Searching for Mobile Intruders in a Polygonal Region by a Group of Mobile Searchers*

Masafumi Yamashita
Department of Electrical Engineering
Faculty of Engineering
Hiroshima University
Kagamiyama, Higashi-Hiroshima 739, Japan
mak@se.hiroshima-u.ac.jp

Hideki Umemoto†
System Engineering Section
Engineering Design Division
Technology Center
Anritsu Corporation
Onna, Atsugi 243, Japan
umemoto@accpd3.anritsu.co.jp

Ichiro Suzuki
Department of Electrical Engineering and Computer Science
University of Wisconsin - Milwaukee
P.O. Box 784, Milwaukee, WI 53201, U.S.A.
suzuki@blatz.cs.uwm.edu

Tsunehiko Kameda
School of Computing Science
Simon Fraser University
Burnaby, B.C., V5A 1S6, Canada
tiko@cs.sfu.ca

*This work was supported in part by a Scientific Research Grant-in-Aid from the Ministry of Education, Science and Culture of Japan under Grant 07243219 and 07680360, the National Science Foundation under Grant IRI-9307506, the Office of Naval Research under Grant N00014-94-1-0284, the Natural Sciences and Engineering Council of Canada, and an endowed chair (KIFUKOZA) supported by Hitachi Ltd. at Faculty of Engineering Science, Osaka University.

†Part of this work was done by the author when he was at the Graduate School of Hiroshima University.

Abstract

The problem of searching for mobile intruders in a polygonal region by mobile searchers is considered. A searcher can move continuously inside a polygon holding a flashlight that emits a single ray of light whose direction can be changed continuously. The visibility of a searcher at any time instant is limited to the points on the ray. The intruders can move continuously with unbounded speed. We denote by $ps(P)$ the polygon search number of a simple polygon P, which is the number of searchers necessary and sufficient to search P. Let n, r, b and g be the number of edges, the number of reflex vertices, the bushiness, and the size of a minimum guard set of P, respectively. In this paper, we present matching upper and (worst case) lower bounds of $1 + \lfloor \log_3(2b+1) \rfloor$ on $ps(P)$. Also upper bounds on $ps(P)$ in terms of n, r and g are presented; $ps(P) \leq 1 + \lfloor \log_3(n-3) \rfloor, ps(P) \leq 1 + \lfloor \log_3 r \rfloor$, and $ps(P) \leq 2 + \lceil \log_2 g \rceil$. These upper bounds are tight or almost tight in the worst case, since we show that for any natural number $s \geq 2$, there is a polygon P such that $ps(P) = \log_3(n+1) = \log_3(2r+3) = 1 + \log_3(2g-1) = s$.

Keywords: polygon search problem, polygon search number, searchlight problem, graph search problem, art gallery problem, multi-robot system

Send all correspondences to: Ichiro Suzuki, +1-414-229-3718

1 Introduction

When a murder case occurs, how can we find the perpetrator hiding in town? The graph search problem [9, 11, 14] is the problem of clearing an initially contaminated graph by a number of searchers, and it has been studied as a formalization of the following problem [14]: "Suppose a man is lost and wandering unpredictably in a dark cave. A party of searchers who know the structure of the cave is to be sent in to find him. What is the minimum number of searchers needed to find the lost man regardless of how he behaves?" In the graph search problem, the cave is represented as a graph, and the searchers move over the graph edges continuously, i.e., at finite but variable speed. In this paper, we represent the cave as a polygon, rather than a graph, and consider the problem of searching for mobile intruders in the polygon by mobile searchers. We refer to this problem as the *polygon search problem*.

Variations of the polygon search problem have been considered in the literature using searchers of different capabilities. In [17], Suzuki and Yamashita introduced a mobile searcher, called the k-searcher, having k flashlights. Each flashlight emits, from the position of the searcher, a single ray of light that cannot penetrate the polygon boundary. The directions of the rays can be changed continuously with finite angular rotation speed, and the visibility of the searcher at any given time instance is limited to the points that are on at least one of the k rays. They also introduced a searcher, called the ∞-searcher, having a point light source who can see in all directions simultaneously. (Here, the searcher cannot see through the polygon boundary.) The problem of searching a polygon by a *single* such searcher is discussed in [4, 17], and some necessary conditions and sufficient conditions for the given polygon to be searchable are presented. However, many of these conditions are only necessary or sufficient, and even a necessary and sufficient condition for a simple polygon to be searchable by a single 1-searcher is left as an open problem. Another variation of the polygon search problem is the searchlight scheduling problem [16], in which the intruder must be found using the flashlights of *stationary* 1-searchers. In the two-guards problem [6, 7], one asks if it is possible for two mobile guards to walk in opposite directions starting from a point on the polygon boundary and meet at another point on the boundary, keeping themselves visible to each other throughout their walk. In our terminology, this corresponds to moving a 1-searcher along a boundary chain in such a way that the points at which the flashlight is aimed moves continuously on the other boundary chain. Hence, a given polygon is searchable by one 1-searcher, if the two-guards problem is solvable for the polygon.

There are other related problems, in which the goal can be thought of as finding a *stationary* intruder. The art gallery problem [8, 10, 12, 13] is the problem of placing stationary guards in a polygon so that every point in the interior of the polygon will be visible from at least one guard. For this problem, a tight upper (and lower) bound on the number of guards necessary to cover a given polygon with n vertices is known: (1) $\lfloor n/3 \rfloor$ for a general simple polygon, and (2) $\lfloor n/4 \rfloor$ for an orthogonal simple polygon. It has also been shown that determining the minimum number of guards to cover a given polygon (with or without holes) is NP-hard. The watchman route problem [2, 3] is another interesting problem and is defined as the problem of constructing a shortest path within a polygon such that every point in the interior of the polygon will be visible from at least one point on the path. The problem is NP-hard for general polygons (possibly with holes) [2, 3], and is solvable in $O(n^3)$ time [1] for any simple polygon with n vertices. A variation of this problem in which a shortest *tour* must be found that starts and ends at a given point on the boundary of a given simple polygon can be solved in $O(n^2)$ time [18].

In this paper, we discuss the problem of finding mobile intruders in a simple polygon using as few 1-searchers as possible. So in the sequel, a "searcher" means a 1-searcher.

The main results are the following. Let $ps(P)$ be the number of searchers necessary and sufficient to search a simple polygon P. We call $ps(P)$ the *polygon search number* of P. We present matching upper and (worst case) lower bounds of

$$1 + \lfloor \log_3(2b + 1) \rfloor$$

on $ps(P)$ in terms of the *bushiness* b of P, which is defined as the minimum number of triangles that share no edges with P over all triangulations of P.

We also present upper and lower bounds in terms of other measures of shape complexity. As for upper bounds, we have

1. $ps(P) \leq 1 + \lfloor \log_3(n - 3) \rfloor$,
2. $ps(P) \leq 1 + \lfloor \log_3 r \rfloor$, and
3. $ps(P) \leq 2 + \lceil \log_2 g \rceil$,

where n, r and g are the number of edges, the number of reflex vertices, and the size of a minimum guard set of P, respectively. These upper bounds are either tight or almost tight in the worst case, since we show that for each natural number $s \geq 2$, there is a polygon P such that $ps(P) = \log_3(n + 1) = \log_3(2r + 3) = 1 + \log_3(2g - 1) = s$.[1]

The upper bounds are obtained either by a reduction of the polygon search problem to the *graph search problem* [11], or by a decomposition of the given problem into a number of smaller related search problems, called *chasing*, *shooing*, and *corridor search problems*, that deal with a region with "entrances." (These problems are defined formally in Section 3.) The decomposition is often done by repeated applications of a search strategy called the *one-way sweep strategy*.

In Section 2, we introduce formally the polygon search problem and measures of shape complexity. In Section 3, we introduce three related search problems as well as basic search strategies that play a central role in constructing search schedules. Section 4 derives upper bounds on $ps(P)$ by constructing search schedules using the basic search strategies introduced in Section 3. In Section 5, we give lower bounds on $ps(P)$ by presenting a series of simple polygons achieving the bounds. The concluding remarks are found in Section 6.

2 Preliminaries

2.1 Problem Formulation

Let P be a simple polygon with n edges (or equivalently n vertices). We consider P as a set consisting of all points lying in P, and denote by ∂P the boundary of P. Two points p and $q \in P$ are said to be mutually *visible* if $\overline{pq} \subseteq P$. We denote by $V(x)$ the set of points in P that are visible from a point $x \in P$.

[1]Independently de Berg and Klein proved $ps(P) = \Theta(\log n)$ in the worst case for ∞-searchers [5].

Let us introduce a searcher who can walk within P holding a flashlight, where the flashlight emits a single light beam that is blocked as soon as it intersects with the exterior of P. The sight of the searcher at a given time instant is limited to the points on the ray. Suzuki and Yamashita [17] defined a k-searcher who holds k flashlights, and discussed search schedules of one k-searcher. In this paper we will discuss only 1-searchers, which is defined as follows.

Definition 1 Let P be a simple polygon and τ be a positive real number. A *schedule* of a searcher on P is a pair $\sigma = (\alpha, \beta)$ of two continuous functions $\alpha : [0, \tau] \rightarrow P$ and $\beta : [0, \tau] \rightarrow \mathcal{R}$, where $[0, \tau]$ is an interval of real time and $\mathcal{R}$ is the set of real numbers. A point $x \in P$ is *illuminated* at time $t \in [0, \tau]$ during the execution of σ if x lies in the intersection of $V(\alpha(t))$ and the ray emanating from $\alpha(t)$ in the direction $\beta(t)$. □

Functions $\alpha(t)$ and $\beta(t)$ specify the location of the searcher and the direction of the flashlight at time t, respectively. (The direction is measured in radian counter-clockwise from the positive x-axis.) Intuitively, the illuminated points are those that the searcher can "see" at any given time.

Definition 2 Let P be a simple polygon. A *schedule* of s searchers $S_1, S_2, \ldots, S_s$ on P is an s-tuple $\mathcal{S} = \langle \sigma_1, \sigma_2, \ldots, \sigma_s \rangle$, where $\sigma_i = (\alpha_i, \beta_i), i = 1, 2, \ldots, s$, is a schedule of S_i on P. Without loss of generality, we assume that all α_i's and β_i's are defined over the same interval $[0, \tau]$ of real time. □

Definition 3 Let $\mathcal{S}$ be a schedule of s searchers on P defined over $[0, \tau]$. A point $x \in P$ is said to be *contaminated* at time $t \in [0, \tau]$ during the execution of $\mathcal{S}$ if there exists a continuous function $I : [0, t] \rightarrow P$ such that $I(t) = x$, and that for any $t' \in [0, t]$ $I(t')$ is not illuminated by the ray of any searcher. A point that is not contaminated is said to be *clear*. A region $R \subseteq P$ is said to be *contaminated* if it contains a contaminated point; otherwise, it is *clear*. □

Intuitively, I represents the trajectory of an intruder, and a point x is contaminated at time t if an intruder who has never been illuminated can be located at x at time t.

Definition 4 Let $\mathcal{S}$ be a schedule of s searchers on P defined over the interval $[0, \tau]$. Then $\mathcal{S}$ is called a *search schedule* for P if all points of P are simultaneously clear at time τ. P is said to be *searchable* by s searchers if there exists a search schedule of s searchers for P. The *polygon search number* $ps(P)$ of P is the minimum number of searchers for which there exists a search schedule for P. □

In this paper, we discuss the problem of evaluating the polygon search number $ps(P)$ in terms of four measures of shape complexity that we will introduce in the next subsection.

Let $\sigma = (\alpha, \beta)$ be a schedule of a searcher on a simple polygon P defined over the interval $[0, \tau]$ of real time. Define α^R and β^R by

$$\alpha^R(t) = \alpha(\tau - t)$$

and

$$\beta^R(t) = \beta(\tau - t).$$

Letting $\sigma^R = (\alpha^R, \beta^R)$, σ^R is also a schedule of a searcher on P defined over $[0, \tau]$.

Definition 5 Let $\mathcal{S} = \langle \sigma_1, \sigma_2, \ldots, \sigma_s \rangle$ be a search schedule of s searchers for P defined over $[0, \tau]$. Then $\mathcal{S}^R = \langle \sigma_1^R, \sigma_2^R, \ldots, \sigma_s^R \rangle$ is called the *reverse schedule* of $\mathcal{S}$. □

Theorem 1 *If $\mathcal{S}$ is a search schedule of s searchers for P, so is the reverse schedule $\mathcal{S}^R$.*

Proof Suppose that $\mathcal{S}^R$ is not a search schedule for P. Then there is a continuous function $I^R : [0, \tau] \rightarrow P$ such that $I^R(t)$ is not illuminated by the ray of any searcher at any time $t \in [0, \tau]$. Define a continuous function $I : [0, \tau] \rightarrow P$ as the reverse of I^R by

$$I(t) = I^R(\tau - t).$$

Since the ray of a searcher S obeying $\mathcal{S}^R$ illuminates a point $x \in P$ at time t if and only if the ray of S obeying $\mathcal{S}$ illuminates x at $tau - t$, $I^R(t)$ is illuminated by a ray if and only if $I(\tau - t)$ is illuminated by the ray. Hence $I(t)$ is not illuminated at any time $t \in [0, \tau]$, a contradiction. □

In the following sections, we may describe a schedule of searchers S by using expressions such as "place S at a point x" and "aim its ray at a point x", instead of specifying functions α and β explicitly.

2.2 Measures of Shape Complexity

Let P be a simple polygon with n edges. Besides n and the number r of reflex vertices, we use two additional measures of shape complexity of P; the size g of a minimum guard set and the bushiness b.

A finite set G of points in P is called a *guard set* of P if every point $x \in P$ is visible from some point $y \in G$ (i.e., $x \in V(y)$). We define g as the minimum size $|G|$ among all guard sets G of P. To define the bushiness b, we use the concept of triangulation. A triangulation of P is a decomposition of P into $n - 2$ non-overlapping triangles by $n - 3$ nonintersecting diagonals between pairs of mutually visible vertices of P. The dual of every triangulation of P is a tree T having a vertex for each triangle and an edge between those vertices that correspond to two triangles that share a diagonal. See Figure 2.2. The degree of a vertex in T is one, two or three, if the corresponding triangle shares two, one, or no edges with P, respectively. A triangulation is said to be *thin* if its dual tree has the smallest number of degree-three vertices among all triangulations of P [15]. An $O(n^3)$ time dynamic programming algorithm for computing a thin triangulation of a simple polygon with n vertices is given in [15]. The *bushiness* of P is defined to be the number of degree-three vertices in the dual tree of a thin triangulation of P.

3 Basic Search Strategies

In the course of solving the polygon search problem, we encounter three closely related problems—the chasing problem, the shooing problem, and the corridor search problem. Unlike the polygon search problem that deals with a room without entrances, these problems deal with a room with entrances. They are introduced below.

Given a simple polygon P and an edge e, the *chasing problem* with respect to e is the problem of discovering all intruders in room P without letting any of them escape from entrance e. Formally, the goal is to clear P in such a way that edge e remains clear at

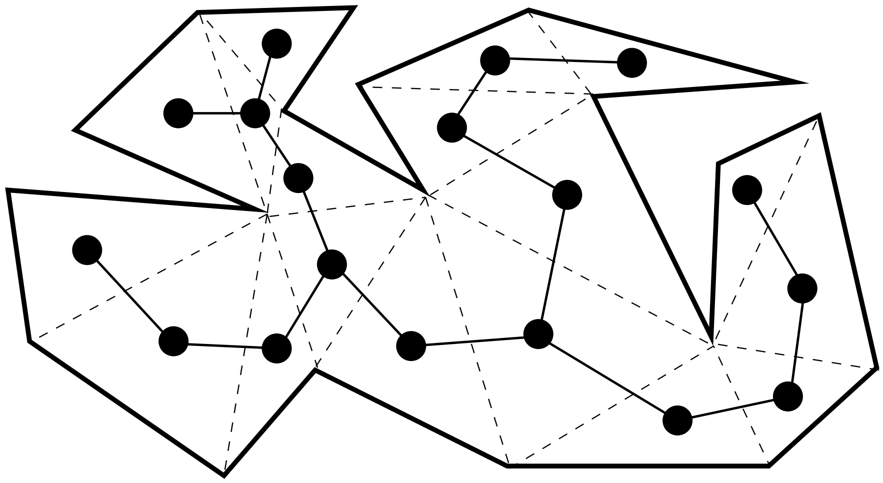


Figure 1: A triangulation of a simple polygon and its dual tree.

any time during the search, and hence a search schedule for P solves the chasing problem with respect to e if and only if e remains clear at any time. By definition, without loss of generality we may assume that when a chasing schedule starts, there is a searcher at an endpoint of e and its ray is aimed at the other endpoint of e (to clear e).

Given a simple polygon P and an edge e, the *shooing problem* with respect to e is the problem of clearing room P with entrance e through which a new intruder may enter into P at any time (unless the entire e is illuminated). Formally, the goal is to clear P under an additional assumption that at any time, any point in e that is not illuminated is considered contaminated. By definition, without loss of generality we may assume that when a shooing schedule finishes, there is a searcher at an end point of e and its ray is aimed at the other endpoint of e (to clear e).

Theorem 2 *Schedule $\mathcal{S}$ is a chasing schedule for P with respect to e, if and only if its reverse schedule $\mathcal{S}^R$ is a shooing schedule for P with respect to e.*

Proof Without loss of generality, we assume that $\mathcal{S}$ (and therefore $\mathcal{S}^R$) is a search schedule for P. Schedule $\mathcal{S}$ is not a chasing schedule for P with respect to e, if and only if there is a continuous function $I : [0, \tau_f] \to P$ such that

1. $\tau_f \in [0, \tau]$,

2. $I(\tau_f) \in e$, and

3. $I(t)$ is not illuminated by the ray of any searcher following $\mathcal{S}$ for any $t \in [0, \tau_f]$,

i.e., there could be an intruder who escaped from P at time τ_f. Define the reverse I^R of I by

$$I^R(t) = I(\tau - t).$$

Then I satisfies the three condition listed above, if and only if I^R satisfies

1. $\tau - \tau_f \in [0, \tau]$,

2. $I^R(\tau - \tau_f) \in e$, and

3. $I^R(t)$ is not illuminated by the ray of any searcher following $\mathcal{S}^R$ for any $t \in [\tau - \tau_f, \tau]$,

since the ray of a searcher S following $\mathcal{S}$ illuminates a point $x \in P$ at time t if and only if the ray of S following $\mathcal{S}^R$ illuminates x at $\tau - t$. The three conditions for I^R imply that $\mathcal{S}^R$ is not a shooing schedule for P with respect to e, since there could be an intruder who entered P through a point in e at time $\tau - \tau_f$ and remained undetected until τ; his move is given by I^R. □

Given a simple polygon P and two edges e and e', the *corridor search* problem for P with respect to "entrance" e and "exit" e' is the problem of designing a search schedule for P which is both a chasing schedule for P with respect to e and a shooing schedule for P with respect to e'. That is, the goal is to clear P in such a way that e is kept clear at any time during the search, under an additional assumption that at any time, any point in e' that is not illuminated is considered contaminated. (Crass, et al. [4], discussed the corridor search problem for a *single* k-searcher, and characterized the class of polygonal regions for which there is a search schedule.)

By Theorem 2, a schedule $\mathcal{S}$ is a corridor search schedule for P with respect to entrance e and exit e' if and only if the reverse schedule $\mathcal{S}^R$ is a corridor search schedule for P with respect to entrance e' and exit e. Hence, we may ignore the distinction between the entrance and exit when discussing the corridor search problem.

Theorem 3 *Suppose that a given simple polygon P is partitioned into two simple polygons X and Y by a segment $\overline{uv}$, where $u, v \in \partial P$. Then P is searchable by s searchers if there is a chasing (and therefore shooing) schedule of s searchers for each of X and Y with respect to $\overline{uv}$.*

Proof The claim follows from Theorem 2 immediately: We can clear P by executing a chasing schedule for Y followed by a shooing schedule for X. □

The chasing (or shooing) problem for a simple polygon P with respect to an edge e can be decomposed further by using the following theorem.

Theorem 4 *Suppose that a given simple polygon P is partitioned into two simple polygons X and Y by a segment $\overline{u'v'}$, where $u', v' \in \partial P$. Let $e = (u, v)$ be an edge in Y such that $e \neq \overline{u'v'}$. Then the chasing (and therefore shooing) problem for P with respect to e is solvable by s searchers if both the chasing (and therefore shooing) problem for X with respect to (u', v') and the corridor search problem for Y with respect to e and $\overline{u'v'}$ are solvable by s searchers.*

Proof A shooing schedule for X with respect to (u', v') followed by a corridor search schedule for Y with respect to (u', v') and e is a shooing schedule for P with respect to e. □

Our method for constructing a search schedule is to decompose the original problem into smaller chasing, shooing and corridor search problems using Theorems 3 and 4 until each subproblem becomes tractable. In what follows, we will use these theorems frequently without explicitly referring to them.

We now introduce a search strategy called *one-way sweep*, which we will use as a main tool for constructing search schedules in the subsequent sections. Let P be a simple polygon, and u a point on the boundary of P. (Point u may or may not be a vertex of P.) Let v and w be the vertices of P adjacent to u clockwise and counterclockwise along ∂P, respectively. See Figure 2 for illustration.

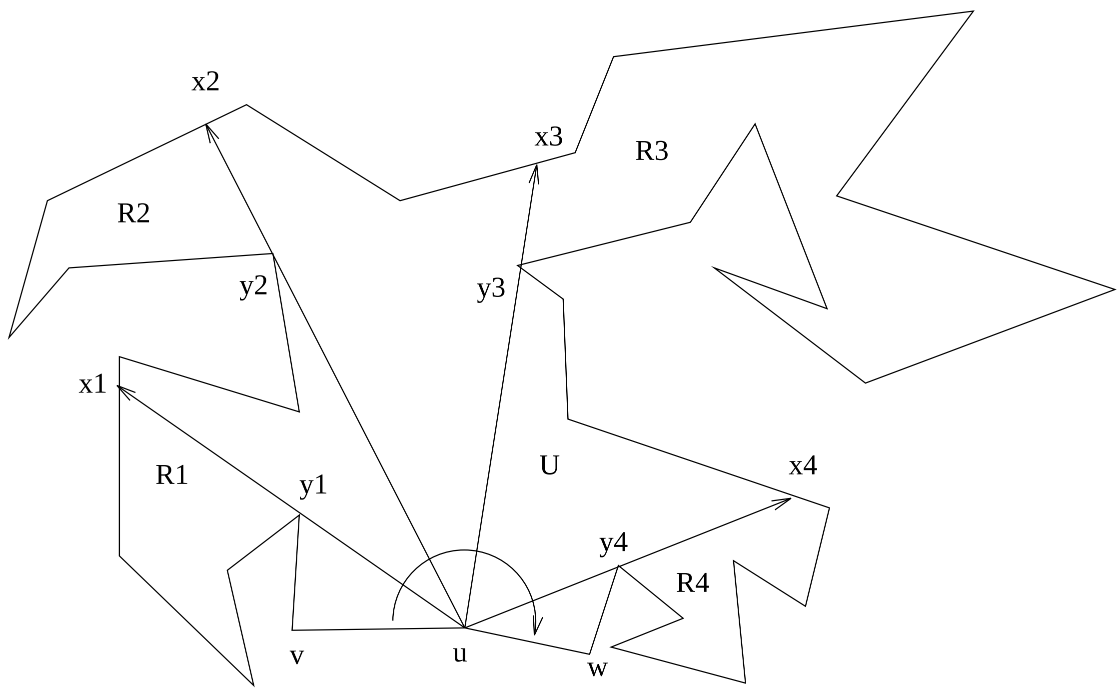


Figure 2: One-way sweep strategy.

Consider region $P - V(u)$. It consists of a number of connected subregions $R_1, R_2, \ldots, R_k$ appearing in clockwise order. We call each R_i a *pocket* (with respect to u), and the edge of R_i that separates R_i from $P - R_i$ is called the *lid* of R_i and is denoted by $e(R_i)$. In Figure 2, $P - V(u)$ consists of four subregions R_1, R_2, R_3 and R_4, and $e(R_i) = (x_i, y_i)$ for $i = 1, 2, 3, 4$. The *one-way sweep strategy* (OWSS) of s (≥ 2) searchers for P performed from u clockwise is the following:

1. Place a searcher S at u and aim its ray F at v.

2. Turn F clockwise until it is aimed at the lid $e(R_1)$ of an R_1, and then clear R_1 using the remaining $s - 1$ searchers. Clear $R_2, \ldots, R_k$ in this order in a similar manner by continuing to turn F clockwise and stopping it temporarily when it illuminates the lids of these regions.

3. When R_k is cleared, turn F clockwise until it is aimed at w.

Point u from which OWSS is executed is called the *pivot*. Analogously, we can define OWSS in counter-clockwise direction in which F is turned counter-clockwise. OWSS by itself is not a search schedule since it does not specify a complete schedule. However, if each R_i is searchable by at most $s-1$ searchers, then OWSS specifies a complete search schedule of s searchers for P. In this sense it is a scheme for reducing the given problem into smaller problems of searching R_i by $s-1$ searchers. When u is a vertex, OWSS can be a chasing or a shooing schedules for P with respect to $e=(u,v)$, depending on the direction in which it is executed.

OWSS as described above fails to clear P using s searchers if some R_i's require more than $s-1$ searchers. One special case that can be handled with s searchers is the following. Suppose that the shooing problem for R_i with respect to $e(R_i)=(x_i,y_i)$ is solvable by s searchers. The segment $\overline{ux_i}$ partitions P into three (not necessarily connected) regions. (Notice that u, y_i and x_i are collinear.) $\overline{x_iy_i}$ separates R_i from $P-R_i$ and $\overline{uy_i}$ partitions $P-R_i$ into two regions X and Y, as is shown in Figure 3. If X is searchable by $s-1$ searchers and the chasing problem for Y with respect to $\overline{ux_i}$ is solvable by s searchers, then we can use the following strategy to clear P.

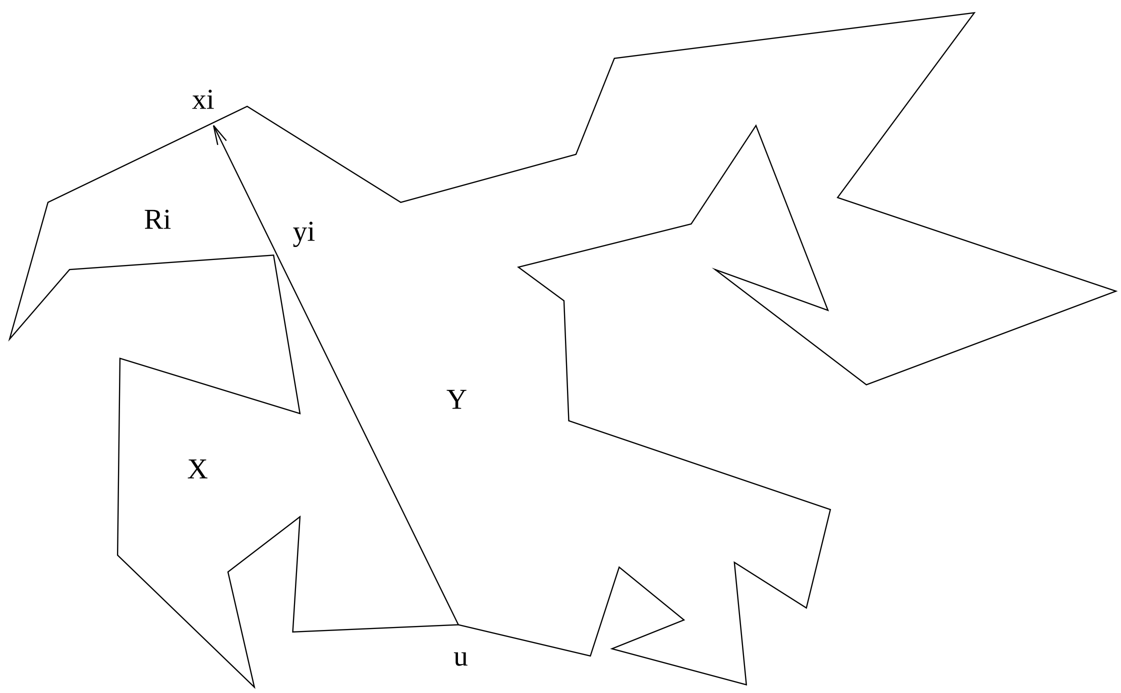


Figure 3: Reduction strategy.

1. Execute a shooing schedule by s searchers for R_i with respect to $e(R_i)$. Assume that searcher S is at an endpoint of $e(R_i)$ when the search is completed.
2. Place a searcher $S'(\neq S)$ at u and aim its ray at x_i. (Note that S can now be relocated without recontaminating R_i.)
3. Execute a search schedule for X using the remaining $s-1$ searchers.

4. Execute a chasing schedule for Y with respect to $\overline{ux_i}$ using all s searchers.

Therefore the original problem of searching P by s searchers is reduced to the shooing problem for R_i with respect to (x_i, y_i) and the chasing problem for Y with respect to (u, x_i), both by s searchers. A similar strategy is applicable when Y is searchable by $s-1$ searchers and the chasing problem for X with respect to $\overline{uy_i}$ is solvable by s searchers. Although both X and Y can require s searchers in general, this special case occurs often in the discussions that follow.

4 Upper Bounds on Polygon Search Numbers

4.1 The Number of Edges and the Bushiness

In this subsection, we derive upper bounds on the search number $ps(P)$ of any simple polygon P with the number of edges n and the bushiness b, using the concept of graph search. We start with a brief introduction of the graph search problem.

The "edge search" version of the graph search problem is the following [11]: Given a connected graph in which all edges are initially "contaminated," make all edges simultaneously "clear" by a sequence of the following basic operations: (1) placing a searcher on a vertex, (2) removing a searcher from a vertex, and (3) moving a searcher from one vertex to another along an edge. A contaminated edge $e = (u, v)$ becomes clear if we place a searcher on u as a guard and move a second searcher along e from u to v. If all edges incident on u other than e are already clear, however, then we can dispense with the second searcher and clear e simply by moving the guard along e from u to v. A clear edge remains clear until a searcher-free path to a contaminated edge is formed, in which case the edge is recontaminated. A *search strategy* is a sequence of the basic operations that clears an initially contaminated graph. The *graph search number* $es(G)$ of a graph G is the minimum number of searchers for which a search strategy exists.

Lemma 1 [Parsons [14]] *Let G be a connected graph. If H is a connected subgraph of G, then $es(H) \leq es(G)$.* □

Definition 6 Given a tree $T = (V, E)$ and a vertex $v \in V$, a branch at v is a maximal subtree of T that has v as a leaf. □

Lemma 2 [Parsons [14]] *For any tree T and integer $k \geq 1$, $es(T) \geq k+1$ if and only if T has a vertex at which there are three or more branches that have graph search number k or larger.* □

The next lemma follows from Lemma 2 [11].

Lemma 3 *Let T be a tree with $m \geq 2$ vertices. Then $es(T) \leq 1 + \lfloor \log_3(m-1) \rfloor$.* □

Definition 7 A branch with graph search number k is called a k-branch. A path $v_1, v_2, \ldots, v_l$ of two or more vertices having the following properties is said to be a k-avenue:

1. v_1 (or v_l) has exactly one k-branch containing v_2 (or v_{l-1}).

2. For any $2 \leq i \leq l-1$, v_i has exactly two k-branches, one containing v_{i-1} and the other v_{i+1}. □

Lemma 4 [Megiddo, et al. [11]] *For any tree T with $es(T) = k \geq 1$, either*

1. *there exists a vertex $v \in V$ such that search numbers of all branches at v are smaller than k, or*

2. *T has a unique k-avenue.* □

Now, we relate the polygon search number $ps(P)$ to the graph search number $es(T)$ for some T.

Lemma 5 *Let P be a simple polygon. For some points x and y on ∂P such that $\overline{xy} \subseteq P$, let Q be a subpolygon of P cut off by $\overline{xy}$. Then $ps(Q) \leq ps(P)$.*

Proof We give a proof outline. Let $\mathcal{S}$ be a search schedule of s searchers for P. Suppose that we execute $\mathcal{S}$ for Q, in such a way that, at any time instant, if a searcher i is at $p \in P - Q$ illuminating a segment $\overline{pq}$ such that $q \in Q$ (while executing $\mathcal{S}$ for P), then i instead occupies the intersection $\overline{xy} \cap \overline{pq}$ and illuminates $\overline{pq} \cap Q$ (while executing $\mathcal{S}$ for Q). That is, the searchers simulate the process of clearing Q while clearing P, without leaving Q. It is easy to show that this is always possible. Thus Q is searchable by s searchers. □

Lemma 6 *Let T be the dual tree of a triangulation of a simple polygon P. Then $ps(P) \leq es(T)$.*

Proof The proof is by induction on $es(T)$.
(Basis) If $es(T) = 1$, then T is a single path. So a single searcher can clear all triangles in the triangulation of P in the order they appear along the path, as is shown in Figure 4. So $ps(P) = 1$, and hence $ps(P) \leq es(T)$).

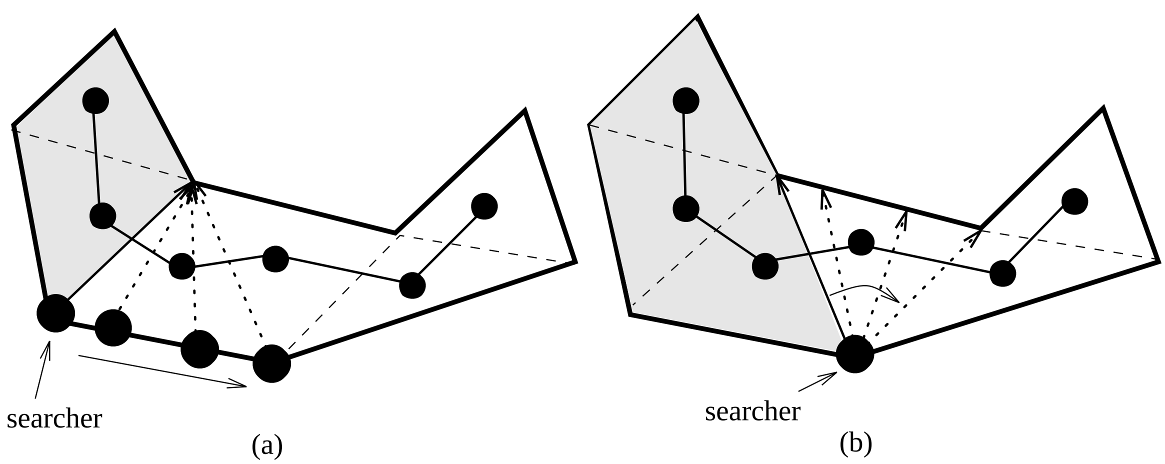


Figure 4: A polygon searchable by a single searcher.

(Induction) Assume that for some $i \geq 1$, $es(T) \leq i$ implies $ps(P) \leq es(T)$ for any simple polygon P and the dual tree T of one of its triangulations. Let P be a simple polygon such that $es(T) = i+1$, where T is the dual tree of one of its triangulations. Then by Lemma 4, one of the following two cases occurs:

(Case 1) There exists a vertex v such that the search number of any branch at v is smaller than $i+1$.

(Case 2) T has a unique $(i+1)$-avenue.

Let us examine each case.

(Case 1) Let $\triangle_v$ be the triangle corresponding to vertex v, and let P_1, P_2, and P_3 be the subpolygons of P corresponding to the branches at v. (See Figure 5.) Note that P_1, P_2 and P_3 contain $\triangle_v$ as a subpolygon. Consider any branch B of v, and let B' be the tree obtained from B by removing v. Then $es(B') \leq es(B) \leq i$ by Lemma 1, and hence by the induction hypothesis, each $P_i - \triangle_v$, $i = 1, 2, 3$, is searchable by i searchers. Then by Lemma 5, every subpolygon Q of $P_1 - \triangle_v$, $P_2 - \triangle_v$ and $P_3 - \triangle_v$ cut off by a ray emanating from a corner of $\triangle_v$ is searchable by at most i searchers. So as is shown in Figure 5, P can be cleared by OWSS using $i+1$ searchers, in which one searcher placed at a corner of $\triangle_v$ sweeps P by a ray and the other i searchers clear each pocket created by the ray. (See the dotted arrows in Figure 5.) (End of Case 1)

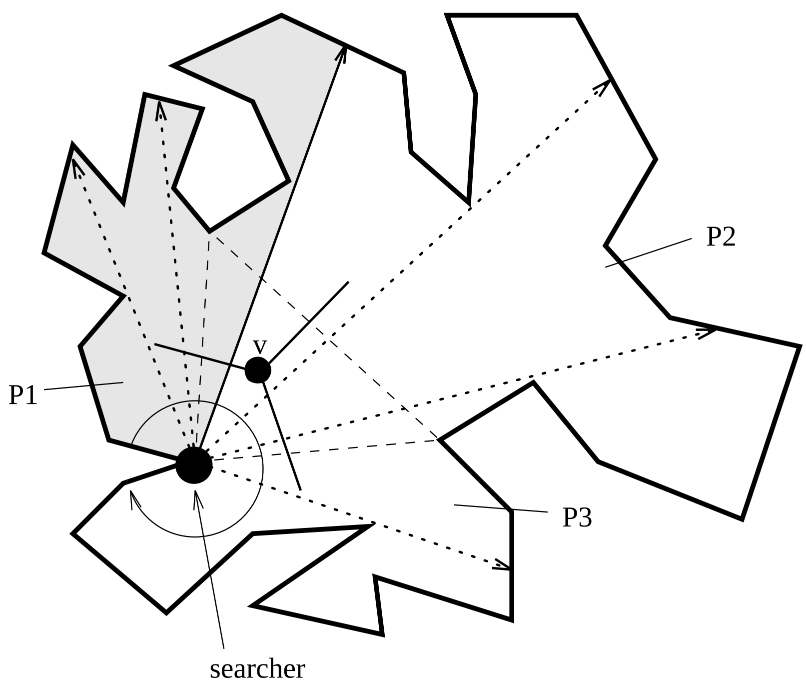


Figure 5: OWSS from a corner of $\triangle_v$.

(Case 2) Let v_1, v_2, ..., v_l be the vertices of the $(i+1)$-avenue of T, and $\triangle_1$, $\triangle_2$, ..., $\triangle_l$ the triangle corresponding to them, respectively. Since each vertex in T has degree at most 3, the following hold.

1. v_1 has at most two branches (not containing v_2), $B_{1,1}$ and $B_{1,2}$, each with graph search number at most i.

2. for each j, $1 < j < l$, v_j has at most one branch (containing neither v_{j-1} nor v_{j+1}), B_j, with graph search number at most i.

3. v_l has at most two branches (not containing v_{l-1}), $B_{l,1}$ and $B_{l,2}$, each with graph search number at most i.

Now, as we did in the proof for Case 1, we can show the following.

1. If we place a searcher at a corner of $\triangle_1$ and cut off a subpolygon Q out of the subpolygons corresponding to $B_{1,1} - \{v_1\}$ and $B_{1,2} - \{v_1\}$, then Q is searchable by i searchers.

2. For each j, $1 < j < l$, if we place a searcher at any corner of $\triangle_j$ and cut off a subpolygon Q out of the subpolygon corresponding to $B_j - \{v_j\}$, then Q is searchable by i searchers.

3. If we place a searcher at a corner of $\triangle_l$ and cut off a subpolygon Q out of the subpolygons corresponding to $B_{l,1} - \{v_l\}$ and $B_{l,2} - \{v_l\}$, then Q is searchable by i searchers.

Thus as is shown in Figure 6, we can clear P using OWSS repeatedly along path v_1, v_2, $\ldots$, v_l: Two searchers alternately sweep the subpolygons corresponding to $B_{1,1} \cup B_{1,2}$, B_2, $\ldots$, B_{l-1}, and $B_{l,1} \cup B_{l,2}$ in this order from opposite sides of the path depending on the directions in which the path "turns," and the remaining i searchers (including the one that is not currently sweeping in OWSS) clear every subpolygon cut off by the ray of the searcher that is currently sweeping. (End of Case 2)
Therefore, $ps(P) \leq es(T)$. This concludes the induction. □

Theorem 5 *For any simple polygon P with n edges, $ps(P) \leq 1 + \lfloor \log_3(n-3) \rfloor$.*

Proof By Lemmas 3 and 6, $ps(P) \leq es(T) \leq 1 + \lfloor \log_3(m-1) \rfloor$, where T and m are the dual graph of a triangulation of P, and the number of vertices of T, respectively. Since $m = n - 2$, we have $ps(P) \leq 1 + \lfloor \log_3(n-3) \rfloor$. □

The *reduction* of a graph G is the graph G' obtained from G by repeatedly replacing any degree-two vertex u and its two incident edges by a single edge connecting the two neighbors of u, until no degree-two vertices remain. We have:

Lemma 7 [Megiddo, et al. [11]] *A graph and its reduction have the same search number.* □

Theorem 6 *For any simple polygon P with bushiness b, $ps(P) \leq 1 + \lfloor \log_3 (2b+1) \rfloor$.*

Proof Let T be the dual tree of a thin triangulation of P, and T' the reduction of T. Since T' has b degree-three vertices and no degree-two vertices, there are $2b+2$ vertices in T'. Therefore by Lemmas 3, 6 and 7,

$$ps(P) \leq es(T) = es(T') \leq 1 + \lfloor \log_3(2b+1) \rfloor.$$

□

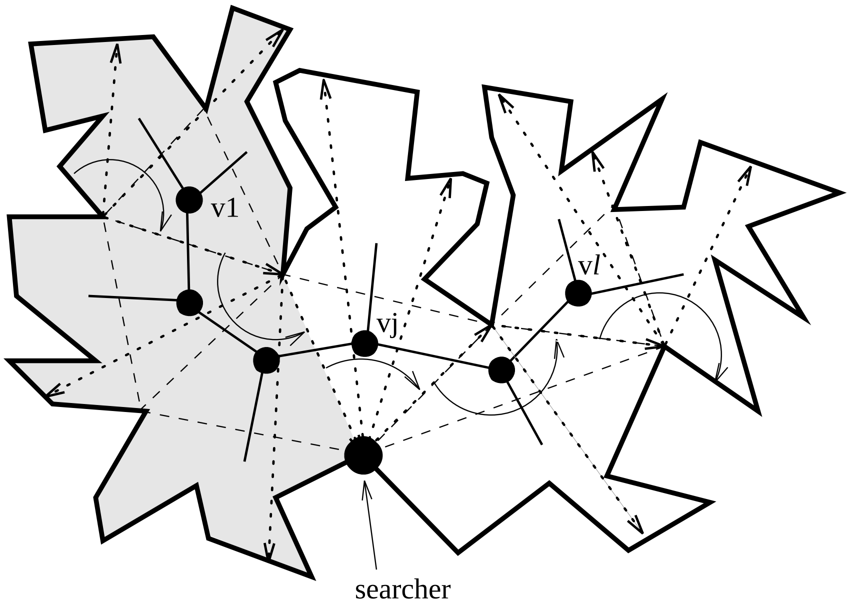


Figure 6: A sequence of OWSS to clear P. Only one of the two searchers that perform OWSS is shown.

4.2 The Number of Reflex Vertices

The objective of this subsection is to show the following theorem that presents an upper bound on $ps(P)$ in terms of the number r of reflex vertices in a simple polygon P. If $r = 0$, P is convex and therefore $ps(P) = 1$. So we consider the case where $r \geq 1$. In this subsection, instead of using a transformation to the graph search problem, we directly construct a search schedule based on recursive applications of OWSS.

Theorem 7 *For any simple polygon P with $r \geq 1$ reflex vertices, $ps(P) \leq 1 + \lfloor \log_3 r \rfloor$.*

Proof It suffices to prove the inequality for all r of the form $r = 3^s - 1$ for some natural number $s \geq 1$. This is because any polygon Q with r' reflex vertices (where $3^{s-1} - 1 \leq r' < 3^s - 1$) can be obtained from a polygon with $3^s - 1$ reflex vertices by a cutting off operation as described in Lemma 5. The proof is by induction on s.
(Basis) We show that every simple polygon with two reflex vertices is searchable by one searcher. Let u and v be two reflex vertices of P. They are either adjacent or not. The case in which u and v are adjacent is illustrated in Figure 7(a), which is easily searchable by one searcher as follows:

1. Place the searcher at u and aim the ray at x.

2. Turn the ray counterclockwise until it is aimed at y.

3. Move the searcher to v, aiming the ray at y.

4. Turn the ray counterclockwise until it is aimed at z.

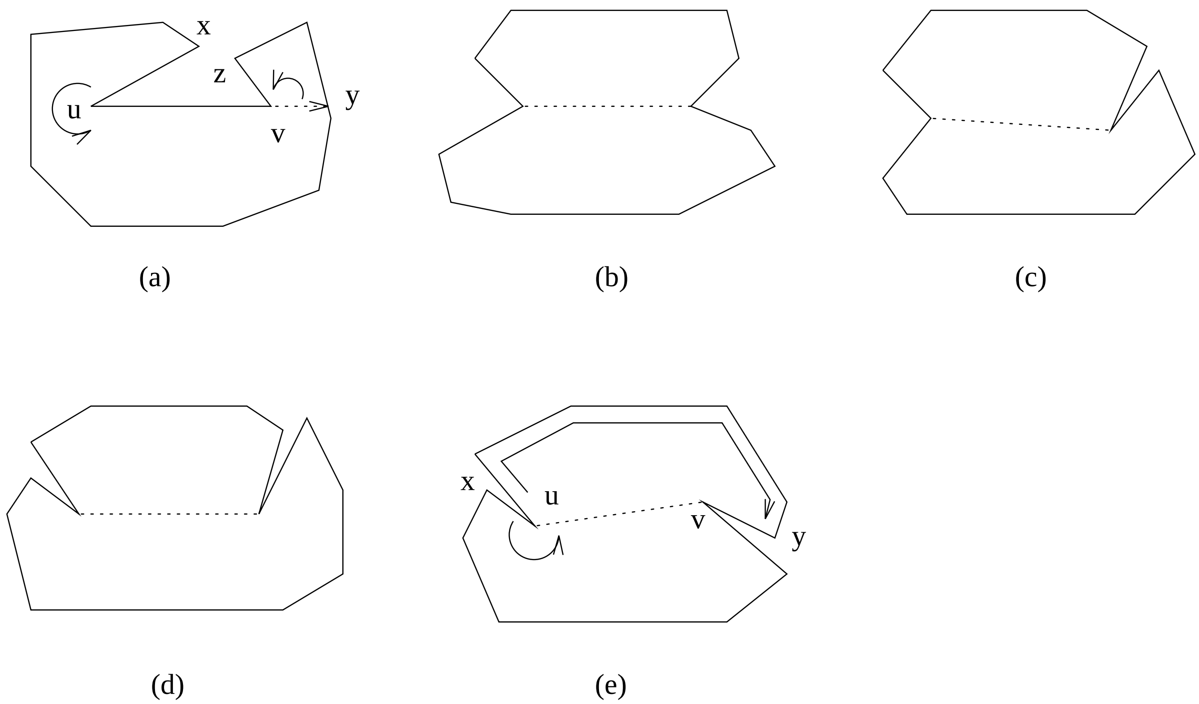


Figure 7: The classification of simple polygons with two reflex vertices.

When u and v are not adjacent, we cut P into two polygons by segment $\overline{uv}$ and classify P in terms of the number of reflex vertices that each of the two subregions has. There are four different types illustrated in Figures 7(b)-(e), and for each, one can easily find a search schedule of one searcher. For example, polygons of type (e) can be searched by one searcher as follows. Construction of a search schedule for other types are equally easy and is thus omitted.

1. Place the searcher at u and aim the ray at x.
2. Turn the ray counterclockwise until it is aimed at v.
3. Move the searcher on the boundary of upper subregion from u to y while aiming the ray at v.

(Induction) Assuming that every simple polygon with at most $3^s - 1$ reflex vertices is searchable by s searchers, we construct a search schedule of $s + 1$ searchers for any given simple polygon P with $3^{s+1} - 1$ reflex vertices.

Let us fix an arbitrary point $u \in \partial P$. Let $R_1, \ldots R_k$ be the connected regions comprising $P - V(u)$ (or pockets with respect to u). We assume that they occur in the clockwise order around u. For $i = 1, 2, \ldots, k$, by $e(R_i)$ and $r(R_i)$ we denote the lid of R_i and the number of reflex vertices in R_i, respectively. In what follows we may identify the pockets with their indices. (Later we will use a sequence of natural numbers as the index of a subregion inside a pocket.)

There may be some i's such that $r(R_i) \geq 3^s$. For each of such i's, we fix an endpoint v of $e(R_i)$ and repeat the same process: If there are h pockets comprising $R_i - V(v)$, then we label them $i.1, \ldots, i.h$. Figure 8 illustrates a case where there are four pockets R_1, ..., R_4 comprising $P - V(u)$ and pocket R_1 is decomposed further with respect to v assuming that $r(R_1) \geq 3^s$. The pockets comprising $R_1 - V(v)$ are labeled 1.1, 1.2 and 1.3. Regions R_2, R_3 and R_4 are not decomposed further because they are assumed to have fewer than 3^s reflex vertices. This process is applied recursively until no further partitions are needed. Obviously this procedure terminates in a finite number of steps.

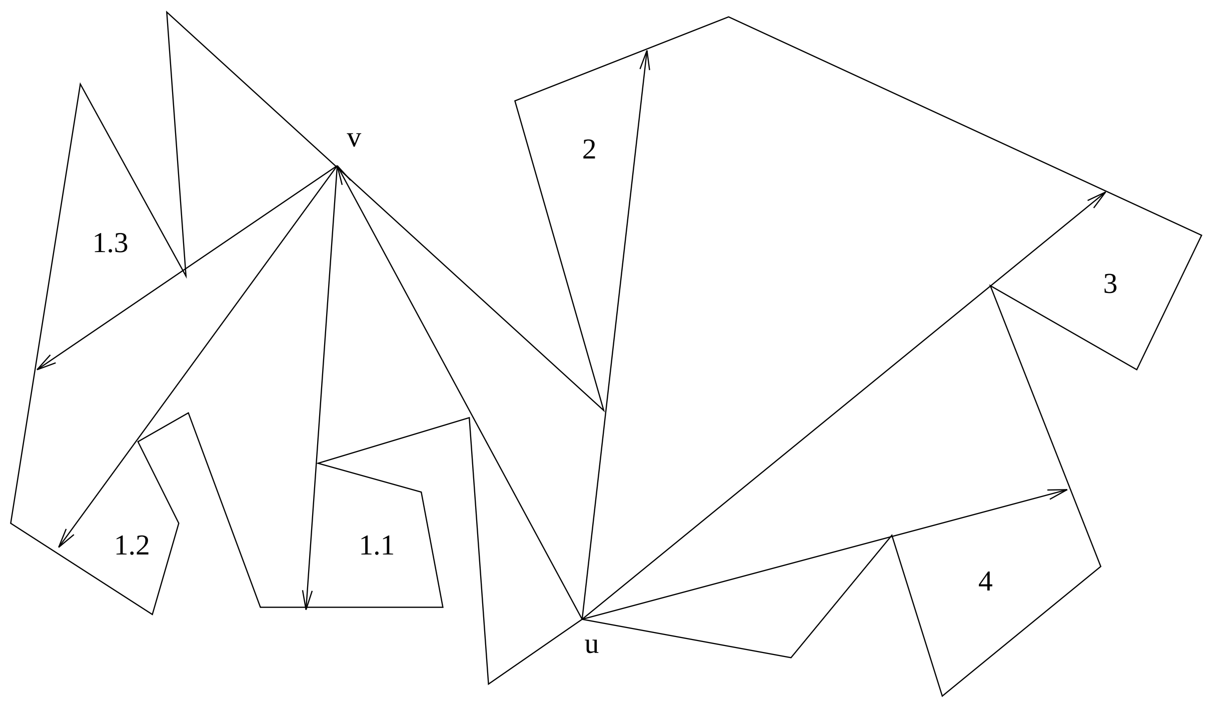


Figure 8: A partition of P.

We may represent the resulting partition by a rooted tree $T(P)$, called a *decomposition tree* of P, where each node is labeled with the index of the region it represents. In the case of Figure 8, the root labeled λ represents P itself. The root has four children labeled 1, ..., 4, and vertex 1 has three children 1.1, 1.2 and 1.3. That is, in general, $T(P)$ is constructed from the set of subregions by adding an arc from a node labeled $\mathbf{x}$ to nodes labeled $\mathbf{x}.i$. Of course, $T(P)$ is not determined uniquely in general; there may be many different trees depending on the selection of u and v's. Note that the parent of each leaf of $T(P)$ is associated with a subpolygon with 3^s or more reflex vertices.

Consider the rooted tree $T^*(P)$ constructed from $T(P)$ by removing all leaves (and the arcs incident to them). Then $T^*(P)$ has at most two leaves, as is shown below. By definition, if a vertex $\mathbf{x}$ has k children $\mathbf{x}.1, \mathbf{x}.2, \ldots, \mathbf{x}.k$ then

$$r(\mathbf{x}) \geq \sum_{i=1}^{k} r(\mathbf{x}.i) + k$$

holds. (Here we use $r(\mathbf{x})$ to denote the number of reflex vertices in the region with index $\mathbf{x}$.) So if $T^*(P)$ had more than two leaves then P would have at least 3^{s+1} reflex vertices, a contradiction.

There are two cases to consider; when $T^*(P)$ has exactly one leaf ℓ (i.e., $T^*(P)$ is a chain), and when it has two leaves ℓ and ℓ'.

(Case 1) Suppose that $T^*(P)$ has exactly one leaf ℓ, where $\ell = j_1.j_2.\ldots.j_d$ $(d \geq 2)$. For $i = 1, 2, \ldots, d$, U_i and p_i respectively denote region $j_1.j_2.\ldots.j_i$ and the pivot selected to decompose U_i. That is, p_i is an endpoint of the lid $e(U_i)$ of U_i. For convenience, assume that $U_0 = P$ and $p_0 = u$. In what follows, we construct a shooing schedule of $s+1$ searchers for U_i with respect to $e(U_i)$ using a recursive argument.

First, we construct a shooing schedule for U_d, which constitutes the basis of the recursive construction. Place a searcher at p_d and execute OWSS with pivot p_d. Since each vertex of the form $j_1.j_2.\ldots.j_d.h$ is a leaf of $T(P)$, region $j_1.j_2.\ldots.j_d.h$ has at most $3^s - 1$ reflex vertices, and therefore it is searchable by s searchers by the induction hypothesis. Hence OWSS successfully finishes and solves the shooing problem for U_d with respect to $e(U_d)$ by $s+1$ searchers.

Let $\mathcal{S}_i$ be the shooing schedule of $s+1$ searchers for U_i with respect to $e(U_i)$. We now construct a shooing schedule of $s+1$ searchers for U_{i-1} with respect to $e(U_{i-1})$. See Figure 9 for illustration. As shown in Figure 9, we define regions X and Y where $U_{i-1} = U_i \cup X \cup Y$. Depending on the number of reflex vertices $r(Y)$ of Y, we select one of the following two shooing schedules.

Suppose $r(Y) \leq 3^s - 1$.

1. Execute $\mathcal{S}_i$ to solve the shooing problem for U_i with respect to $e(U_i)$ by $s+1$ searchers.
2. Place searcher S_{s+1} at p_{i-1} and aim its ray at p_i. (We assume that S_{s+1} is not the searcher that is at p_i at the end of $\mathcal{S}_i$).
3. Clear Y using searchers $S_1, S_2, \ldots, S_s$.
4. Execute OWSS using the ray of S_{s+1} to clear X (until the ray of S_{s+1} is aimed at the lid of U_{i-1}).

Clearly, this is a shooing schedule of $s+1$ searchers for U_{i-1} with respect to $e(U_{i-1})$.

Next, suppose $r(Y) \geq 3^s$. If we let $Z = P - (U_i \cup Y)$, then Z has at most $3^s - 1$ reflex vertices. We can then clear P (rather than shooing U_{i-1}) as follows.

1. Execute $\mathcal{S}_i$ to solve the shooing problem for U_i with respect to $e(U_i)$ by $s+1$ searchers.
2. Place searcher S_{s+1} at p_{i-1} and aim its ray at p_i. (We assume that S_{s+1} is not the searcher that is at p_i at the end of $\mathcal{S}_i$.)
3. Clear Z using searchers $S_1, S_2, \ldots, S_s$.
4. Execute OWSS using the ray of S_{s+1} to clear Y.

Again clearly, this is a search schedule of $s+1$ searchers for P. (End of Case 1)

(Case 2) Suppose that $T^*(P)$ contains two leaves ℓ and ℓ'. Let $\mathbf{x}$ and p be the vertex that has two children in $T^*(P)$ and the pivot selected for decomposing $\mathbf{x}$. By U and W we

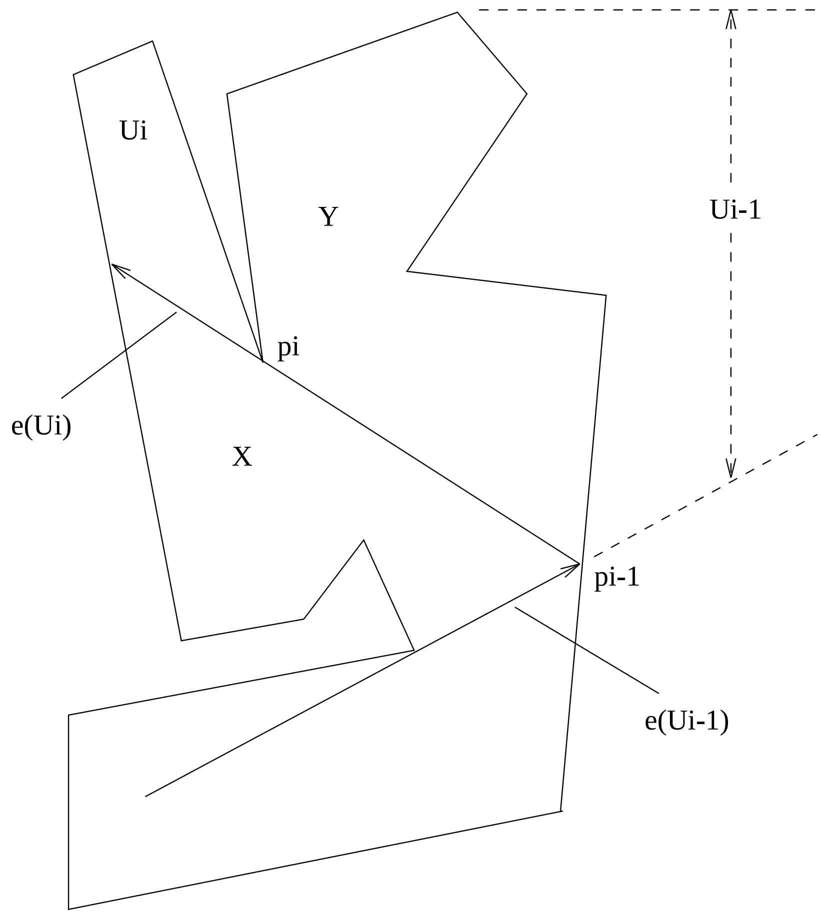


Figure 9: The case in which exactly one subregion has at least 3^s reflex vertices.

denote the two regions corresponding to the two children of $\mathbf{x}$. See Figure 10 for illustration. As shown in Figure 10, we define regions X and Y, and let $Z = P - (U \cup W \cup X \cup Y)$. Since both U and W contain at least 3^s reflex vertices, each of X, Y and Z has at most $3^s - 1$ reflex vertices. As we showed in Case 1, we can construct a shooing schedule of $s+1$ searchers for each of U and W with respect to their lids $e(U)$ and $e(W)$. Let us call these schedules $\mathcal{S}_U$ and $\mathcal{S}_W$. The following is a search schedule of $s+1$ searchers for P.

1. Execute $\mathcal{S}_U$ to solve the shooing problem for U with respect to $e(U)$ by $s+1$ searchers.

2. Place searcher S_{s+1} at p and aim its ray at $e(U)$. (We assume that S_{s+1} is not the searcher that is at p at the end of $\mathcal{S}_U$.)

3. Clear Z using searchers $S_1, S_2, \ldots, S_s$.

4. Execute OWSS using the ray of S_{s+1} to clear X.

5. Clear Y using searchers $S_1, S_2, \ldots, S_s$.

6. Place searcher S_1 at an endpoint of $e(W)$ and aim its ray at $e(W)$.

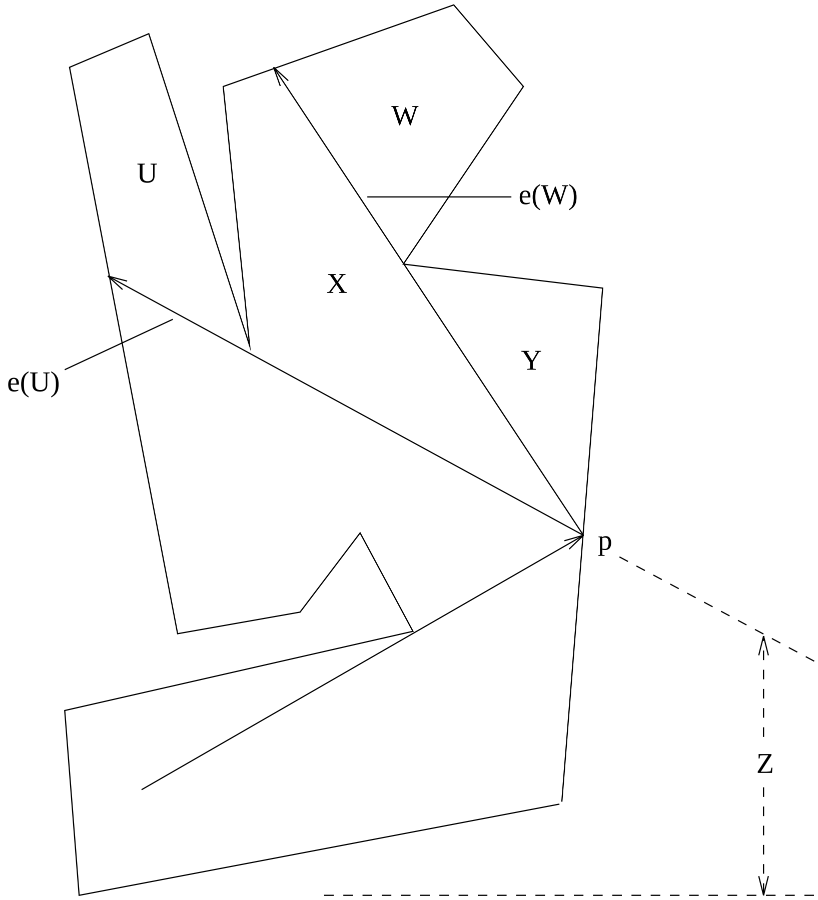


Figure 10: The case in which two subregions, U and W, have at least 3^s reflex vertices.

7. Execute the reverse schedule $\mathcal{S}_W^R$ of $\mathcal{S}_W$ to solve the chasing problem for W with respect to $e(W)$ by $s+1$ searchers.

Again, this is a search schedule of $s+1$ searchers for P. (End of Case 2) $\square$

As a final note of this subsection, the upper bound of $1 + \lfloor \log_3 r \rfloor$ on $ps(P)$ in terms of r is at least as strong as that $(1 + \lfloor \log_3(n-3) \rfloor)$ in terms of n we derived in the last subsection, since $r \leq n-3$ holds in general. On the other hand, the upper bounds in terms of r and b are incomparable.

4.3 The Size of a Minimum Guard Set

The objective of this subsection is to show $ps(P) \leq 2 + \lfloor \log_2 g \rfloor$, where g is the size of a minimum guard set G of P. The proof is by induction similar to that used in the proof of Theorem 7. However, this time, we need to take into account the effect of guards that are located outside of the region under consideration; for some region $R \subset P$, $R \cap G$ may not be a guard set of R, although G is a guard set of P.

We start with several results that characterize classes of polygons searchable by a small number of searchers.

Theorem 8 [17] *If P is weakly visible from an edge e, i.e., every point in P is visible from some point in e, then P is searchable by one searcher.* □

Lemma 8 *Suppose that P is visible from a point $a \in P$. Then the shooing (and therefore chasing) problem for P with respect to any edge e is solvable by two searchers.*

Proof Let u be an endpoint of e. Region $P - V(u)$ consists of a number of pockets $R_1, \ldots, R_k$, where each R_i is weakly visible from its lid $e(R_i)$ since it is visible from a and a ($\in V(u)$) is not in the interior of R_i. So each R_i is searchable by one searcher by Theorem 8, and thus P can be cleared using two searchers by OWSS with pivot u. As we noted earlier, the resulting schedule can be a chasing or a shooing schedules for P with respect to e depending on the direction in which it is executed. □

Lemma 9 *Suppose that P is weakly visible from a point a and an edge e, i.e., every point in P is visible either from a or from some point in e. Then the shooing (and therefore chasing) problem for P with respect to e is solvable by two searchers.*

Proof Let $e = (u, v)$. There are two cases.

(Case 1) Suppose that $a \in V(u) \cup V(v)$. Without loss of generality assume that $a \in V(u)$. Since each connected region in $P - V(u)$ is weakly visible from its lid, it is searchable by one searcher by Theorem 8. Hence OWSS executed with pivot u is a shooing schedule of two searchers for P with respect to e. (End of Case 1)

(Case 2) Suppose that $a \notin V(u) \cup V(v)$. Let U be the pocket with respect to u that contains a. Define regions X and Y as shown in Figure 11. Region U is weakly visible from a and its lid $e(U) = (x, y)$. Assume that there is a shooing schedule $\mathcal{S}_U$ of two searchers for U with respect to the lid (x, y) of U. Then the following is a shooing schedule of two searchers for P with respect to e.

1. Execute $\mathcal{S}_U$ to solve the shooing problem for U with respect to (x, y) by two searchers.

2. Let S_1 be the searcher that is at an endpoint of (x, y) when $\mathcal{S}_U$ terminates. Place S_2 at u and aim its ray at y. (Now S_1 can be relocated without contaminating U.)

3. Clear X using S_1.

4. Clear Y by executing OWSS with pivot u using the ray of S_2.

Then the shooing problem for P is reduced to a smaller shooing problem for U, and the recursion terminates when Case 1 is encountered, in which case the existence of a shooing schedule is guaranteed. Hence the above gives a shooing schedule of two searchers for P with respect to e. (End of Case 2) □

Lemma 10 *If P is weakly visible from two edges e and e', then the corridor search problem for P with respect to e and e' is solvable by two searchers.*

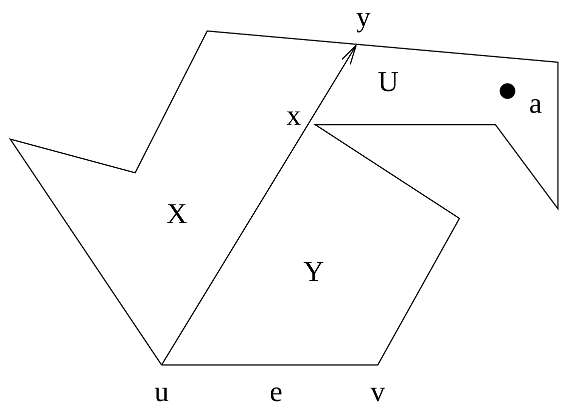


Figure 11: Illustration for the proof of Lemma 9.

Proof We construct a corridor search schedule of two searchers for P with respect to e and e' using the recursive argument used in the proof of Lemma 9. Let $e = (u, v)$ and $e' = (u', v')$. We consider the following two cases, where u, v, v', u' appear on ∂P in this order.

(Case 1) Suppose that e and e' are adjacent. Without loss of generality assume that $u = u'$. Since every pocket with respect to u is weakly visible from its lid, the corridor search problem for P with respect to e and e' is solvable by two searchers, simply by executing OWSS with pivot u $(= u')$. (End of Case 1)

(Case 2) Suppose that e and e' are not adjacent. Then at least one of the following conditions holds:

1. There is a vertex y between u and u' that is visible from v.

2. There is a vertex x between v and v' that is visible from u.

Figure 12 illustrates the case in which the second condition holds. Then the following schedule solves the corridor search problem for P with respect to e and e' by two searchers, using the corridor search schedule given in Case 1 as the base case of recursion.

1. Execute OWSS with pivot u until the ray is aimed at $\overline{ux}$, to clear Y.

2. Execute a corridor search schedule of two searchers for X with respect to $\overline{ux}$ and e'.

(End of Case 2) □

Lemma 11 *Let P be a simple polygon, and $G \subseteq P$ a nonempty set of points in P of size $|G| = g \leq 2^s - 1$ for some natural number $s \geq 1$. If P is weakly visible from G and an edge e, then P is searchable by $s + 1$ searchers.*

Proof Without loss of generality, we may assume that $g = 2^s - 1$ for some $s \geq 1$. The proof is by induction on s.

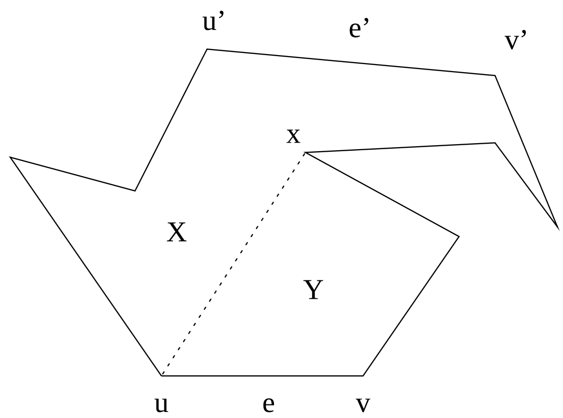


Figure 12: Illustration for the proof of Lemma 10.

(Basis) For the case $s = 1$, the shooing problem for P with respect to e is solvable by two searchers by Lemma 9, since P is weakly visible from one point and e.

(Induction) Assume that the claim holds for some $s \geq 1$, and consider the case $g = 2^{s+1} - 1$. Let $e = (u, v)$. We use an argument similar to that in the proof of Theorem 7. We construct a decomposition tree $T(P)$ starting with u, in such a way that we stop decomposing a region R if R is known to be searchable by $s + 1$ searchers, i.e., if $|R \cap G| \leq 2^s - 1$ holds. Assume that, during the construction of $T(P)$, a node had two child nodes representing pockets R_1 and R_2 such that $|R_1| \geq 2^s$ and $|R_2| \geq 2^s$. Then we would have $|G| \geq 2^{s+1}$, a contradiction. Therefore, $T^*(P)$ is a chain $\ell = j_1.j_2.\ldots.j_d$. (We use the symbols defined in the proof of Theorem 7.) Reasoning as in Case 1 of the proof of Theorem 7, we can construct a shooing schedule of $s+2$ searchers for P with respect to e as follows: If $i = d$, then OWSS terminates successfully and solves the shooing problem for U_d with respect to $e(U_d)$ by $s+2$ searchers, since each region $j_1.j_2.\ldots.j_d.h$ is searchable by $s+1$ searchers by the induction hypothesis. Now, let $\mathcal{S}_i$ be the shooing schedule of $s+2$ searchers for U_i with respect to $e(U_i)$. We now construct a shooing schedule of $s+2$ searchers for U_{i-1} with respect to $e(U_{i-1})$. Define X and Y as shown in Figure 9, where $U_{i-1} = U_i \cup X \cup Y$. Since $|Y \cap G| < 2^s$, the following is a shooing schedule of $s+2$ searchers for U_{i-1} with respect to $e(U_{i-1})$.

1. Execute $\mathcal{S}_i$ to solve the shooing problem for U_i with respect to $e(U_i)$ by $s+2$ searchers.
2. Place searcher S_{s+2} at p_{i-1} and aim its ray at p_i. (We assume that S_{s+2} is not the searcher that is at p_i at the end of $\mathcal{S}_i$).
3. Clear Y using searchers $S_1, S_2, \ldots, S_{s+1}$.
4. Execute OWSS with pivot p_{i-1} using the ray of S_{s+2} (until the ray is aimed at the lid of U_{i-1}), to clear X.

□

Theorem 9 *Let P be a simple polygon having a guard set of size g. Then $ps(P) \leq 2 + \lfloor \log_2 g \rfloor$.*

Proof The theorem follows immediately from Lemma 11. □

5 Lower Bounds on Polygon Search Numbers

In this section we present a series of simple polygons $P(i)$, $i = 1, 2, \ldots$, such that $ps(P(i)) = i + 1$, that provide lower bounds on the polygon search number in terms of the four parameters discussed in the previous section—the number of edges, the number of reflex vertices, the bushiness, and the size of a minimum guard set. Let us start with the following simple theorem by Suzuki and Yamashita[17].

For points x, y and $z \in P$, if the Euclidean shortest path between y and z within P contains at least one point of $V(x)$, then y and z are said to be *separable* by x. Points x, y and $z \in P$ are said to be *mutually nonseparable* if no two points out of the three are separable by the third.

Theorem 10 [17] *Let P be a simple polygon. If P is searchable by a single searcher, then no three points in P are mutually nonseparable.* □

$P(1)$ is given in Figure 13(a). Notice that three vertices a, b and c are mutually nonseparable, and hence $ps(P(1)) \geq 2$. $P(2)$ is shown in Figure 13(b), and is constructed by "pasting" three copies of $P(1)$ (of which one is drawn backwards) to the three "hooks" of a fourth copy of $P(1)$ (drawn at the center). Analogously, $P(i+1)$ is recursively constructed from three copies of $P(i)$ and a copy of $P(1)$: the three copies of $P(i)$ (of which one is drawn backwards) are pasted to a copy of $P(1)$ (drawn at the center) at its three hooks. Figure 13(c) shows $P(3)$ that consists of three copies of $P(2)$ and a copy of $P(1)$. The following lemma gives the values of the four parameters of $P(i)$ that are of interest to us.

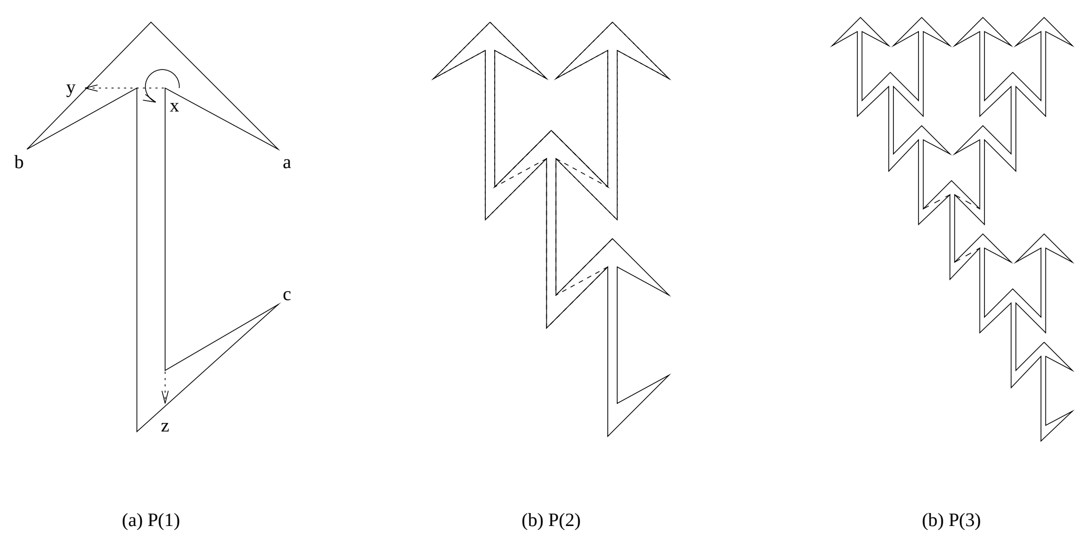


Figure 13: Simple polygons $P(1)$, $P(2)$ and $P(3)$.

Lemma 12 *Let n_i, r_i, b_i and g_i be the number of edges, the number of reflex vertices, the bushiness, and the size of a minimum guard set of $P(i)$, respectively. Then:*

1. $n_i = 3^{i+1} - 1$.

2. $r_i = 3(3^i - 1)/2$.

3. $b_i = (3^i - 1)/2$.

4. $g_i = (3^i + 1)/2$.

Proof As for n_i and r_i, observe that the following equations hold:

$$n_1 = 8 \text{ and } n_{i+1} = 3n_i + 2,$$

and

$$r_1 = 3 \text{ and } r_{i+1} = 3(r_i + 1).$$

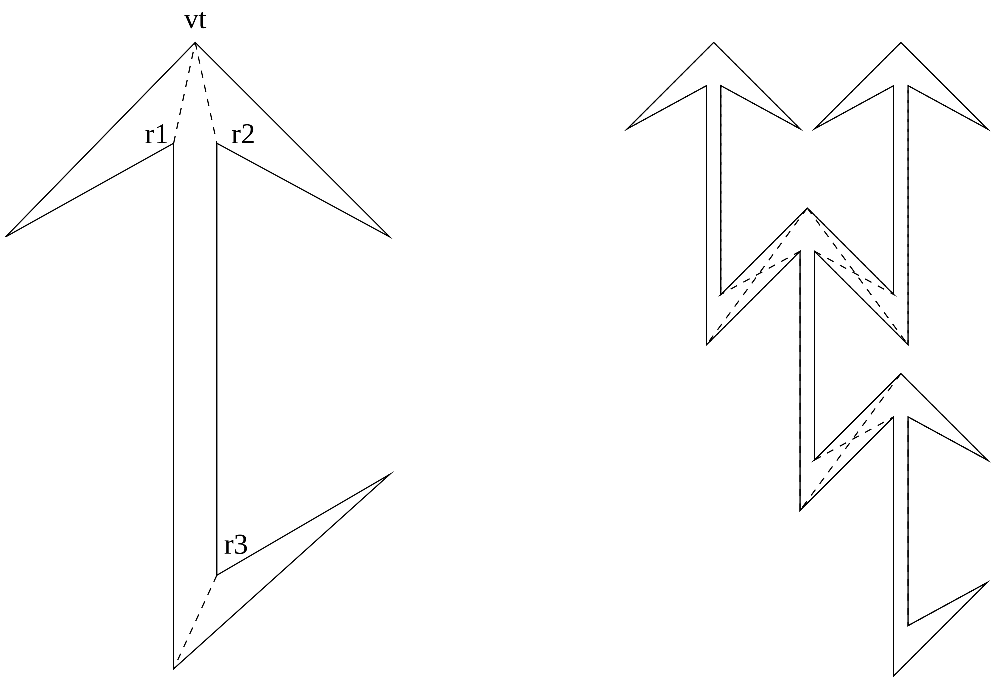
Then one can easily obtain n_i and r_i by solving the equations.

We next consider the bushiness b_i of $P(i)$. Clearly any triangulation of $P(1)$ requires three diagonals, $\overline{v_t r_1}$, $\overline{v_t r_2}$ and $\overline{v_b r_3}$, shown as dashed lines in Figure 14(a). They create three triangles, and it is easy to see that the rest of $P(1)$, which is a convex polygon $v_t r_2 r_3 v_b r_1$, cannot be triangulated without creating a triangle that shares no edges with $P(1)$. On the other hand, introducing one such triangle is sufficient to triangulate $P(1)$. So the bushiness of $P(1)$ is 1. As for $P(2)$, inside each of the "corridors" where adjacent copies of $P(1)$ are pasted, any triangulation must use one of the two diagonals shown in dashed lines in Figure 14(b), one joining the bottom vertex of $P(1)$ with the top vertex of another copy of $P(1)$, and the other joining the reflex vertex near the bottom of $P(1)$ with a reflex vertex near the top of another copy of $P(1)$. In either case, the argument for $P(1)$ given above applies to each copy of $P(1)$ in $P(2)$, and hence any triangulation of $P(2)$ requires four triangles that share no edges with $P(2)$. On the other hand, introducing four such triangles is sufficient to triangulate $P(2)$. So the bushiness of $P(2)$ is 4. The proof for general $P(i)$ is by induction and a similar argument.

We now estimate g_i. First observe that $g_1 = 2$. Figures 15(a) and (b) illustrate examples of guard sets for $P(1)$ and $P(2)$, where the circles represent the positions of the guards.

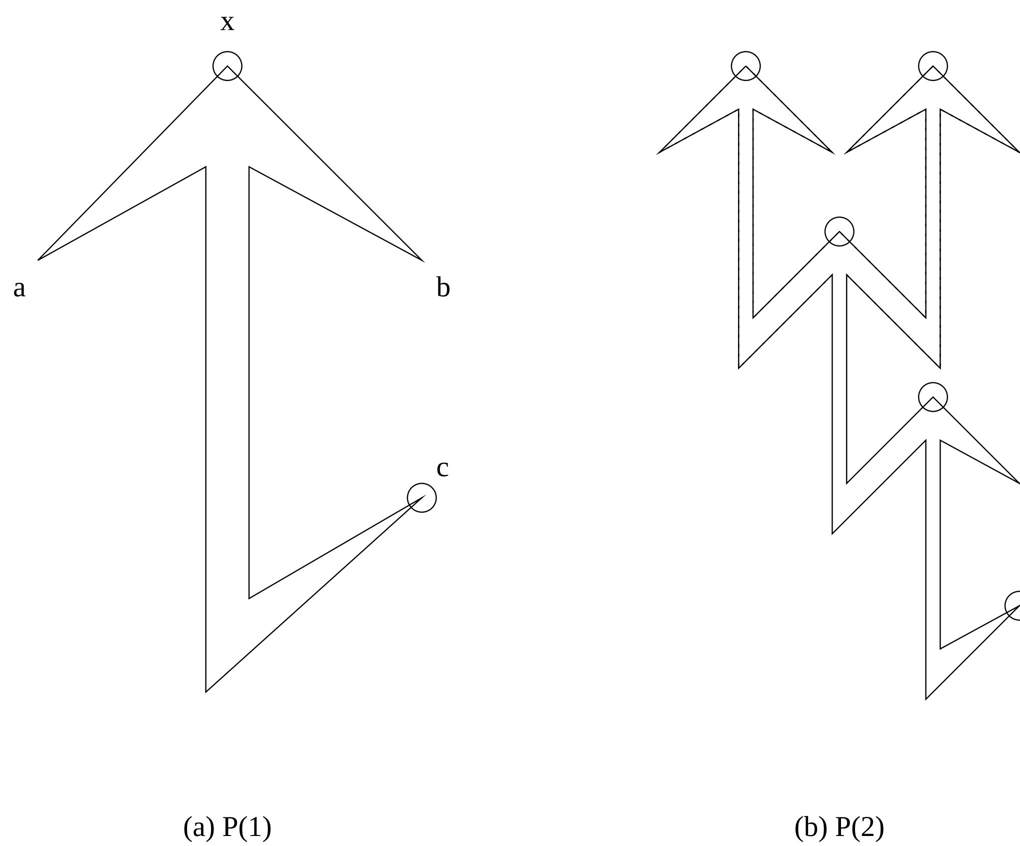
We next show $g_{i+1} = 3g_i - 1$ for any $i \geq 1$. In $P(1)$, we call vertices x and c the *head* and *tail*, respectively. By the definition of $P(i)$, $P(i)$ contains many copies of $P(1)$, each having a copy of head x. We call these copies of x in $P(i)$ the *heads* of $P(i)$. On the other hand, all but one copy of tail c have disappeared during the pasting process. We call the remaining unique copy of c, which belongs to the lowest copy of $P(1)$, the *tail* of $P(i)$. As we observe from Figures 15(a) and (b), the set of consisting of the heads and the tail of $P(i)$ form a guard set for $P(i)$ for any $i \geq 1$. Therefore $g_{i+1} \leq 3g_i - 1$.

Finally, we assume $g_{i+1} < 3g_i - 1$ for some i and derive a contradiction. Let G be a guard set for $P(i+1)$ such that $|G| = g_{i+1}$. Let P_1, P_2, P_3 and P_4 be the three copies of $P(i)$ and a copy of $P(1)$ in $P(i+1)$, respectively. By $g(j)$ we denote the number of guards in P_j for $i = 1, 2, 3, 4$. Let $G_j = G \cap P_j$ be the set of guards in P_j for $j = 1, 2, 3$. Then by the definition of $P(i+1)$, $G_j \cup \{x\}$ is a guard set for P_j, where x is the tail of P_j. Since $|G_j \cup \{x\}| \geq g_i$, we have $g(j) \geq g_i - 1$ for $j = 1, 2, 3$. Hence $g(4) \leq 1$. Suppose that $g(4) = 0$. Since the interior of the other two copies of $P(i)$ is not visible from any guard located in a copy of $P(i)$, $g(j) \geq g_i$ holds for $j = 1, 2, 3$, which is a contradiction. Suppose that $g(4) = 1$. Let a be the guard in the copy of $P(1)$. By the definition of $P(i+1)$, the interior of at most



(a) P(1) (b) P(2)

Figure 14: Triangulating $P(1)$ and $P(2)$.



(a) P(1)

(b) P(2)

Figure 15: Examples of guard sets for $P(1)$ and $P(2)$.

two copies of $P(i)$ is visible from a. Without loss of generality we can assume that P_1 and P_2 are the copies of $P(i)$ whose interior is visible from a. Then $g(1) = g(2) \geq g_i - 1$ and $g(3) \geq g_i$ hold, which is a contradiction since $g_{i+1} \geq 3g_i - 1$.

We have proved $g_1 = 2$ and $g_{i+1} = 3g_i - 1$, and the claim of the lemma immediately follows. □

The next theorem states that $i+1$ searchers are necessary and sufficient to search $P(i)$.

Theorem 11 *For any* $i \geq 1$, $ps(P(i)) = i + 1$.

Proof The proof is by induction on i.
(Basis) For the case $i = 1$, since $ps(P(1)) \geq 2$ as we mentioned above, it suffices to give a search schedule of two searchers for $P(1)$. The following is such a schedule that uses two searchers S and S'. See Figure 13(a).

1. Place S at x and aim its ray F_S at a.

2. Turn F_S counter-clockwise until it is aimed at y. Now the region of $P(1)$ that F_S has scanned is clear.

3. While aiming F_S at y, place S' at y and aim its ray $F_{S'}$ at b.

4. Turn $F_{S'}$ counter-clockwise until it is aimed at x, to clear the "hook b". It is worth noting that we can now remove S' without recontaminating the subregion that has been cleared.

5. Turn F_S counter-clockwise until it is aimed at z. Then all except "hook c" has been cleared.

6. While aiming F_S at z, place S' at z and aim $F_{S'}$ at c.

7. Turn $F_{S'}$ counter-clockwise until it is aimed at x.

Thus $ps(P(1)) = 2$. This schedule is an easy application of OWSS: S executes OWSS in counter-clockwise from pivot x.
(Induction) We assume that $ps(P(i)) = i+1$ for some $i \geq 1$ and prove $ps(P(i+1)) = i+2$. First, we show $ps(P(i+1)) \geq i+2$. To derive a contradiction, assume $ps(P(i+1)) = i+1$ and let $\mathcal{S} = \langle \sigma_1, \sigma_2, \ldots \sigma_{i+1} \rangle$ be a search schedule of $i+1$ searchers for $P(i+1)$ defined over interval $[0, T]$ of real time. Let P_1, P_2 and P_3 be the three copies of $P(i)$ in $P(i+1)$. For each $j = 1, 2, 3$, let T_j be the set of time instants at which every searcher illuminates some point in the interior of P_j (i.e., $P_j - \partial(P_j)$). Note that $T_j \neq \emptyset$ for each $j = 1, 2, 3$, since $ps(P(i)) = i+1$ implies that $P(i)$ cannot be cleared unless at some time instants all $i+1$ searchers simultaneously illuminate some points in the interior of P_j. Also, T_1, T_2 and T_3 are disjoint subsets of $[0, T]$ since by the topology of $P(1)$, no searcher can illuminate the interiors of two copies of $P(i)$ simultaneously. Furthermore, any three points x_1, x_2 and x_3 such that $x_j \in P_j - \partial(P_j)$, $j = 1, 2, 3$, are mutually nonseparable, and hence at any time $t \in T_j$, the two copies of $P(i)$ other than P_j are either both clear or both contaminated. Now, let $I_1, I_2, \ldots, I_m$ be the time intervals comprising $T_1 \cup T_2 \cup T_3$ in increasing order of time. Then there exist some ℓ, j and h ($1 \leq \ell \leq m-1$, $j, h \in \{1, 2, 3\}$) such that at time $t_\ell \in I_\ell \subseteq T_j$ the two copies of $P(i)$ other than P_j are both contaminated, and that at time $t_{\ell+1} \in I_{\ell+1} \subseteq T_h$ the two copies of $P(i)$ other than P_h are both clear. This is a contradiction, since P_k ($\neq P_j, P_h$) cannot be cleared between I_ℓ and $I_{\ell+1}$. To complete the induction step we present a search schedule of $i+2$ searchers for $P(i+1)$. A recursive description of the schedule is the following. Let $S_1, S_2, \ldots, S_{i+2}$ be the searchers. See Figure 16(a) for illustration.

1. Place S_{i+2} at x and aim its ray at y.

2. Clear the triangle cut off by segment $\overline{xy}$ from the copy of $P(1)$ using searcher S_1.

3. Clear P_1 using searchers $S_1, S_2, \ldots, S_{i+1}$.

4. Clear P_2 using searchers $S_1, S_2, \ldots, S_{i+1}$.

5. Place S_1 at x' and aim its ray at y'.

6. Move S_1 to u along the boundary in such a way that its ray is always aimed at a point between y' and v and that it is aimed at v when S_1 reaches u.

7. Clear P_3 using searchers $S_2, S_3, \ldots S_{i+2}$.

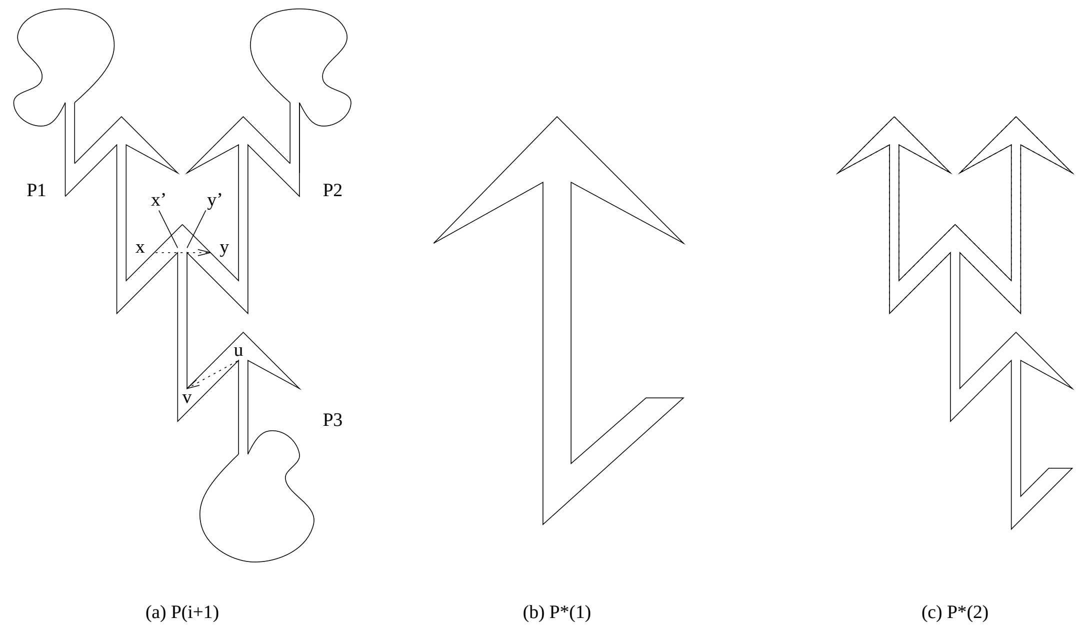


Figure 16: A recursive search schedule for $P(i+1)$, and simple polygons $P^*(1)$ and $P^*(2)$.

The only detail that is missing in the above schedule is that in steps 3 and 4, the subpolygon to be cleared by $S_1, S_2, \ldots, S_{i+1}$ is not exactly a copy of $P(i)$ with a "pointy tail" in the lowest copy of $P(1)$, but is a slightly different polygon which we call $P^*(i)$ with a "fat tail" as shown in Figures 16(b) and (c). However, it is easy to see that $P^*(1)$ is searchable by two searchers using the schedule for $P(1)$ presented earlier, and that we can construct a search schedule of $i+1$ searchers for $P^*(i)$ by a recursive strategy essentially similar to the one given above. Using that search schedule in steps 3 and 4, the induction step for constructing a search schedule of $i+2$ searchers for $P(i+1)$ is now completed. □

Theorem 12 *For any natural number $s \geq 2$, there is a simple polygon P satisfying $ps(P) = \log_3(n+1) = \log_3(2r+3) = \log_3(2b+1)+1 = \log_3(2g-1)+1 = s$, where n, r, b and g are the numbers of edges, the number of reflex vertices, the bushiness, and the size of a minimum guard set of P, respectively.*

Proof The theorem follows immediately from Theorem 11 and Lemma 12. □

It is worth noting that the above theorem does not guarantee the existence of a polygon P satisfying the condition for each value of n, r, b or g. The following theorem narrows the gap for the case of parameter b.

Theorem 13 *For any $b \geq 1$, there exists a simple polygon P with bushiness b such that $ps(P) = 1 + \lfloor \log_3 (2b+1) \rfloor$.*

Proof If $b = (3^i - 1)/2$, $i \geq 1$, then the bushiness of $P(i)$ is b (Lemma 12) and $ps(P(i)) = i+1 = 1 + \lfloor \log_3(2b+1) \rfloor$ (Theorem 11). So the claim of the theorem holds for such b. Now, imagine we "grow" $P(i)$ into $P(i+1)$ by adding one copy of $P(1)$ at a time and obtain a series of polygons $Q_{i,0} = P(i)$, $Q_{i,1}$, $Q_{i,2}$, ..., $Q_{i,3^i-1}$, and $Q_{i,3^i} = P(i+1)$, where for each ℓ, $1 \leq \ell \leq 3^i - 1$, $Q_{i,\ell}$ has exactly $(3^i-1)/2 + \ell$ copies of $P(1)$ and hence, by an argument similar to the one used in the proof of Lemma 12, $Q_{i,\ell}$ has bushiness $b_{i,\ell} = (3^i-1)/2 + \ell$. Then for $1 \leq \ell \leq 3^i - 1$, by Theorem 6 $ps(Q_{i,\ell}) \leq 1 + \lfloor \log_3(2b_{i,\ell}+1) \rfloor = i+1$. On the other hand, since each $Q_{i,\ell}$, $1 \leq \ell \leq 3^i - 1$, contains $P(i)$ as a subpolygon that can be cut off by a single chord, by Lemma 5 $ps(Q_{i,\ell}) \geq ps(P(i)) = i+1$. This completes the proof of the theorem. □

6 Concluding Remarks

The problem of searching for mobile intruders in a polygonal region by mobile searchers has been investigated. We have presented upper bounds, and lower bounds in the worst case, on the number $ps(P)$ of searchers necessary and sufficient to search any polygon P in terms of four measures of shape complexity of P—the number n of edges, the number r of reflex vertices, the bushiness b, and the size g of a minimum guard set.

As for n, we showed (1) $ps(P) \leq 1 + \lfloor \log_3(n-3) \rfloor$, and (2) there is a simple polygon P having n edges such that $ps(P) = \log_3(n+1) = s$ for each natural number s. Since $1 + \lfloor \log_3(n-3) \rfloor \leq \lceil \log_3 n \rceil$ and it is easy to observe that for any $n \geq 3$, there is a simple polygon P of n edges such that $\lfloor \log_3(n+1) \rfloor \leq ps(P)$,[2] the gap between the upper and lower bounds on $ps(P)$ is at most one for any n (and they agree for some values of n).

As for r, we showed (1) $ps(P) \leq 1 + \lfloor \log_3 r \rfloor$, and (2) there is a simple polygon P having r reflex vertices such that $ps(P) = \log_3(2r+3) = s$ for each natural number s. Since again we can observe that for any $r \geq 0$, there is a simple polygon P of r reflex vertices such that $\lfloor \log_3(2r+3) \rfloor \leq ps(P)$,[3] the gap between the upper and lower bounds on $ps(P)$ is at most one for any r (and they agree for some values of r). Since $r \leq n-3$ in general, the upper bound in r is stronger than that in n; there exist simple polygons having large n and small r.

As for b, we showed matching upper and lower bounds of $1 + \lfloor \log_3(2b+1) \rfloor$ on $ps(P)$.[4]

Finally as for g, we showed (1) $ps(P) \leq 2 + \lceil \log_2 g \rceil$, and (2) there is a simple polygon P with a minimum guard set of size g such that $ps(P) = 1 + \log_3(2g-1) = s$ for each natural number s. Unfortunately, we could not bound the gap between the upper and lower bounds by a constant, although we conjecture the following.

Conjecture 1 *For any simple polygon P with a minimum guard set of size g, $ps(P) \leq 2 + \lceil \log_3 g \rceil$.* □

In this paper, we examined four natural measures of shape complexity, but it turned out that neither of them can capture the difficulty of searching, since the search number of an arbitrary given polygon cannot be bounded from below by any one of the four measures. Discovering such a measure of shape complexity remains as an open problem. Another open problem is to evaluate the difference in search power between 1-searcher and ∞-searcher from the viewpoint of polygon search numbers.

[2] Such a polygon can easily be constructed by adding extra vertices and edges to some $P(i)$.

[3] Such a polygon can easily be constructed by adding extra reflex vertices and edges to some $P(i)$.

[4] Recently the authors proved that the same upper and lower bounds hold for the case of ∞-searchers.

References

[1] S. Carlsson, H. Jonsson, and B. J. Nilsson: "Finding the shortest watchman route in a simple polygon," *Proc. 4th International Symposium on Algorithms and Computation,* Hong Kong, 1993, 58–67.

[2] W. -P. Chin and S. Ntafos, "Optimum watchman routes," *Proc. 2nd ACM Symposium on Computational Geometry,* 1986, 24–33.

[3] W. -P. Chin and S. Ntafos, "Optimum watchman routes," *Information Processing Letters 28,* 1988, 39–44.

[4] D. Crass, I. Suzuki and M. Yamashita, "Searching for a mobile intruder in a corridor – The open edge variant of the polygon search problem," *International J. Computational Geometry and Applications 5,* 4, 1995, 397–412.

[5] M. de Berg and R. Klein, private communication, 1993.

[6] P.J. Heffernan, "An optimal algorithm for the two–guard problem," *Proc. 9th ACM Symposium on Computational Geometry,* 1993, 348–358.

[7] C. Icking and R. Klein, "The two guards problem," *Proc. 7th ACM Symposium on Computational Geometry,* 1991, 166–175.

[8] J. Kahn, M. Klawe and D. Kleitman, "Traditional galleries require fewer watchmen," *SIAM J. Algebraic and Discrete Methods 4*, 2, 1983, 194–206.

[9] A. S. LaPaugh, "Recontamination does not help to search a graph," *J. ACM 40,* 2, 1993, 224–245.

[10] D. T. Lee and A. K. Lin, "Computational complexity of art gallery problems," *IEEE Trans. Information Theory IT–32,* 2, 1986, 276–282.

[11] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson and C. H. Papadimitriou, "The complexity of searching a graph," *J. ACM, 35,* 1, 1988, 18–44.

[12] J. O'Rourke, "An alternate proof of the rectilinear art gallery theorem," *J. Geometry 21,* 1983, 118–130.

[13] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press, New York, Oxford, 1987.

[14] T. D. Parsons, "Pursuit–evasion in a graph," in *Theory and Applications of Graphs*, Y. Alavi and D. Lick (eds.), Lecture Notes in Mathematics 642, Springer-Verlag, Berlin, 1976, 426–441.

[15] T. Shermer, "Computing bushy and thin triangulations," in *Snapshots of Computational and Discrete Geometry*, G. Toussaint (ed.), Technical Report SOCS-88.11, School of Computer Science, McGill University, Montreal, Canada, June 1988, 119–133.

[16] K. Sugihara, I. Suzuki and M. Yamashita, "The searchlight scheduling problem," *SIAM J. Computing 19,* 6, 1990, 1024–1040.

[17] I. Suzuki and M. Yamashita , "Searching for a mobile intruder in a polygonal region," *SIAM J. Computing 21,* 5, 1992, 868–888.

[18] X. Tan and T. Hirata: "Constructing shortest watchman routes by divide-and-conquer," *Proc. 4th International Symposium on Algorithms and Computation,* Hong Kong, 1993, 68–77.