages and Programming, Aarhus, 1982.

[Lin 82b] A. Lingas, *Heuristics for Minimum Edge Length Rectangular Partition*, manuscript, MIT,1982.

[LPRS 81] A. Lingas, R. Pinter, R. Rivest and A. Shamir, *Minimum Edge Length Partitioning of Rectilinear Polygons*, Proc. 20th Annual Allerton Conference on Communication, Control and Computing, 1982, 53-63.

[Lip 79] W. Lipski, *Finding a Manhattan path and related problems*, Tech. Rep. ACT-17 Coordinated Science Lab., University of Illinois at Urbana-Champaign, August 1979.

[LLLMP 79] W. Lipski, E. Lodi, F. Luccio, C. Mugnai and L. Pagli, *On two dimensional data organization II*, Fundamenta Informaticae, 2 (1979), 245-260.

[Mas 79] W.J. Masek, *Some NP-complete set covering problems*, unpublished manuscript, August 1979.

[MA 79] D. McCallum and D. Avis, *A linear algorithm for finding the convex hull of a simple polygon*, Infor. Proc. Lett., 9 (1979),201-206.

[MF 82] D.Y. Montuno and A. Fournier, *Detecting Intersections Among Star Polygons*, Tech. Pept. CSRG-146, University of Toronto, Toronto, Canada, Sept. 1982.

[NS 79] W. Newman and R. Sproull, *Principles of Interactive Computer Graphics*, Second Edition, McGraw Hill, New York, 1979.

[Oht 82] T. Ohtsuki, *Minimum dissection of rectilinear regions*, Proc. of 1982 IEEE International Symposium on Circuits and Systems, Rome, 1982, 1210-1213.

[OSTT 83] T. Ohtsuki, M. Sato,M. Tachibana and S. Torii, *Minimum partitioning of rectilinear regions*, Trans. of Information Processing Society of Japan.

[ORo 82] J. O'Rourke, *The complexity of computing minimum convex covers for polygons*, Proc. 20th Allerton Conf., Allerton IL, Oct. 1982.

[OS 83] J. O'Rourke and K.J. Supowit, *Some NP-Hard Polygon Decomposition Problems*, IEEE Trans. on Information Theory, Vol. IT-29, 2, (1983), 181-190.

[Pav 68] T. Pavlidis, *Analysis of set patterns*, Pattern Recognition, 1 (1968), 165-178.

[Pav 77] T. Pavlidis, *Structural pattern recognition*, Springer Series in Electrophysics Vol. 1, Springer Verlag Berlin, Heidelberg, New York, 1977.

[Pre 77] F.P. Preparata, *Medial Axis of a Convex Polygon*, in Steps into Computational Geometry, F.P.Preparata, ed., Coordinated Science Lab., Tech. Rept. R-760, 1977, 6-9.

[Sac 82] J.R. Sack, *An O(nlog(n)) algorithm for decomposing simple rectilinear polygons into convex quadrilaterals*, Proc. 20th Annual Allerton Conf. on Communication, Control and Computing, 1982, 64-75.

[Sac 83] J.-R. Sack, *A simple hidden-line algorithm for simple polygons*, School of Computer Science, Carleton University, unpublished manuscript, 1983.

[ST 81] J.-R. Sack and G.T. Toussaint, *A linear time algorithm for decomposing rectilinear star-shaped polygons into convex quadrilaterals*, Proc. 19th Annual Allerton Conf. on Communication, Control and Computing, Urbana, 1981, 21-30.

[Sch 78] B. Schachter, *Decomposition of polygons into convex sets*, IEEE Transactions on Computers, C-27, 1978, 1078-1082.

[SvL 80] A.A. Schoone and J. van Leeuwan, *Triangulating a star-shaped polygon*, Tech. Rept. RUV-CS-80-3, University of Utrecht, April 1980.

[Sha 78] M.I. Shamos, *Computational Geometry*, Ph.D. Thesis, Yale University, 1978.

[Tou 80] G.T. Toussaint, *Pattern recognition and geometrical complexity*, Proc. Fifth International Conf. on Pattern Recognition, Miami Beach, 1980, 1324-1347.

[Yao 79] A. Yao, *A lower bound to finding convex hulls*, internal report, Computer Science Department, Stanford University, 1979.

CARLETON UNIVERSITY

School of Computer Science

BIBLIOGRAPHY OF SCS TECHNICAL REPORTS

SCS-TR-1 **THE DESIGN OF CP-6 PASCAL**
Jim des Rivieres and Wilf R. LaLonde, June 1982.

SCS-TR-2 **SINGLE PRODUCTION ELIMINATION IN LR(1) PARSERS: A SYNTHESIS**
Wilf R. LaLonde, June 1982.

SCS-TR-3 **A FLEXIBLE COMPILER STRUCTURE THAT ALLOWS DYNAMIC PHASE ORDERING**
Wilf R. LaLonde and Jim des Rivieres, June 1982.

SCS-TR-4 **A PRACTICAL LONGEST COMMON SUBSEQUENCE ALGORITHM FOR TEXT COLLATION**
Jim des Rivieres, June 1982.

SCS-TR-5 **A SCHOOL BUS ROUTING AND SCHEDULING PROBLEM**
Wolfgang Lindenberg, Frantisek Fiala, July 1982.

SCS-TR-6 **ROUTING WITHOUT ROUTING TABLES**
Nicola Santoro, Ramez Khatib, July 1982.

SCS-TR-7 **CONCURRENCY CONTROL IN LARGE COMPUTER NETWORKS**
Nicola Santoro, Hasan Ural, July 1982.

SCS-TR-8 **ORDER STATISTICS ON DISTRIBUTED SETS**
Nicola Santoro, Jeffrey B. Sidney, July 1982.

SCS-TR-9 **OLIGARCHICAL CONTROL OF DISTRIBUTED PROCESSING SYSTEMS**
Moshe Krieger, Nicola Santoro, August 1982.

SCS-TR-10 **COMMUNICATION BOUNDS FOR SELECTION IN DISTRIBUTED SETS**
Nicola Santoro, Jeffrey B. Sidney, September 1982.

SCS-TR-11 **SIMPLE TECHNIQUE FOR CONVERTING FROM A PASCAL SHOP TO A C SHOP**
Wilf R. LaLonde, John R. Pugh, November 1982.

SCS-TR-12 **EFFICIENT ABSTRACT IMPLEMENTATIONS FOR RELATIONAL DATA STRUCTURES**
Nicola Santoro, December 1982.

SCS-TR-13 **ON THE MESSAGE COMPLEXITY OF DISTRIBUTED PROBLEMS**
Nicola Santoro, December 1982.

SCS-TR-14 **A COMMON BASIS FOR SIMILARITY MEASURES INVOLVING TWO STRINGS**
R. L. Kashyap and B. J. Oommen, January 1983.

SCS-TR-15 **SIMILARITY MEASURES FOR SETS OF STRINGS**
R. L. Kashyap and B. J. Oommen, January 1983.

SCS-TR-16 **THE NOISY SUBSTRING MATCHING PROBLEM**
R. L. Kashyap and B. J. Oommen, January 1983.

SCS-TR-17 **DISTRIBUTED ELECTION IN A CIRCLE WITHOUT A GLOBAL SENSE OF ORIENTATION**
E. Korach, D. Rotem, N. Santoro, January 1983.

SCS-TR-18 **A GEOMETRICAL APPROACH TO POLYGONAL DISSIMILARITY AND THE CLASSIFICATION OF CLOSED BOUNDARIES**
R. L. Kashyap and B. J. Oommen, January 1983.

SCS-TR-19 **SCALE PRESERVING SMOOTHING OF POLYGONS**
R. L. Kashyap and B. J. Oommen, January 1983.

SCS-TR-20 **NOT-QUITE-LINEAR RANDOM ACCESS MEMORIES**
Jim des Rivieres, Wilf LaLonde and Mike Dixon, August 1982, Revised March 1, 1983.

SCS-TR-21 **SHOUT ECHO SELECTION IN DISTRIBUTED FILES**
D. Rotem, N. Santoro, J. B. Sidney, March 1983.

SCS-TR-22 **DISTRIBUTED RANKING**
E. Korach, D. Rotem, N. Santoro, March 1983.

SCS-TR-23 **A REDUCTION TECHNIQUE FOR SELECTION IN DISTRIBUTED FILES : I**
N. Santoro, J. B. Sidney, April 1983.

SCS-TR-24 **LEARNING AUTOMATA POSSESSING ERGODICITY OF THE MEAN : THE TWO ACTION CASE**
M. A. L. Thathachar and B. J. Oommen, May 1983.

SCS-TR-25 **ACTORS - THE STAGE IS SET**
John R. Pugh, June 1983.

SCS-TR-26 **ON THE ESSENTIAL EQUIVALENCE OF TWO FAMILIES OF LEARNING AUTOMATA**
M. A. L. Thathachar and B. J. Oommen, May 1983.

SCS-TR-27 **GENERALIZED KRYLOV AUTOMATA AND THEIR APPLICABILITY TO LEARNING IN NONSTATIONARY ENVIROMENTS**
B. J. Oommen, June 1983.

SCS-TR-28 **ACTOR SYSTEMS: SELECTED FEATURES**
Wilf R. LaLonde, July 1983.

SCS-TR-29 **ANOTHER ADDENDUM TO KRONECKER'S THEORY OF PENCILS**
M. D. Atkinson, August 1983.

SCS-TR-30 **SOME TECHNIQUES FOR GROUP CHARACTER REDUCTION**
M. D. Atkinson and R. A. Hassan, August 1983.

SCS-TR-31 **AN OPTIMAL ALGORITHM FOR GEOMETRICAL CONGRUENCE**
M. D. Atkinson, August 1983.

SCS-TR-32 **MULTI-ACTION LEARNING AUTOMATA POSSESSING ERGODICITY OF THE MEAN**
B. J. Oommen and M. A. L. Thathachar, August 1983.

SCS-TR-33 **FIBONACCI GRAPHS, CYCLIC PERMUTATIONS AND EXTREMAL POINTS**
N. Santoro and J, Urrutia, December 1983

SCS-TR-34 **DISTRIBUTED SORTING**
D. Rotem, N. Santoro, and J. B. Sidney, December 1983

SCS-TR-35 **A REDUCTION TECHNIQUE FOR SELECTION IN DISTRIBUTED FILES: II**
N. Santoro, M. Scheutzow, and J. B. Sidney, December 1983

SCS-TR-36 **THE ASYMPTOTIC OPTIMALITY OF DISCRETIZED LINEAR REWARD-INACTION LEARNING AUTOMATA**
B. J. Oommen and Eldon Hansen, January 1984

SCS-TR-37 **GEOMETRIC CONTAINMENT IS NOT REDUCIBLE TO PARETO DOMINANCE**
N. Santoro, J. B. Sidney, and J. Urrutia, January 1984

SCS-TR-38 **AN IMPROVED AGLORITHM FOR BOOLEAN MATRIX MULTIPLICATION**
N. Santoro and J. Urrutia, January 1984

SCS-TR-39 **CONTAINMENT OF ELEMENTARY GEOMETRIC OBJECTS**
J. Sack, N. Santoro and J. Urrutia, February 1984

SCS-TR-40 **SADE: A PROGRAMMING ENVIRONMENT FOR DESIGNING AND TESTING SYSTOLIC ALGORITHMS**
J. P. Corriveau and N. Santoro, February 1984

SCS-TR-41 **INTERSECTION GRAPHS, {B }-ORIENTABLE GRAPHS AND PROPER CIRCULAR ARC GRAPHS**
Jorge Urrutia, February 1984

SCS-TR-42 **MINIMUM DECOMPOSITIONS OF POLYGONAL OBJECTS**
J. Mark Keil and Jorg-R. Sack, March 1984

SCS-TR-43 **AN ALGORITHM FOR MERGING HEAPS**
Jorg-R. Sack and Thomas Strothotte, March 1984