Computational Complexity of Art Gallery Problems

D. T. LEE, SENIOR MEMBER, IEEE, AND ARTHUR K. LIN, MEMBER, IEEE

***Abstract*—We study the computational complexity of the art gallery problem originally posed by Klee, and its variations. Specifically, the problem of determining the minimum number of vertex guards that can see an n-wall simply connected art gallery is shown to be NP-hard. The proof can be modified to show that the problems of determining the minimum number of edge guards and the minimum number of point guards in a simply connected polygonal region are also NP-hard. As a byproduct, the problem of decomposing a simple polygon into a minimum number of star-shaped polygons such that their union is the original polygon is also shown to be NP-hard.**

I. Introduction

THE ART GALLERY problem of determining how many guards are sufficient to see every point in the interior of an n-wall art gallery room was posed by Klee [11]. Conceptually, the room is a simple polygon P with n vertices, and the guards are stationary points in P that can see any point of P connected to them by a straight line segment that lies entirely within P. It has been proven by Chvatal [6] that $\lfloor n/3 \rfloor$ guards are always sufficient, and this number is the best possible in some cases. Fisk [9] later found a simpler proof which lends itself to an $O(n \log n)$ algorithm developed by Avis and Toussaint [3] for locating these $\lfloor n/3 \rfloor$ stationary guards. If the polygon is rectilinear, that is, the edges of the polygon are either horizontal or vertical, Kahn *et al.* [12] have shown that $\lfloor n/4 \rfloor$ guards are sufficient and sometimes necessary. O'Rourke [14] later gave a completely different and somewhat simpler proof of this result. Sack [17] and Edelsbrunner, *et al.* [7] have, based on the results of [12] and [14], respectively, devised an $O(n \log n)$ algorithm for locating these $\lfloor n/4 \rfloor$ guards.

Presumably, the guards can be arbitrarily placed in the interior of the art gallery. However, the aforementioned results indicate that the guards can be restricted to the vertices of the polygon without affecting the outcome. Hence we shall refer to this problem as the *vertex guard problem* and the former as the *point guard problem*. The vertex guard problem can be treated as a polygon decomposition problem in which a polygon is to be decomposed according to vertex visibility. For polygon decompositions into various types of "primitives" and their complexities see, for example, [2], [5], [8], and [13]. In this paper we address the computational complexity of the *minimum vertex guard problem*, that is, determining the minimum number of vertex guards for an n-edge simple polygon, and we show that this problem and the *minimum point guard problem* are NP-hard.

In [4] Avis and Toussaint studied and provided an $O(n)$ algorithm for the following edge visibility problem, that is, given an n-edge simple polygon P and a guard, called an *edge guard*, patrolling along an edge of P, decide if every point of P can be seen by the guard. In the context of the art gallery problem, O'Rourke [15] showed that $\lfloor n/4 \rfloor$ such mobile guards, where the guards can move along fixed line segments (the edges of the polygon, for example) are always sufficient and sometimes necessary. We shall show that the *minimum edge guard problem*, that is, determining the minimum number of edge guards needed to see any point of P for an n-edge simple polygon, is also NP-hard. For definitions of NP-hardness and NP-completeness, refer to [10].

The terminology and notation will be given in Section II, and in Sections III and IV the main results of NP-hardness of the aforementioned problems are presented.

Manuscript received May 24, 1984; revised June 17, 1985. This work was supported in part by the National Science Foundation under Grant MCS8342682.

D. T. Lee is with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60201.

A. K. Lin was with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL. He is now with Gerber System Technology, 40 Gerber Road East, South Winsor, CT 06074.

IEEE Log Number 8406631.

II. Definitions and Notation

A simple polygon $P = (v_0, v_1, \cdots, v_n)$ is a closed plane figure whose boundary bd(P) is composed of straight line edges (v_i, v_{i+1}), $i = 0, 1, \cdots, n-1$, $v_0 = v_n$, and where no two nonconsecutive edges intersect. The edges of P are oriented in the counterclockwise sense, that is, when they are traversed, the interior of P always lies to the left. If P contains holes that are themselves represented as simple polygons (with no holes), then P is said to be *multiply connected*; if P contains no holes, then it is said to be *simply connected*. Unless otherwise specified, the term polygon refers to a simply connected simple polygon, and P refers to the region enclosed in bd(P).

Two points p and q of polygon P are said to be visible from each other if the line segment $\overline{pq}$ connecting p and q lies completely in P. A point p of P is said to be visible from an edge (u, v) of P if there exists a point q on (u, v) such that p and q are visible. For any point v of P the

0018-9448/86/0300-0276\$01.00 ©1986 IEEE

v-cover is the locus of points of P that are visible from v, and the (u, v)*-cover*, where (u, v) is an edge of P, is defined similarly. The v-cover is also known as the *visibility polygon* for v in P and the (u, v)-cover is known as the *edge-visibility polygon* for (u, v) in P (see, for example, [4]). A polygon P is *star-shaped* if a point v in P exists such that the v-cover is the polygon P. Thus any v-cover for any point v in P is a star-shaped polygon. The kernel of a polygon P is the region in P such that for any point q in P, q is visible from any point in this region.

III. Vertex Guard and Edge Guard Problems

Let V denote a certain subset of vertices of P. We say that a V*-cover* of P exists or P is *coverable* by V if a subset $T \subseteq V$ of vertices exists such that $\bigcup_{v \in T} v\text{-cover} = P$. T is said to be a *minimum cover* if $|T|$ is minimum among all V-covers of P.

We are concerned with the following problems. Given P and V, we must determine if P is coverable by V and, if so, find a minimum cover. The former problem can be solved in polynomial time, since for any v in V the v-cover and the union of any two covers (polygonal regions) can be computed in polynomial time. The latter, however, is NP-hard as we shall show.

An application of this problem is the following. We have a region P to be monitored by a number of guards that are placed at positions specified in V, and we want to find if the entire region of P is indeed covered by the guards, and, if so, we want to minimize the number of guards needed.

Vertex Guard Problem of a Polygonal Region

Instance: A polygonal region $P = (v_0, v_1, \cdots, v_{n-1}, v_n)$ and a positive integer $K < n$ exist.

Question: Does a subset $T \subseteq V$ exist with $|T| \leq K$, such that

$$\bigcup_{v_i \in T} v_i\text{-cover} = P$$

where $V = \{v_0, v_1, \cdots, v_{n-1}\}$?

Proposition 1: The vertex guard problem for simply connected polygonal regions (VGSCP) is NP-complete.

Proof: It is easy to see that VGSCP is in NP since a nondeterministic algorithm needs only to guess a subset $V' \subseteq V$ of K vertices and check in polynomial time if

$$\bigcup_{a \in V'} a\text{-cover} = P.$$

To show NP-completeness we reduce the following NP-complete problem to the VGSCP problem.

Boolean Three Satisfiability (3SAT)

Instance: A set $U = \{u_1, u_2, \cdots, u_n\}$ of Boolean variables and a collection $C = \{c_1, c_2, \cdots, c_m\}$ of clauses over U exist such that $c_i \in C$ is a disjunction of precisely three literals.

Question: Does a satisfying truth assignment for C, that is, a truth assignment to the n variables in U, exist such that the conjunctive normal form $c_1 c_2 \cdots c_m$ is true?

We will show that 3SAT is polynomially transformable to VGSCP. The goal is to accept an instance of 3SAT as input and to construct in polynomial time a simply connected polygonal region P such that P is coverable by K or fewer vertices of P if and only if the instance of 3SAT is satisfiable. We first introduce some basic constructs on which the simply connected polygonal region is built and identify a number of distinguished points in this polygonal region such that no two different distinguished points can be covered by a single vertex. In the following construction the dotted lines shown in the figures indicate where these basic constructs are to be "attached" to the main polygonal region. Let the bound K used in the VGSCP be $3m + n + 1$.

Literal Patterns: The literal pattern is shown in Fig. 1. The dot shown is a distinguished point associated with the pattern. The important characteristic of the pattern is that only vertex a or vertex b can cover the entire region defined by the pattern. Three such patterns per clause will exist in the final construction, each of which corresponds to one literal.

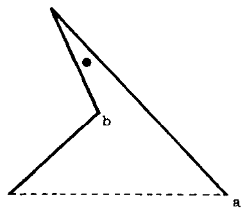

Fig. 1. Literal pattern.

Clause Junctions: Without loss of generality, consider the clause $C = A + B + D$, where $A \in \{u_i, \overline{u_i}\}$, $B \in \{u_j, \overline{u_j}\}$, and $D \in \{u_k, \overline{u_k}\}$ are literals, and u_i, u_j, and u_k are variables in U. The basic pattern for the clause junction C_h is shown in Fig. 2.

Let Δabc denote the triangle area determined by points a, b, and c, and let $(a_1, \cdots, a_h)$ denote that points a_1, $a_2, \cdots$, and a_h are collinear. Thus in the pattern of Fig. 2 we have $(g_{h2}, g_{h8}, a_{h4}, a_{h1}, b_{h4}, b_{h1}, d_{h4}, d_{h1}, g_{h9}, g_{h1})$,

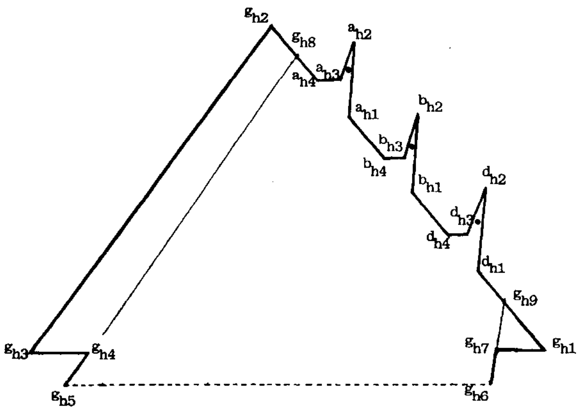

Fig. 2. Clause junction C_h.

$(g_{h3}, g_{h4}, g_{h7}, g_{h1})$, (g_{h8}, g_{h4}, g_{h5}), and (g_{h9}, g_{h7}, g_{h6}). Moreover, $|(g_{h2}, g_{h8})| = |(g_{h8}, a_{h4})|$, and $|(d_{h1}, g_{h9})| = |(g_{h9}, g_{h1})|$, where $|(u, v)|$ denotes the length of the edge (u,v).

Since none of g_{hi}, $i = 1,2,\cdots,7$ can cover $\Delta a_{h1}a_{h2}a_{h3}$, $\Delta b_{h1}b_{h2}b_{h3}$, and $\Delta d_{h1}d_{h2}d_{h3}$ in the pattern for C_h, they cannot be used to cover these three triangles as shown in Fig. 2. Furthermore, any two vertices of each literal pattern in the clause are not sufficient by themselves to cover the region defined by the pattern C_h. We therefore have the following.

Lemma 1: At least three vertices of C_h are required to cover the region defined by the pattern C_h shown in Fig. 2.

Lemma 2: Only seven three-vertex covers exist that can cover the region defined by the pattern C_h.

Proof: Since none of a_{h4}, b_{h4}, d_{h4}, and g_{hi}, $i = 1,2,\cdots,7$, can cover any of the three literal patterns, they cannot be chosen as possible candidates in the cover for Fig. 2. Therefore, only $H_i = \{a_{hi}, b_{hi}, d_{hi}\}$, $i = 1,2,3$, need be considered. If a_{h2} is chosen, $\Delta a_{h1}a_{h3}a_{h4}$ cannot be covered no matter which vertices are selected for the literal patterns B and D. This implies that vertices in H_2 cannot be candidates either. Vertices a_{h1} and a_{h3} should not be selected at the same time. If they were selected, we would need two more vertices to cover the literal patterns for B and D. This is also true for b_{h1} and b_{h3}, and d_{h1} and d_{h3}. This means that eight possible three-vertex covers exist from the sets H_1 and H_3 of vertices. Since the three vertices in H_3 cannot cover $\Delta g_{h1}g_{h2}g_{h3}$, we are left with seven three-vertex covers. It is easy to check that each of these seven three-vertex covers, that is, $\{a_{h1}, b_{h1}, d_{h1}\}$, $\{a_{h1}, b_{h1}, d_{h3}\}$, $\{a_{h1}, b_{h3}, d_{h1}\}$, $\{a_{h1}, b_{h3}, d_{h3}\}$, $\{a_{h3}, b_{h1}, d_{h1}\}$, $\{a_{h3}, b_{h1}, d_{h3}\}$, and $\{a_{h3}, b_{h3}, d_{h1}\}$, can indeed cover the region defined by the pattern C_h.

Labeling Vertices: We shall label vertices a_{h1} and a_{h3} of literal pattern A as follows. If $A = u_i$, a_{h1} and a_{h3} are labeled as T and F (true and false), respectively; if $A = \overline{u_i}$, a_{h1} and a_{h3} are labeled as F and T, respectively. In other words, vertex a_{h1} represents truth value *true* for literal A, and a_{h3} represents *false*. Vertices b_{h1} and b_{h3} of literal pattern B, and d_{h1} and d_{h3} of literal pattern D are labeled in the same manner.

We say that the truth value assignment to the variables u_i, u_j, and u_k in clause C_h is a true assignment if the resulting truth value of C_h is true; otherwise, it is a false assignment. The key idea in the labeling of vertices in a clause junction C_h is that the vertices that are selected according to the truth value assignment of $\{u_i, u_j, u_k\}$ can cover the region defined by $g_{h1}g_{h2}\cdots g_{h7}$ "free," that is, without increasing the number of vertices needed to cover the region defined by the pattern C_h, if and only if the assignment is a true assignment. Specifically, we have the following.

Lemma 3: The three vertices selected from the clause pattern $C_h = A + B + D$ cover the region defined by C_h if and only if the truth values represented by the labels of these vertices give a true assignment for C_h.

Proof: C_h has a truth value false if and only if the truth values of literals A, B, and D are false. According to the vertex labels, this implies that the vertices selected must be a_{h3}, b_{h3}, and d_{h3}. From Lemma 2 exactly seven three-vertex covers exist that can cover the region defined by C_h. We claim that the labels of each vertex in the three-vertex cover correspond to a true assignment to the variables u_i, u_j, and u_k for C_h. Since the verification for each of these seven three-vertex covers of the claim is straightforward, we leave out the details.

Variable Patterns: The variable pattern for variable u_i is as shown in Fig. 3. Note that a distinguished point exists in $\Delta t_{i1}t_{i2}t_{i3}$ and that $\Delta t_{i1}t_{i2}t_{i3}$ can only be covered by t_{i1}, t_{i2}, t_{i3}, t_{i5}, t_{i6}, or t_{i8}. One such pattern will exist per variable in the final construction. We shall henceforth refer to the two legs of the variable pattern as *rectangles* or *rectangular regions*, although they are not really rectangles. Now we consider how to put variable patterns and clause junctions together.

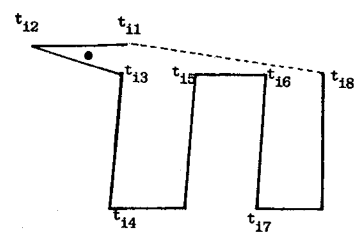

Fig. 3. Variable pattern; $t_{i1}t_{i2}$ and $(t_{i3}, t_{i5}, t_{i6}, t_{i8})$ are parallel.

Complete Construction of a Simply Connected Polygonal Region: Three steps are needed.

a) Putting variable patterns and clause junctions together: We put literal patterns and clause junctions together as shown in Fig. 4. In Fig. 4 we have that 1) vertex W can cover the n literal patterns, except $\Delta t_{i1}t_{i2}t_{i3}$, $i = 1,2,\cdots,n$, and 2) $(W, g_{15}, g_{16}, g_{25}, g_{26}, \cdots, g_{m5}, g_{m6})$, $(t_{11}, g_{h5}, g_{h4}, g_{h8})$, and $(t_{n8}, g_{h6}, g_{h7}, g_{h9})$, $h = 1,2,\cdots,m$.

b) Augmenting variable patterns with "spikes": Suppose variable u_i appears in clause C_h. If u_i itself is in C_h, then the two spikes, $\overline{pq}$ and $\overline{p_1q_1}$ as shown in Fig. 5, where (p, q, t_{i5}, a_{h1}) and $(p_1, q_1, t_{i8}, a_{h3})$, are added to Fig. 4. If its complement $(\overline{u_i})$ is in C_h, then the two spikes, $\overline{pq}$ and $\overline{p_1q_1}$ as shown in Fig. 6, where (p, q, t_{i5}, a_{h3}) and $(p_1, q_1, t_{i8}, a_{h1})$, are added to Fig. 4. This is equivalent to labeling vertex t_{i5} F and vertex t_{i8} T. That is, vertices t_{i5} and t_{i8} represent, respectively, truth values false and true for variable u_i.

From the previous construction the picture is not a simply connected polygonal region because the new added spikes are line segments. For each spike $\overline{pq}$, p and q are collinear with b and t, where b is a vertex (a_{h1}, for example) of a literal pattern and t is a vertex (t_{i8}, for example) of a variable pattern. Therefore, $\overline{pq}$ can be covered by p, q, b, and t. The next step is to construct a polygonal region out of $\overline{pq}$ such that this polygonal region can only be covered by b, t, and its vertices.

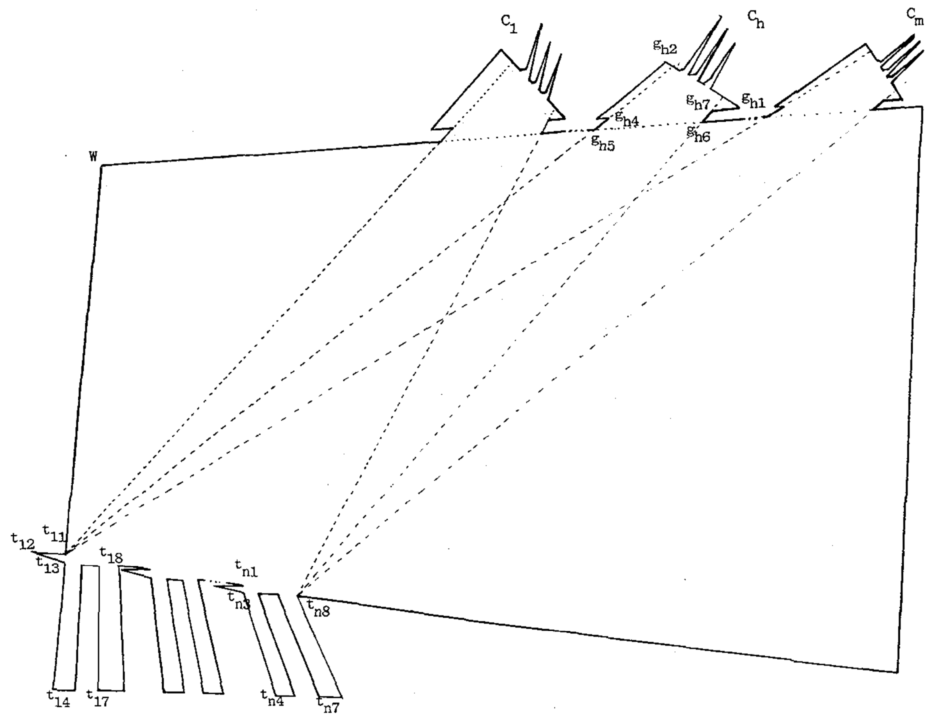

Fig. 4. Putting variable patterns and clause junctions together.

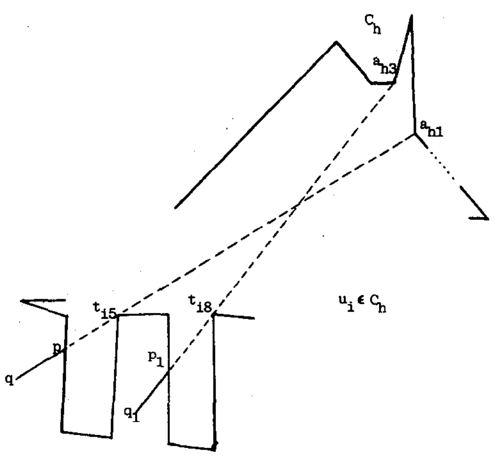

Fig. 5. Augmenting spikes when u_i in C_h.

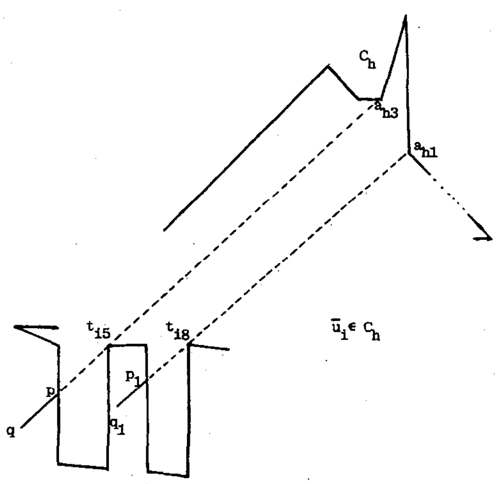

Fig. 6. Augumenting spikes when $\bar{u}_i$ in C_h.

c) Replacing spikes by polygonal regions: Suppose a is in literal pattern A, and A is in clause junction C_h. If a corresponds to a_{h1}, the shaded area in Fig. 7 is the polygonal region to replace $\overline{pq}$. More precisely, we have in Fig. 7 (a_{h1}, t_{i5}, p, q) and (a_{h1}, e, f). That is, we have a triangle $\Delta a_{h1} fq$. We replace $\overline{pq}$ similarly if a corresponds to a_{h3}. We call the polygonal regions created as described the *consistency-check patterns*.

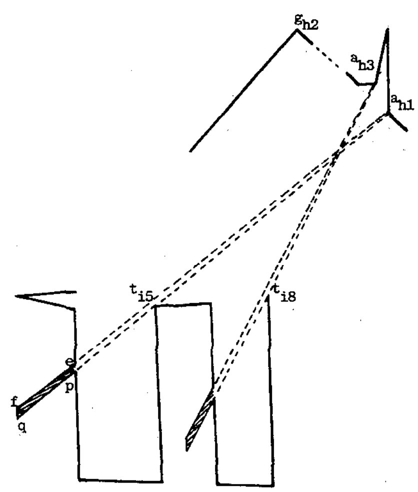

Fig. 7. Each spike is replaced with small region.

Fig. 8 is an example for converting the Boolean formula $F = (u_1 + u_2 + u_3) \wedge (u_1 + \bar{u}_2 + u_3) \wedge (u_1 + \bar{u}_2 + \bar{u}_3)$, and the dots are vertices in the minimum cover. From the minimum cover obtained we conclude that the truth values of u_1, u_2, and u_3 are true, true, and false, respectively.

Lemma 4: At least $K = 3m + n + 1$ vertices are needed for covering the simply connected polygonal region.

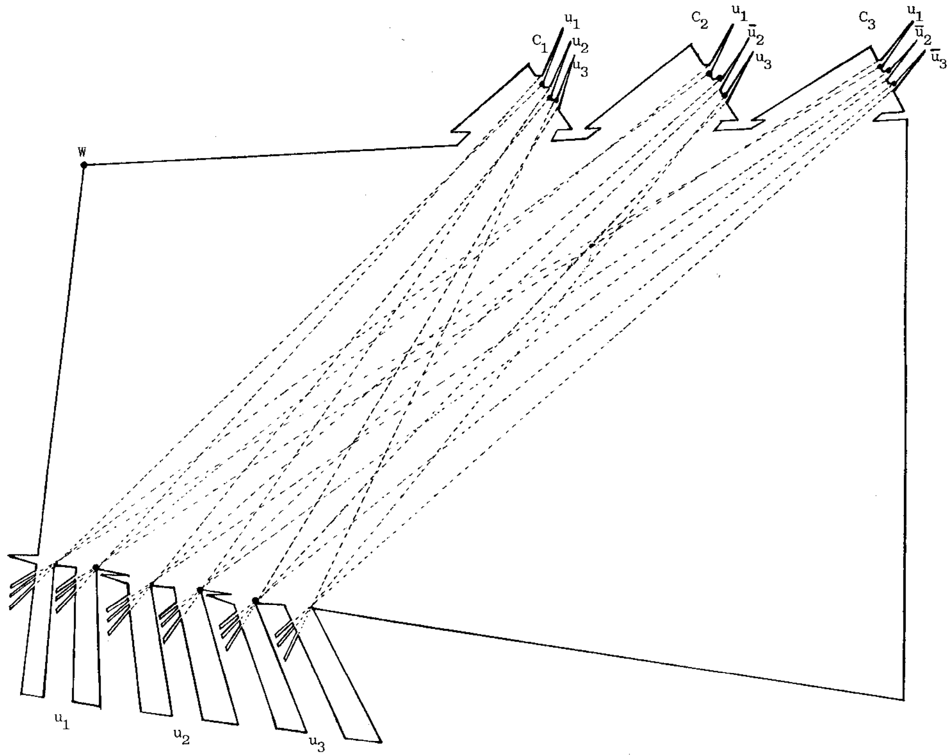

Fig. 8. Example and minimum cover.

Proof: At least $3m + n$ vertices are needed for covering $3m$ literal patterns, and $\Delta t_{i1}t_{i2}t_{i3}$, $i = 1, 2, \cdots, n$. At least one more vertex is needed to cover all the variable patterns' rectangles. Therefore, the lemma follows.

Lemma 5: The minimum number of vertices needed to cover the simply connected polygonal region is $K = 3m + n + 1$ if and only if C is satisfiable.

Proof: ($\leftarrow$) If C is satisfiable, then a truth value assignment to the variables exists such that each of the clauses in C has a truth value true. If u_i is true, then t_{i8} of the variable pattern u_i is included in the cover T, and either a_{h1} or a_{h3} of the literal pattern is included in T depending, respectively, on whether u_i or $\bar{u}_i$ appears in the clause C_h. If u_i is false, then t_{i5} is included in T and either a_{h1} or a_{h3} of the literal pattern is included depending, respectively, on whether $\bar{u}_i$ or u_i appears in the clause C_h. From the construction the regions defined by the consistency-check patterns and literal patterns are covered with $3m + n$ vertices. The remaining rectangular regions defined by the variable patterns can be covered by vertex W. Therefore, by including W in T we have a minimum cover with $K = 3m + n + 1$ vertices.

($\rightarrow$) Suppose a cover T with $K = 3m + n + 1$ vertices exists to cover the polygonal region. If W is not in T, K vertices are in no way sufficient to cover the regions defined by the $3m$ literal patterns and the $2n$ rectangles contained in the n variable patterns. Thus $K - 1 = 3m + n$ vertices are left to be considered.

In the polygonal region $3m$ literal patterns exist and n triangles $\Delta t_{i1}t_{i2}t_{i3}$, $i = 1, 2, \cdots, n$, each of which contains a distinguished point. We know that any vertex that covers a distinguished point cannot cover other distinguished points. Therefore, at least $3m + n$ vertices are needed to cover the above $3m + n$ subregions. To get a K-vertex cover each of the above $3m + n$ polygonal regions is allowed to be covered by one vertex. This means that for any literal pattern A, supposing A in C_h, either a_{h1} or a_{h3} must be included in T.

However, we cannot arbitrarily include in T any $3m$ vertices in these $3m$ literal patterns for they may make the n variable patterns inconsistent. The definition of consistency follows.

We say the any variable pattern L is *consistent* if all the consistency-check patterns connected to one of its two rectangles are covered by the $3m$ vertices in literal patterns and those connected to the other rectangle are not covered at all by the same $3m$ vertices; it is *inconsistent* otherwise.

In any variable pattern L, the number of consistency-check patterns connected to the first rectangle and second rectangle is the same; let it be q. The total number of consistency-check pattens in the variable pattern L is $2q$. From the construction the $3m$ vertices included in T can cover only q consistency-check patterns in L. If variable pattern L is not consistent, then some of the consistency-check patterns connected to the first rectangle and some connected to the second rectangle will not be covered. Then at least two extra vertices of L are needed to cover L. On the other hand, if it is consistent, one extra vertex is enough. Therefore, to get a K-vertex cover consistency of all variable patterns is required.

Since these $3m$ vertices of the literal patterns must satisfy the consistency requirement, the remaining n vertices chosen from the variable patterns can be determined. We know that these n vertices and vertex W cannot cover any $\Delta g_{h1}g_{h2}g_{h3}$, $h = 1, 2, \cdots, m$. Therefore, $\Delta g_{h1}g_{h2}g_{h3}$, $h = 1, 2, \cdots, m$, must be coverable by the $3m$ vertices. From Lemma 2, if they are covered free, then each c_i is satisfiable. This implies C is satisfiable. Once it is known that the instance of 3SAT is satisfiable, the truth value assignment to the variables can easily be determined from the consistency property possessed by the minimum cover.

Thus we have the following main results.

Theorem 1: The minimum vertex guard problem for simply connected polygonal regions is NP-hard.

Proof: It follows from Lemma 5 and from the fact that the construction of the simply connected polygonal region, as described earlier, from a given instance of 3SAT takes polynomial time.

Edge Guard of a Polygonal Region

In this section we consider the minimum edge guard problem for a simply connected polygonal region where the guard can move along an edge. Since the NP-hardness proof parallels that for the minimum vertex guard problem, we summarize it as follows.

Theorem 2: The minimum edge guard problem for simply connected polygonal regions is also NP-hard.

Proof: Use similar arguments except that slight modifications to the literal patterns, variable patterns, and W are needed. They are shown in Figs. 9–11. In Fig. 9 only edges $\overline{a_{h1}a_{h8}}$ and $\overline{a_{h3}a_{h12}}$ can cover the entire literal pattern a_h. In Fig. 10 edges $\overline{t_{i5}t_{i6}}$ and $\overline{t_{i8}t_{i9}}$ are the only edges that can cover $\Delta t_{i1}t_{i2}t_{i3}$ and one of the rectangles in the variable pattern t_i. Change W into $\overline{W_1W_2}$. W_1 is just like W in the proof of vertex guard problem. W_2 covers $\Delta W_2W_3W_4$.

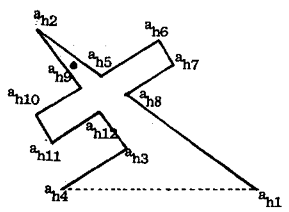

Fig. 9. Literal pattern for edge guards.

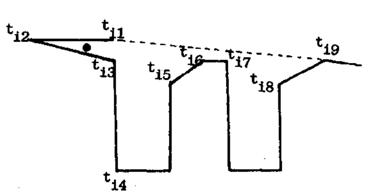

Fig. 10. Variable pattern for edge guards.

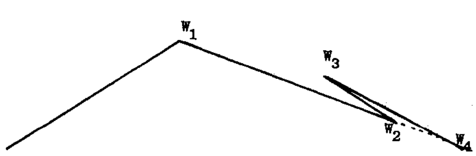

Fig. 11. Vertex W for edge guards.

IV. Point Guard Problem

In this section we consider the minimum point guard problem for simply connected polygons, which is similar to the minimum vertex guard problem except that the guards need not necessarily be fixed on the vertices. This problem can also be shown to be NP-hard using similar arguments as given earlier with some modifications [1]. The roles played in the proof by vertices a_{h1}, a_{h3}, t_{i5}, and t_{i8} for variable u_i should be replaced by their corresponding kernels. The shaded regions around t_{i5} and t_{i8} in Fig. 12, referred to as the kernels, are regions such that any point in these regions can cover $\Delta t_{i1}t_{i2}t_{i3}$ and the consistency-check patterns visible from t_{i5} and t_{i8}, respectively. We note that the kernels of the literal pattern a_h and the consistency-check patterns visible from a_{h1} and a_{h3}, respectively, are the vertices a_{h1} and a_{h3} themselves. Thus the point guards in the minimum cover are restricted to vertex W and these kernels in the simply connected polygonal region.

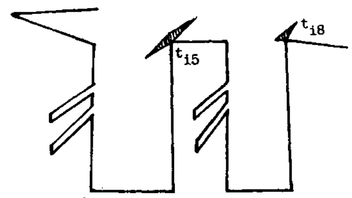

Fig. 12. Variable pattern for point guards.

Theorem 3: The minimum point guard problem for simply connected polygonal regions is NP-hard.

Since each point guard covers a region that is star-shaped, we have the following corollary.

Corollary: The problem of decomposing a simple polygon into the minimum number of star-shaped polygons whose union is the given polygon is NP-hard.

V. Conclusion

We have shown that the minimum vertex guard problem for a simply connected polygon is NP-hard. With slight modifications to the construction of the proof of the NP-hard result we have also shown that the minimum edge guard problem and minimum point guard problem for simple polygons are NP-hard. As a result of the NP-hardness of the minimum point guard problem, the problem of decomposing a simple polygon P into a minimum number of star-shaped polygons such that their union is P is also NP-hard.

Acknowledgment

The authors wish to thank Alok Aggarwal at Johns Hopkins University for useful comments on an earlier draft of this paper and for providing insight into the complexity result of the minimum point guard problem in Section IV, and the referees for constructive suggestions that greatly helped the presentation of the paper.

References

[1] A. Aggarwal, personal communication.

[2] N. Ahuja, R. T. Chien, R. Yen, and N. Birdwell, "Interface detection and collision avoidance among three dimensional objects," in *Proc. 1st Nat. Conf. Artificial Intelligence,* 1980, pp. 44–48.

[3] D. Avis and G. T. Toussaint, "An efficient algorithm for decomposing a polygon into star-shaped polygons," *Pattern Recognition*, vol. 13, pp. 395–398, 1981.

[4] ——, "An optimal algorithm for determining the visibility of a polygon from an edge," *IEEE Trans. Comput.*, vol. C-30, pp. 910–914, Dec. 1981.

[5] B. M. Chazelle and D. Dobkin, "Decomposing a polygon into its convex parts," in *Proc. 11th ACM Symp. Theory Comput.*, 1979, pp. 38–48.

[6] V. Chvatal, "A combinatorial theorem in plane geometry," *J. Comb. Theory, Ser. B*, vol. 18, pp. 39–41, 1975.

[7] H. Edelsbrunner, J. O'Rourke, and E. Welzl, "Stationing guards in rectilinear art galleries," *Comput. Vision, Graphics, Image Processing*, vol. 27, pp. 167–176, 1984.

[8] H.-Y. Feng and T. Pavlidis, "Decomposition of polygons into simpler components: Feature generation for syntactic pattern recognition," *IEEE Trans. Comput.*, vol. C-24, pp. 636–650, June 1975.

[9] S. Fisk, "A short proof of Chvatal's watchman theorem," *J. Comb. Theory Ser. B*, vol. 24, p. 374, 1978.

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

[11] R. Honsberger, *Mathematical Gems II*. Mathematical Assoc. of America, 1976.

[12] J. Kahn, M. Klawe, and D. Kleitman, "Traditional galleries require fewer watchmen," *SIAM J. Alg. Disc. Meth.*, vol. 4, pp. 194–206, June 1983.

[13] J. O'Rourke, "The complexity of computing minimum convex covers for polygons," in *Proc. 20th Allerton Conf.*, 1982, pp.75–84.

[14] ——, "An alternate proof of the rectilinear art gallery theorem," *J. Geometry*, vol. 211, pp. 118–130, 1983; also in Dep. Elec. Eng. and Comput. Sci., Johns Hopkins Univ., Baltimore, MD, Tech. Rep. 82/15, Dec. 1982.

[15] ——, "Galleries need fewer mobile guards: A variation on Chvatal's theorem," *Geometricae Dedicata*, vol. 14, pp. 273–283, 1983.

[16] J. O'Rourke and K. Supowit, "Some NP-hard polygon decomposition problems," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 181–190, Mar. 1983.

[17] J. Sack, "An $O(n \log n)$ algorithm for decomposing simple rectilinear polygons into convex quadrilaterals," in *Proc. 20th Allerton Conf.*, 1982, pp. 64–74.