アルゴリズム 68－6
（1999. 5. 10）

Searching a Simple Polygon by a k-Searcher

Xuehou TAN
Tokai University, 317 Nishino, Numazu 410-0318, Japan

Abstract

The *polygon search problem* is the problem of searching for a mobile intruder in a simple polygon by the mobile searcher having flashlights whose visibility is limited to the rays emanating from his position. The intruder can move arbitrarily faster than the searcher. A searcher is called the *k-searcher* if he can see along k rays emanating from his position, and the ∞*-searcher* if he has a 360^0 field of vision. We present the necessary and sufficient conditions for a polygon to be searchable by a 1-searcher and by a 2-searcher, and give $O(n^2)$ time algorithms for testing the 1-searchability and 2-searchability of simple polygons and $O(n^3)$ time algorithms for generating a search schedule if one exists. We also show that any polygon that is searchable by an ∞-searcher is searchable by a 2-searcher, which confirms a conjecture due to Suzuki and Yamashita.

K-探索者による単純多角形の探索問題について

東海大学 開発工学部 譚 学厚

この論文では、k個の方向を見ることができる k-探索者による多角形探索の必要十分条件を与える。それにより、2-探索者と ∞-探索者に能力の差がないことが示されている。

1 Introduction

In recent years, much attention has been devoted to the problem of searching for a mobile intruder in a polygonal region P by the mobile searcher having flashlights whose visibility is limited to the rays emanating from his position [1, 4, 5]. The goal is to decide whether there exists a *search schedule* for the searcher to detect the intruder, no matter how fast he moves, and generate a search schedule if one exists. This problem, called the *polygon searching problem*, was introduced by Suzuki and Yamashita [4]. Both the searcher and the intruder are modeled by points that can move continuously in P. A searcher is called the k-searcher if he has k flashlights and can see along the rays emanating from his searchlights, or the ∞-searcher if he has a light bulb. A polygon is said *searchable* by a given searcher if a search schedule exists. A number of necessary conditions and sufficient conditions for a polygon to be searchable by a k-searcher and an ∞-searcher are given in [4]. But, the necessary and sufficient conditions remain unknown. It is conjectured in [4] that any polygon that is searchable by an ∞-searcher is searchable by a 2-searcher.

Let us now review some basic definitions given in [4]. Let P denote a simple polygon in the plane. Two points $x, y \in P$ are said to be mutually *visible* if the line segment $\overline{xy}$ connecting them is entirely contained within P. For two regions $R, Q \subseteq P$, we say that R is *weakly visible* from Q if every point in R is visible from some point in Q.

A *search schedule* of the k-searcher for P is a tuple $S = < s, f_1, \cdots, f_k >$ of $k+1$ continuous functions s, f_1, $\cdots$, f_k: $[0,1] \rightarrow P$ such that the intruder is located at at least one of the line segments $\overline{s(1)f_1(1)}, \cdots, \overline{s(1)f_k(1)}$, no matter how he moves. A point $x \in P$ is *illuminated* at time $t \in [0,1]$ during the execution of S if x lies on one of the line segments $\overline{s(t)f_1(t)}, \cdots, \overline{s(t)f_k(t)}$, where $s(t)$ is the position of the searcher and $f_1(t), \cdots, f_k(t)$ are the positions of endpoints of flashlights on the boundary of P at time t, respectively. A schedule of the ∞-searcher can be given analogously [4]. Polygon P is said to be k-searchable (or ∞-searchable) if there exists a search schedule of the k-searcher (or ∞-searcher) for P. Any region that might contain the intruder at a time is said to be *contaminated*; otherwise it is said to be *clear*. If polygon P is searchable, then P should be cleared at time $t = 1$. A search schedule is called *trivial* if the cleared regions at a time t, $0 < t < 1$, degenerate into $\overline{s(t)f_1(t)}, \cdots, \overline{s(t)f_k(t)}$.

Observation 1: Polygon P is searchable iff there is a non-trivial search schedule for P.

In this paper, we give the necessary and sufficient conditions for a polygon to be 1-searchable and 2-searchable, which solves an open problem in computational geometry.

2 Two-guard walkability of simple polygons

We first review the two-guard problem [3], which is related to the polygon search problem. Given a simple polygon with two marked vertices u and v, the two-guard problem asks if there is a walk in P such that two guards l and r move along two polygonal chains of P form u to v, one clockwise and the other counterclockwise, in such a way that l and r are always mutually visible. More formally, let L and R denote two polygonal chains of P from u to v, and let $l(t)$ and $r(t)$ denote the moving functions of l and r on L and R, respectively. A *walk* on polygon P is a pair of continuous functions $l : [0,1] \to L$ and $r : [0,1] \to R$, where $l(0) = r(0) = u$, $l(1) = r(1) = v$, and $l(x)$ and $r(x)$ are mutually visible for all x. Any line segment $\overline{l(x)r(x)}$ is called a *walk segment* of the walk. Correspondingly, a *walk on P from $\overline{p_0q_0}$ to $\overline{p_1q_1}$*, where $p_0 < p_1$ and $q_0 < q_1$, has to fulfill the conditions $l(0) = p_0$, $r(0) = q_0$, $l(1) = p_1$ and $r(1) = q_1$.

Assume that both chains L and R are oriented from u to v. Points on L (R) are denoted by p, p', p_1, etc. (q, q', q_1, etc.). For a vertex x of a polygonal chain, *Succ*(x) denotes the vertex of the chain immediately succeeding x, and *Pred*(x) the vertex immediately preceding x. For two points $p, p' \in L$, we say that p *precedes* p' (and p' *succeeds* p) if we encounter p before p' when traversing L from s to t. We write $p < p'$. The chain $L_{<p}$ ($L_{>p}$) is the subchain of L consisting of all points that precede (succeed) p. The definition for R is symmetric.

A vertex of P is *reflex* if its interior angle is greater than 180°. The backward ray shot from a reflex vertex r of chain L or R, denoted by *Backw(r)*, is the first point of P hit by a "bullet" shot at r in the direction from *Succ*(r) to r, and the forward ray shot *Forw(r)* is the first point hit by the bullet shot at r in the direction from *Pred*(r) to r. We define the orientation of the line segment $\overline{rBackw(r)}$ or $\overline{rForw(r)}$ as from r to $Backw(r)$ or $Forw(r)$. A pair of reflex vertices $p \in L$, $q \in R$ is said to form a *backward deadlock* if $q < Backw(p) \in R$ and $p < Backw(q) \in L$ hold or a *forward deadlock* if $q > Forw(p) \in R$ and $p > Forw(q) \in L$ hold. See Fig. 1.

Lemma 1 *[3] A simple polygon P is walkable if and only if the chains L and R are mutually weakly visible and no deadlocks occur.*

Lemma 2 *[2, 3] It takes $\theta(n)$ time to test the two-walkability of a simple polygon, and $O(n \log n + k)$ time to generate a search schedule where k ($\leq n^2$) is the minimal number of search instructions.*

3 Necessary and sufficient conditions for 1-searchable polygons

Let s denote the 1-searcher and f_1 the endpoint of his flashlight. A search instruction of a 1-searcher is one of the following actions (Fig. 2). (i) Both s and f_1 move forward along segments of single edges. (ii) One moves forward but the other moves backward along segments of single edges. (iii) s or f_1 jumps from one point x on the boundary of P to the other point y such that the ray (or line segment) between them is extended or shortened. The first two instructions of a 1-searcher are allowed for two guards, if we regard s and f_1 as two guards l and r, but the last is not. It then follows that any polygon that is walkable by two guards is 1-searchable. (It is also shown in [5] that the capability of a 1-searcher is exactly the same as that of two guards for the two-guard problem.)

We will present the necessary and sufficient conditions for a polygon to be 1-searchable. Any search schedule of a 1-searcher should start at some point (vertex) a of the boundary of P. So we order the points of the boundary of P by a counterclockwise scan of the boundary of P, starting and also ending at a. For a complete ordering, we consider the vertex a as two vertices a_l and a_r such that $a_l \leq p \leq a_r$, for all points p in the boundary of P. Similar definitions can be given as those in the previous section. For a vertex x of P, $Succ(x)$ denotes the vertex succeeding x, and $Pred(x)$ the vertex immediately preceding x. Also for a reflex vertex r, the backward and forward ray shots $Backw(r)$ and $Forw(r)$ are the first points of P hit by the bullets shot at r in the directions from $Succ(r)$ to r and from $Pred(r)$ to r, respectively.

Theorem 1 *Polygon P is not 1-searchable if one of the following conditions is true.*

*(**A1**) There are three points in P such that no point of the shortest path between any pair of points is visible to the third (Figs. 3a-b).*

*(**A2**) For any vertex a, there are two reflex vertices v_1 and v_2 such that (**A2-a**) $v_1 < Backw(v_1) < v_2$ and $v_1 < Forw(v_2) < v_2$ (Figs. 3c-d), or (**A2-b**) $v_1 < Backw(v_1) < v_2 < Backw(v_2)$ (Fig. 3e)or $Forw(v_1) < v_1 < Forw(v_2) < v_2$ (Fig. 3f).*

Proof. The necessity of the condition **A1** is shown in [4].

For three vertices a, v_1 and v_2 satisfying the condition **A2-a**, any search schedule starting at a becomes trivial when the vertex $Succ(v_1)$ or $Pred(v_2)$ is cleared. Since $Succ(v_1)$ or $Pred(v_2)$ has to be cleared once, any search schedule starting at a is trivial.

Consider the condition **A2-b**. Assume w.l.o.g. that the first alternative of condition **A2-b** applies. Since $v_1 < v_2$, any non-trivial search schedule starting at a has to clear first $Succ(v_2)$ and then $Succ(v_1)$ (Fig. 3e). Assume that the region right to $\overline{v_2 Backw(v_2)}$ is cleared at a time t, $0 < t < 1$, in a search schedule starting at a; otherwise, any search schedule starting at a is trivial. There is at least one vertex a' in the region right to $\overline{v_2 Backw(v_2)}$, and there exists a non-trivial search schedule for P starting at a if and only if there exists a non-trivial search schedule for P starting at a'. From the definition of condition **A2**, there is another pair of vertices v_1' and v_2' corresponding to a'. If three vertices a', v_1' and v_2' satisfy the condition **A2-a**, any search schedule starting at a' (and thus a) is trivial. If they satisfy the condition **A2-b**, then another triple of the vertices satisfying the condition **A2** results in. It is possible for several triples of vertices satisfying the condition **A2-b** to have the same vertex (i.e., $Succ(v_1)$) to be cleared last, but this vertex has to be eventually changed; otherwise the condition **A2-b** would be violated. In this way, either the condition **A2-a** is satisfied, or some vertex appears twice in the set of vertices to be cleared last, that is, there is a cycle among the vertices to be cleared last. Thus, any search schedule starting at a is either trivial or sticks at an infinite cycle, which contradicts with the definition that polygon P is cleared at time $t = 1$ in a search schedule. Since a can be any vertex of polygon P, we obtain that polygon P is not 1-searchable if condition **A2** is true. □

Observation 2: For any three points p_1, p_2 and p_3 satisfying the condition **A1**, we can find three reflex vertices r_1, r_2 and r_2 such that each r_i ($i = 1, 2$ and 3) blocks the corresponding point p_i from being visible to any point in the shortest path between other two points. For short presentation, we also say that such three reflex vertices satisfy the condition **A1**.

Theorem 2 *A simple polygon P is 1-searchable if none of the conditions of Theorem 1 applies.*

Proof. Let a be a vertex of P that does not satisfy the condition **A2** of Theorem 1. We assume that the whole polygon P is not visible from a; in this case, polygon P can be simply cleared. A reflex vertex r is *critical* if one of its ray shots produces a convex angle at r in the piece containing a, i.e., either $r < Backw(r)$ or $Forw(r) < r$. Let $Ray(r)$ denote the point hit by the corresponding ray shot of the critical vertex r. For simplicity, we denote by $P(r)$ and $P - P(r)$ the regions left and right to the line segment $\overline{rRay(r)}$, respectively.

Let $r_1, \cdots, r_m$ be the sequence of critical vertices indexed in a counterclockwise scan of the boundary of P, starting at a. Our general method is to either clear each region $P(r_i)$ for $i = 1, \cdots, m$ or clear each region $P - P(r_i)$ for $i = m, \cdots, 1$.

Before giving the way to decide which order is applied, let us consider two exceptions to our method. Suppose that we clear each region $P(r_i)$ for $i = 1, \cdots, m$. The first exception is that the union of two adjacent regions $P(r_i)$ and $P(r_{i+1})$ gives the whole polygon P. In this case, we clear the whole polygon P after $P(r_i)$ is cleared and then output the search schedule.

The second exception is that the union of $P(r_i)$ and $P(r_{i+1})$ is empty. This case has to be avoided in any non-trivial search schedule. We claim that any two adjacent regions intersect either in the counterclockwise order or in the clockwise order. We prove it by contradiction. W.l.o.g., assume that there is a pair of r_k and r_{k+1} such that $P(r_k)$ does not intersect with $P(r_{k+1})$. Then $r_k < Ray(r_{k+1}) < r_{k+1} < Ray(r_k)$; otherwise the condition **A2-b** would be satisfied in the case $Ray(r_k) < r_k < Ray(r_{k+1}) < r_{k+1}$ (see also Fig. 3f). If there is another pair of $r_{k'}$ and $r_{k'+1}$ such that $P(r_{k'})$ does not intersect with $P(r_{k'+1})$, then three vertices a, r_{k+1} and $r_{k'+1}$ satisfy the condition **A2-b**, a contradiction (Fig. 4a). Assume now that there is another pair of $r_{k''}$ and $r_{k''-1}$ such that $P - P(r_{k''})$ does not intersect with $P - P(r_{k''-1})$. W.l.o.g., assume that $r_{k+1} < r_{k''-1}$. If $Ray(r_k) < r_{k''}$ and $Ray(r_{k''}) > r_k$, then three vertices a, r_k and $r_{k''}$ satisfy the condition **A2-a** (Fig. 4b), a contradiction. Otherwise there are three vertices (i.e., $r_{k''-1}$, $r_{k''}$ and r_{k+1} in Fig. 4c) which satisfy the condition **A1**, a contradiction again. The claim is proved. Hence, if any two adjacent regions in the counterclockwise order intersects, we clear all regions $P(r_i)$ for $i = 1, \cdots, m$. Otherwise, any two adjacent regions in the clockwise order intersect, and we clear all regions $P - P(r_i)$ for $i = m, \cdots, 1$.

W.l.o.g., assume that any two regions $P(r_i)$ and $P(r_{i+1})$ intersect. Let $a = r_0 = Ray(r_0) = P(r_0)$. Assume also that the region $P(r_{i-1})$ has been cleared by now. In the following, we always denote by R the chain from r_{i-1} to r_i or from r to r_i where $r_{i-1} < r < r_i$, and by L the chain from $Ray(r_{i-1})$ to $Ray(r_i)$ or from $Backw(r)$ or $Forw(r)$ to $Ray(r_i)$.

Case 1 The vertex a is contained in both $P(r_{i-1})$ and $P(r_i)$.

Case 1.1 $i = 1$. Since all vertices in the chain R are visible to a, R is weakly visible to the chain L. Also, all vertices in L are visible from R. This is because if there is a vertex r in L that is invisible from any point in R, then three vertices a, r_1 and r would satisfy the condition **A2** (Fig. 5a-b). Hence, L is also weakly visible from R. Since all vertices in R are visible from a, there are no forward deadlocks between L and R. Furthermore, if a backward deadlock between L and R occurs, two reflex vertices forming the deadlock together with the vertex r_1 would satisfy the condition **A1** (Fig. 5c). Thus, the region $P(r_1)$ is walkable from vertex a to $\overline{r_1Ray(r_1)}$.

Case 1.2 $1 < i \le m$ *and* $\overline{r_{i-1}Ray(r_{i-1})}$ *intersects with* $\overline{r_iRay(r_i)}$. In this case, $r_{i-1} < r_i < Ray(r_{i-1}) < Ray(r_i)$. (It is not true that $r_{i-1} < Ray(r_i) < Ray(r_{i-1}) < r_i$; otherwise, the condition **A2-a** would be satisfied.) If the chain R is weakly visible to L, then we rotate the line segment $\overline{r_{i-1}Ray(r_{i-1})}$ into $\overline{r_iRay(r_i)}$ using instructions (iii). Otherwise, let r denote the first reflex vertex in R that blocks its adjacent vertex from being visible to any point in the opposite chain L. It is possible for r to block $Pred(r)$ from being visible to any point in the chain L (Fig. 6). (But, it is impossible for r to block $Succ(r)$; otherwise the vertex r would be critical.) In this case, the line segment $\overline{rForw(r)}$ intersects with $\overline{r_{i-1}Ray(r_{i-1})}$, but does not intersect with $\overline{r_iRay(r_i)}$ (Fig. 6). We first rotate the line segment $\overline{r_{i-1}Ray(r_{i-1})}$ into $\overline{rForw(r)}$. The current chain L from $Forw(r)$ to $Ray(r_i)$ is weakly visible from the current chain R from r to r_i; otherwise, r, r_i and the reflex vertex in L that blocks some of L would satisfy the condition **A1**. Also, there are no deadlocks because of the existences of r and r_i; otherwise, three vertices satisfying the condition **A1** could be found. If the current chain R is also weakly visible from the current chain L, then the region bounded by L, R, $\overline{rForw(r)}$ and $\overline{r_iRay(r_i)}$ is walkable. Otherwise, let r' be the first reflex vertex in the current chain R such that the line segment $\overline{r'Forw(r')}$ intersects with $\overline{rForw(r)}$, i.e., $Pred(r')$ is invisible to any point in the chain L. The line segment $\overline{rForw(r)}$ is then rotated into $\overline{r'Forw(r')}$. In this way, we can eventually reach a reflex vertex r^* such that the chain from r^* to r_i is weakly visible from the chain from $Forw(r^*)$ to $Ray(r_i)$, and then the line segment $\overline{r^*Ray(r^*)}$ can be moved into $\overline{r_iRay(r_i)}$ by a walk.

Case 1.3 $1 < i \leq m$ *and* $\overline{r_{i-1}Ray(r_{i-1})}$ *does not intersect with* $\overline{r_iRay(r_i)}$. Similar to the treatment for the situation in Case 1.2 where two ray shots does not intersect, we can move the line segment $\overline{r_{i-1}Ray(r_{i-1})}$ into $\overline{r_iRay(r_i)}$ by rotations and a walk.

Case 2 The vertex a is contained in $P(r_{i-1})$, but not in $P(r_i)$.

Case 2.1 $i = 1$. As r_1 is visible from a, we denote by $Ray(a)$ the point of P first hit by a "bullet" shot at a in the direction from a to r_1. Since all points preceding $Ray(a)$ are visible to a, we first clear the region right to $\overline{aRay(a)}$ and then rotate $\overline{aRay(a)}$ into $\overline{r_1Ray(r_1)}$.

Case 2.2 $1 < i \leq m$ *and* $\overline{r_{i-1}Ray(r_{i-1})}$ *intersects with* $\overline{r_iRay(r_i)}$. In this case, a is contained in the chain L. Then all vertices in the chain R are visible to the intersection of $\overline{r_{i-1}Ray(r_{i-1})}$ and $\overline{r_iRay(r_i)}$; otherwise, some reflex vertex in R would be critical. The chain R is then weakly visible from L, and we can rotate $\overline{r_{i-1}Ray(r_{i-1})}$ into $\overline{r_iRay(r_i)}$.

Case 2.3 $1 < i \leq m$ *and* $\overline{r_{i-1}Ray(r_{i-1})}$ *does not intersect with* $\overline{r_iRay(r_i)}$. In this case, $Ray(r_i) < r_{i-1} < Ray(r_{i-1}) < r_i$. (It follows from our assumption described above that $r_{i-1} < Ray(r_i) < r_i < Ray(r_{i-1})$ does not hold.) Hence, all reflex vertices in $P - P(r_{i-1})$ are not critical. Assume that neither r_{i-1} nor $Ray(r_{i-1})$ is visible to all vertices in $P - P(r_{i-1})$; in this case, $P - P(r_{i-1})$ can be simply cleared. Consider two shortest paths from a to r_{i-1} and $Ray(r_{i-1})$ in polygon P. These two shortest paths share segments for parts of their lengths; at some vertex v they part and proceed separately to their destinations. See Fig 7. The region bounded by $\overline{r_{i-1}Ray(r_{i-1})}$ and two shortest paths from v to r_{i-1} and $Ray(r_{i-1})$ is usually called a *funnel*. Each of the funnel paths is *inward convex*: it bulges in toward the funnel region. We further extend each edge on the side of the funnel until it hits the chain from r_{i-1} to $Ray(r_{i-1})$. Each resulting region, which is pseudo-triangular; the base of the region is a subset of the chain from r_{i-1} to $Ray(r_{i-1})$ and the apex of the region is the vertex opposite the base, is wholly visible from its apex. This is because no points proceeding an apex in the shortest path (oriented) from a to r_{i-1} or $Ray(r_{i-1})$ are visible to any point in its pseudo-triangular region, and thus any reflex vertex in the base is not critical with respect to the apex, too. The region $P - P(r_{i-1})$ can then be cleared by rotating each line segment extending a funnel edge within polygon P into the next in order, fixing the first center at r_{i-1}, starting and also ending with the line segment $\overline{r_{i-1}Ray(r_{i-1})}$ (Fig. 7). Since the region $P(r_{i-1})$ is 1-searchable from a to $\overline{r_{i-1}Ray(r_{i-1})}$, it is also 1-searchable from $\overline{r_{i-1}Ray(r_{i-1})}$ to a. Adding with the subschedule for clearing the region $P(r_{i-1})$ from $\overline{r_{i-1}Ray(r_{i-1})}$ to a, we obtain a complete schedule for clearing polygon P.

Case 3 The vertex a is not contained in $P(r_{i-1})$ nor $P(r_i)$. The treatment is similar to that in Case 1 or 2, and we omit the details in this extended abstract.

Case 4 The vertex a is contained in $P(r_i)$, but not in $P(r_{i-1})$. The treatment is similar to that in Case 1 or 2, and we omit the details in this extended abstract.

Case 5 The region $P(r_m)$ is cleared. Clearing the region $P - P(r_m)$ is symmetric to clearing the region $P(r_1)$ in Case 1.1 or Case 2.1. Hence, we can clear polygon P in this case.

All the cases described above ensure that the polygon P is 1-searchable. □

Theorem 3 *It takes $O(n^2)$ time to test the 1-searchability of a simple polygon, and $O(n^3)$ time to generate a search schedule if one exists.*

Proof. Omitted in this extended abstract □

4 Extension to 2-searchable polygons

Similar to the instructions given for 1-searchers, we can define the search instructions of 2-searchers and ∞-searchers. See also [4] for detail.

4.1 Searching a corridor by a 2-searcher

A generalization of the two-guard problem, called the *corridor search problem*, is considered in [1, 5]. Given a simple polygon with two marked vertices u and v, which is called a corridor, can the k-searcher, starting at u, force the intruder out of P through v? Again, the polygonal boundary is divided into two chains, L and R. Both chains L and R are oriented from u to v.

The concept of link-2-visibility is used in the algorithms for solving the corridor search problem [1, 5]. Two points x, $y \in P$ are said to be mutually *link-2-visible* if there exists another point z such that $\overline{xz}$ and $\overline{zy}$ are entirely contained within P. For two regions R, $Q \subseteq P$, we say that R is *weakly link-2-visible* from Q if every point in R is link-2-visible from some point in Q.

The ray shots can also be defined with link-2-visibility [5]. Assume that $p \in L$ is a reflex vertex such that $p < Backw(p) \in L$ holds. Let $S(u, p)$ and $S(u, Backw(p))$ denote the shortest paths from u to p and from u to $Backw(p)$, respectively. These two paths have to diverge at some vertex p_1. (Point p_1 might be u.) Since two shortest paths from p_1 to p and from p_1 to $Succ(p)$ are *inward-convex* polygonal chains; i.e., they are convex with convexity facing toward the interior of P, a portion of $\overline{pBackw(p)}$ is visible from p_1. Let p'_1 be the point on $\overline{pBackw(p)}$ visible from p_1 and closest to $Backw(p)$. The *backward link-2-ray shot* from p, denoted by $Backw^2(p)$, is the first point of P hit by a bullet shot at p'_1 in the direction from p'_1 to p_1. See Fig. 9 for an example. Note that if $Backw^2(p) \in R$, then it precedes all other points in R that are link-2-visible to $Succ(p)$. Similarly, let $S(v, p)$ and $S(v, Backw(p))$ denote the shortest paths from v to p and from v to $Backw(p)$, respectively. Let p_2 denote the vertex where two paths $S(v, p)$ and $S(v, Backw(p))$ diverge, and let p'_2 be the point on $\overline{pBackw(p)}$ visible from p_2 and closest to p. The *forward link-2-ray shot* $Forw^2(p)$ is the first point of P hit by the bullet shot at p'_2 in the direction from p'_2 to p_2 (Fig. 9). Also, if $Forw^2(p) \in R$, then it succeeds all other points in R that are link-2-visible to $Succ(p)$. Therefore, if $Forw^2(p)$, $Backw^2(p) \in R$, then all points link-2-visible to $Succ(p)$ are contained in $[Backw^2(p), Forw^2(p)]$. The link-2-ray shots for reflex vertices $p \in L$ such that $p > Forw(p) \in L$ and for reflex vertices in R can be defined analogously.

A pair of reflex vertices $p \in L$, $q \in R$ is then said to form a *backward link-2-deadlock* if $q < Backw^2(p) \in R$ and $p < Backw^2(q) \in L$ hold or a *forward link-2-deadlock* if $q > Forw^2(p) \in R$ and $p > Forw^2(q) \in L$ hold (see also Fig. 9 for an example).

Lemma 3 *[5] A corridor is 2-searchable if and only if the chains L and R are mutually weakly link-2-visible and no link-2-deadlocks occur.*

Lemma 4 *[5] It takes $O(n \log n)$ time to test the 2-searchability of a corridor, and $O(n \log n + k)$ time to generate a search schedule where k ($\leq n^2$) is the minimal number of search instructions.*

4.2 Searching a simple polygon by a 2-searcher

Any search schedule of the 2-searcher should start at some point (vertex) a of the boundary of P. We order the points of the boundary of P by a counterclockwise scan of the boundary of P, starting at a. For the reflex vertices r such that $r < Backw(r)$ or $r > Forw(r)$, we define the link-2-ray shots $Backw^2(r)$ and $Forw^2(r)$, as did in the previous section. (Remember that all points on the boundary of P are ordered in a counterclockwise scan starting at a, and for a complete ordering, the vertex a is considered as two vertices a_l and a_r such that $a_l \leq p \leq a_r$, for all points p in the boundary of P.) Then the necessary conditions for 1-searchable polygons can be extended to 2-searchable and ∞-searchable polygons as follows.

Theorem 4 *Polygon P is not ∞-searchable if one of the following conditions is true.*

*(**B1**) There are three points in P such that no point of the shortest path between any pair of points is link-2-visible to the third (Fig. 10a).*

*(**B2**) For any vertex a, there are two reflex vertices v_1 and v_2 such that (**B2-a**) $v_1 < Backw^2(v_1) < v_2$ and $v_1 < Forw^2(v_2) < v_2$ (Fig. 10b), or (**B2-b**) $v_1 < Backw^2(v_1) < v_2 < Backw^2(v_2)$ (Fig. 10c) or $Forw^2(v_1) < v_1 < Forw^2(v_2) < v_2$ (Fig. 10d).*

Proof. By an argument similar to the proof of Theorem 1. Note that the searchability is the same when the condition **B1** or **B2** is true, no matter how many flashlights the searcher has. □

Theorem 5 *A simple polygon P is 2-searchable if none of the conditions of Theorem 4 applies.*

Proof. Let a be a vertex of P that does not satisfy the condition **B2** of Thorem 4. We assume that the whole polygon P is not link-2-visible to a; in this case, polygon P can be simply cleared by a 2-searcher. A reflex vertex r is *link-2-critical* if $r < Backw^2(r)$ or $r > Forw^2(r)$. Let $Ray^2(r)$ denote the shot of the link-2-critical vertex r, and $\overline{rRay^2(r)}$ the pair of line segments connecting r and $Ray^2(r)$. Again, the orientation of $\overline{rRay^2(r)}$ is from r to $Ray(r)$. We denote by $P^2(r)$ and $P - P^2(r)$ the regions left and right to $\overline{rRay^2(r)}$, respectively.

The rest of the proof is similar to that of Theorem 2, and thus the details are omitted. □

It follows from Theorems 4 and 5 that the any polygon that is searchable by a ∞-searcher is also searchable by a 2-searcher, which confirms the conjecture due to Suzuki and Yamashita [4].

Theorem 6 *It takes $O(n^2)$ time to test the 2-searchability of a simple polygon, and $O(n^3)$ time to generate a search schedule if one exists.*

Proof. Omitted in this extended abstract. □

References

[1] D.Crass, I.Suzuki and M.Yamashita, "Search for a mobile intruder in a corridor", *IJCGA* **5** (1995) 397-412.

[2] P.J.Heffernan, An optimal algorithm for the two-guard problem, *IJCGA* **6** (1996) 15-44.

[3] C. Icking and R. Klein, The two guards problem, *IJCGA* **2** (1992) 257-285.

[4] I.Suzuki and M.Yamashita, "Searching for mobile intruders in a polygonal region", *SIAM J. Comp.* **21** (1992) 863-888.

[5] X.Tan, A note on the corridor search problem, In *Proc. Japan Conference on Discrete and Computational Geometry* (1998) 119-123.

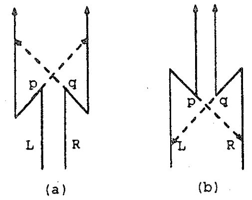

Figure 1

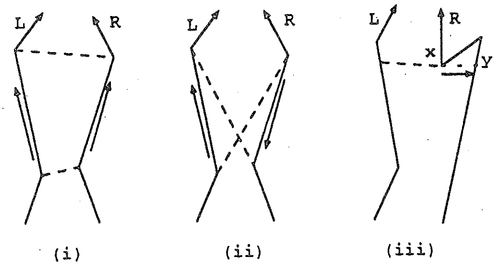

Figure 2

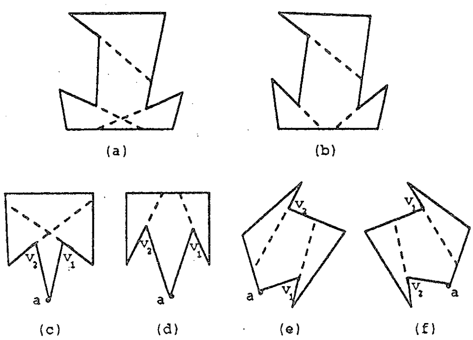

Figure 3

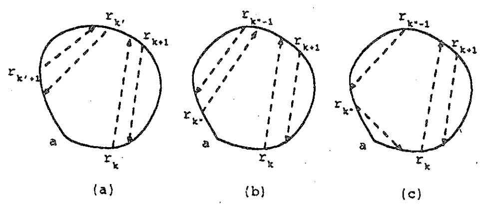

Figure 4

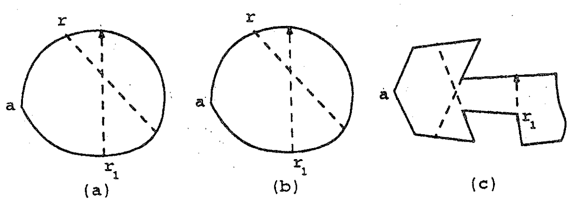

Figure 5: Case 1.1.

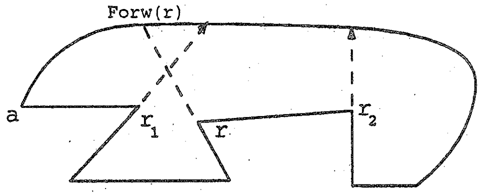

Figure 6

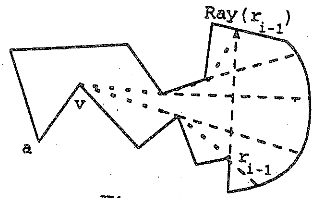

Figure 7

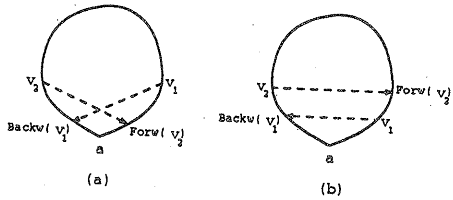

Figure 8

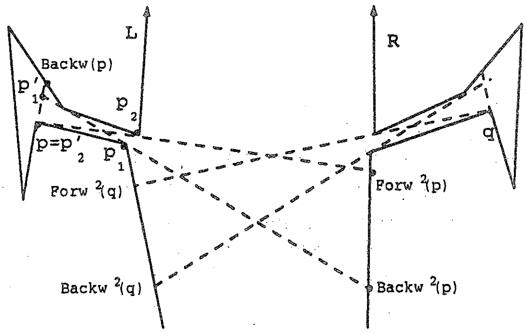

Figure 9

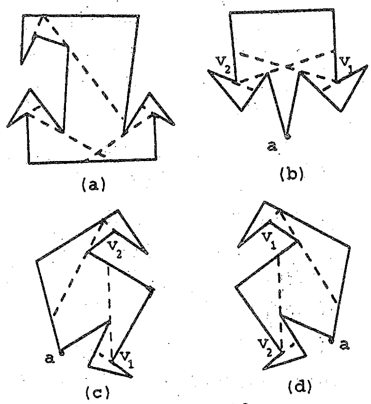

Figure 10