An Algorithm for Searching a Polygonal Region with a Flashlight*

Steven M. LaValle Borislav Simov Giora Slutzki

Dept. of Computer Science
Iowa State University
Ames, IA 50011 USA
{lavalle, simov, slutzki}@cs.iastate.edu

Abstract

We present an algorithm for a single pursuer with one flashlight that searches for an unpredictable, moving target with unbounded speed in a polygonal environment. The algorithm decides whether a simple polygon with n edges and m concave regions (m is typically much less than n, and always bounded by n) can be cleared by the pursuer, and if so, constructs a search schedule in time $O(m^2 + m \log n + n)$. The key ideas in this algorithm include a representation called the "visibility obstruction diagram" and its "skeleton," which is a combinatorial decomposition based on a number of critical visibility events. An implementation is presented along with a computed example.

1 Introduction

Consider the following scenario: in a dark polygonal region there are two moving points. The first one, called the **pursuer**, has the task to find the second one, called the **evader**. The evader can move arbitrarily fast, and his movements are unpredictable by the pursuer. The pursuer is equipped with a flashlight and can see the evader only along the illuminated line segment it emits. The pursuer (or 1-searcher [SY92]) **wins** if she illuminates the evader with her flashlight or if both happen to occupy the same point of the polygon. Clearly, the pursuer should, at all times, be located on the boundary of the polygon and use the flashlight as a moving boundary between the portion of the polygon that has been **cleared** (i.e., the evader is known not to hide there) and the **contaminated** portion of the polygon (i.e., the part in which the evader might be hiding). If there is a movement strategy of the pursuer whereby she wins regardless of the strategy employed by the evader, we say that the polygon is **searchable with one flashlight**, or **1-searchable**.

The problem above was introduced by Suzuki and Yamashita [SY92], who were interested in the existence and complexity of an algorithm which, given a simple polygon P with n edges, decides whether P is 1-searchable and if so, outputs a search schedule. Although the problem has been open for a while, no complete characterizations or efficient algorithms were developed. Naturally, several, restricted variants were considered. Independently of [SY92], Icking and Klein [IK92] defined the two-guard walkability problem, which is a search problem for two guards whose starting and goal position are given, and who move on the boundary of a polygon so that they are always mutually visible. Icking and Klein gave an $O(n \log n)$ solution, which later was improved by Heffernan [Hef96] to the optimal $\Theta(n)$. Tseng et al [THL98] extended the two-guard walkability problem by dropping the requirement that the starting and goal positions are given. They presented an $O(n \log n)$ algorithm which decides whether a polygon can be searched by two guards, and a $O(n^2)$ algorithm which outputs all of the possible starting and goal positions which allow searchability by two guards. Recently, Lee et al [LPC00] defined 1-searchability for a room (i.e., a polygon with one door — a point which has to remain clear at all times) and presented an $O(n \log n)$ decision

*An earlier version of this paper appeared at SoCG'2000.

algorithm and a method to construct a solution in time $O(n^2)$. In this paper we solve the original problem defined in [SY92], and we show that it is a nontrivial generalization of the variants of 1-searchability defined in [IK92] and [LPC00].

Originally, the problem of 1-searchability of a polygon was introduced in [SY92] together with a more general problem in which the pursuer (a.k.a. k-searcher) has k flashlights; when k is not bounded this corresponds to a 360° vision. For results concerning 360° vision refer to [SY92, CSY95, GLL$^+$97, LSC99] for search in polygons and to [LH99] for curved planar environments.

Our model is motivated in part by the need in mobile robotics systems to develop simple sensing mechanisms and to minimize localization requirements (knowing the precise location of the robot). The "flashlight" could be implemented by a camera and vision system that uses feature detection to recognize a target. Alternatively, a single laser beam could be used to detect unidentified changes in distance measurements. Many localization difficulties are avoided since the robot is required to follow the boundary of the environment. Sensors could even be mounted along tracks that are fastened to the walls of a building, as opposed to employing a general-purpose mobile robot. Although it is obviously restrictive to consider only environments that can be cleared by a single pursuer, the problem considered in this paper is rather challenging. In addition, it may be possible to extend some of the ideas in the algorithm for a single 1-searcher to allow the coordination of multiple pursuers, eventually broadening the scope of applications.

The rest of the paper is organized as follows. Section 2 introduces the notation and provides observations which reduce pursuit-evasion by a 1-searcher to a search problem on a torus. In Section 3.1 we define critical points on the boundary which are essential for determining the solution schedule of the 1-searcher. In Section 3.2 we use these points to reduce the search as presented in Section 2 to a search in a finite maze. In Section 4 we present an $O(m^2+m\log n+n)$ algorithm which, given a polygon with n edges and m concave regions, decides whether the polygon is searchable and if so, constructs a schedule for the 1-searcher. Section 5 discusses an implementation of the algorithm, and also the relationship between our result and the results in [IK92, Hef96, THL98], [EGHP$^+$00] and [LPC00]. Section 6 concludes the paper with a summary and directions for future research.

2 Notation and preliminaries

2.1 Notation

Let P be a simple polygon. (From now on, a polygon is always assumed to be simple.) We denote the **boundary** of P by ∂P. We assume that $\partial P \subseteq P$ and that ∂P is oriented in the **clockwise** (also called **positive**) direction. For any two points $a, c \in \partial P$, we write (a, c) to denote the open interval of all points $b \in \partial P$ such that when starting after a in positive direction along ∂P, b is reached before c. We also use the notation $[a, c]$, $[a, c)$ and $(a, c]$ for the closed and half-closed intervals on ∂P.

Let $p_0, p_1, \ldots, p_{n-1}$ denote the **vertices** on P ordered in the positive direction. The **edges** of ∂P are $e_0, e_1, \ldots, e_{n-1}$, where edge e_i has endpoints p_i and p_{i+1}, where $i \in \mathbb{Z}_n$ (i.e., the indices are computed modulo n; e.g., $p_0 = p_n$).

We use the standard definition of visibility. For points $c, d \in \partial P$ we say that d is **visible** from c, if every interior point of the line segment $\overline{cd}$ lies in $P - \partial P$. Obviously, if one point is visible from another, then the two are mutually visible. Note that no two points on the same edge of P are mutually visible.

2.2 Configuration: a snapshot of the pursuit

We start with a simple example of how the pursuer can clear the polygon in Figure 1(a). Initially, she is at point 0 with the flashlight pointing at 0. To start the search she moves from point 0 to point 14, while at the same time[1] she rotates the flashlight from point 0 to point 1. Next, the pursuer, while staying at 14, rotates the flashlight from point 1 clockwise to point 5. Then she moves from point 14 to point 13 constantly illuminating point 5. Following that, she rotates the flashlight from point 5 to point 7, and then moves from point 13 to point 10. After a final rotation of the flashlight from point 7 to point 8, the pursuer is at point 10 illuminating point 8. The search is completed when she moves from 10 to 9 and simultaneously rotates[1] the endpoint of the flashlight from 8 to 9.

The following observations are immediate. Because of his unbounded speed, we can assume that the evader moves only along the boundary. As we mentioned above, the pursuer also moves along ∂P and uses the flashlight to separate the clear and contaminated portions of the polygon. Furthermore, without loss of generality, we will assume in the rest of the paper that the clear portion of the polygon is always to the left of the pursuer as she looks in the direction of the beam of light emitted by her flashlight. We call this assumption the **left invariant**. Thus, a single pair of points is sufficient to record the current status of the pursuit as seen by the pursuer. We define a **configuration** to be a pair $\langle p, q \rangle$ of points $p, q \in \partial P$, and the space of all configurations X to be:

$$X = \partial P \times \partial P = \{ \langle p, q \rangle \mid p, q \in \partial P \} \ .$$

Let $X_d \subset X$ denote the set of all **diagonal configurations** $\langle p, q \rangle$ such that p and q lie on the same edge of ∂P. We denote the set of all **feasible configurations**, $X_v \subseteq X$, consisting of pairs $\langle p, q \rangle$ of mutually visible points, i.e., such that q is visible from p. Intuitively, for $\langle p, q \rangle \in X_v$, p represents the position of the pursuer, while q represents the point illuminated by the flashlight. If the left invariance assumption holds for configuration $\langle p, q \rangle$, then all points between p and q (along ∂P) are clear and all of the points between q and p are contaminated. Finally, let the set of all **non-feasible configurations**, $X_n = X - X_d - X_v$, consist of all pairs $\langle p, q \rangle$ of points which do not lie on the same edge and are not mutually visible. Thus a non-feasible configuration represents a situation, that cannot arise because of the geometry of P.

Definition 2.1 *The **visibility obstruction diagram(VOD)** for a polygon P is defined as the 3-partition of X, $\langle X_d, X_v, X_n \rangle$, where X_d, X_v and X_n are defined as above.*

Figure 1 provides an example of a simple polygon and the VOD corresponding to it. The black squares along the diagonal of Figure 1(b) represent X_d, the shaded area represents X_n, and the white area represents X_v.

2.3 Linking the snapshots together

In the previous section we have shown how for a polygon P with VOD X, we can represent a single snapshot of the pursuit as a feasible configuration in X_v. Next we are interested in a way to encode an entire strategy of the pursuer. Without loss of generality we can assume that the pursuer starts at time $t = 0$ and finishes at time $t = 1$. For every time t, $0 < t < 1$, let us denote by $T(t) \in X_v$ the feasible configuration that the pursuer is in at that time. That is, let $T(t) = \langle T_1(t), T_2(t) \rangle$, where $T_1(t) \in \partial P$ denotes the position of the pursuer and $T_2(t) \in \partial P$ denotes the point on the boundary

[1] Note that the beginning and the end of the search are the only times when the pursuer has to simultaneously move herself and rotate the flashlight.

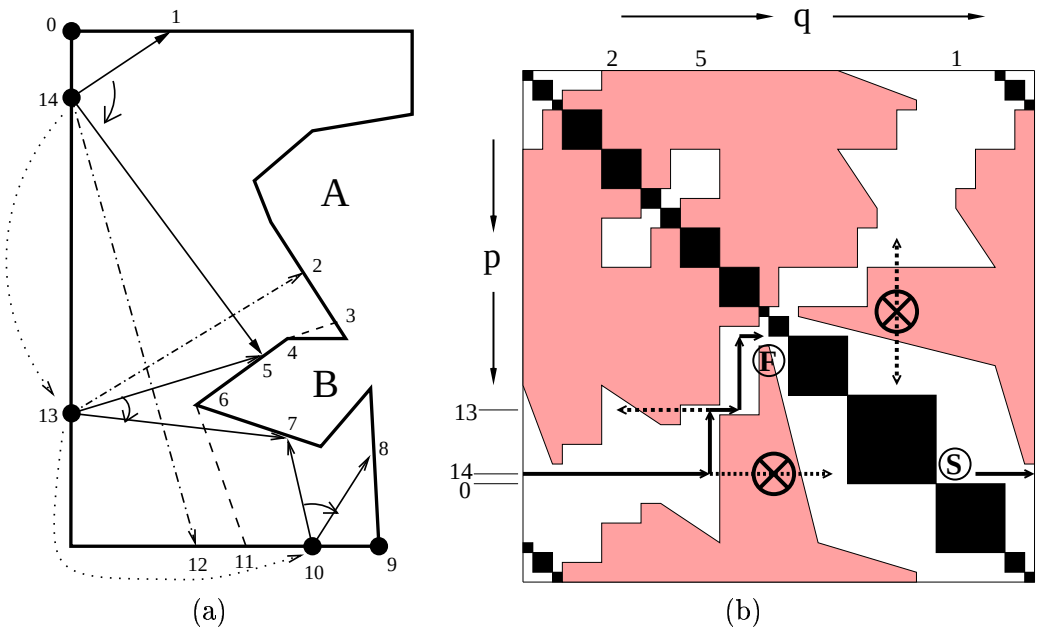


Figure 1: (a) a simple polygon (b) corresponding visibility obstruction diagram

illuminated by the flashlight at time t. We are interested in $T_1(t)$, $T_2(t)$ and $T(t)$ as functions of the time t, because we would like to represent a pursuer strategy (i.e., continuous motion of the pursuer and rotation of the flashlight) as a path in X_v. Clearly, every strategy of the pursuer can be mapped via T into a path in X_v. On the other hand, does every path in X_v have a corresponding pursuer strategy? The answer is "no". If $T(t)$ corresponds to a pursuer strategy, then it **does not contain**:

1. a **vertical jump**: this would imply discontinuity in $T_1(t)$, which is impossible since the pursuer must move continuously along the boundary. Therefore, if there are any discontinuities in $T(t)$, they have to be horizontal jumps.

2. a **horizontal jump entirely over points in** X_v: this would imply that a pursuer who is stationary at point $p \in \partial P$ and is rotating her flashlight from point q_1 to point q_2 does not illuminate some point between q_1 and q_2. But this is impossible, since all points between q_1 and q_2 are visible from p, and every visible point must be illuminated. Therefore, if there are any horizontal jumps in $T(t)$, they have to be over $X - X_v = X_d \cap X_n$.

3. a **horizontal jump over** X_d: this would correspond to the pursuer pointing the flashlight outside the polygon, which is impossible. Therefore, horizontal jumps in $T(t)$ are only possible over X_n.

4. a **left-to-right horizontal jump over configurations in** X_n: this would correspond to a stationary pursuer who tries to rotate the flashlight clockwise across an interval of invisible points in ∂P — this will invalidate the left invariant. For example, in Figure 1(a), if the pursuer is at point 14 and rotates the flashlight from point 5 to point 12 (over the invisible points between 6 and 11), this will cause the whole polygon to be contaminated again, thus losing all of the work to this point. This invalid move is shown as the left-to-right dotted arrow in Figure 1(b).

Finally, by combining (1)-(4) we conclude that if there is any discontinuity in $T(t)$, it can be only a horizontal jump from right to left over X_n. This corresponds to a stationary pursuer, who rotates the flashlight to the left across an interval of invisible points. For example, in Figure 1(a), if the pursuer is in point 13 and rotates her flashlight from point 5 to point 2 (over the interval of invisible points between 3 and 4), the move is represented in Figure 1(b) as the right-to-left dotted arrow in X. We call this a **recontamination move** since in the example originally the points on the boundary between 13 and 5 are clear, while after the move only the points between 13 and 2 remain clear, i.e. the interval between the points 2 and 5 is contaminated again.

Note that, technically, recontamination happens every time when the pursuer rotates the flashlight in counterclockwise direction. However, since counterclockwise rotation over visible points is reversible (i.e., a subsequent clockwise rotation of the flashlight to the original position preserves the left invariant), we consider it as a trivial form of recontamination and reserve the term "recontamination" for the irreversible counterclockwise rotation of the flashlight over an interval of non-visible points.

Definition 2.2 *Let $T(t) = \langle T_1(t), T_2(t)\rangle \in X_v$, $0 < t < 1$, be a path in X_v which has k discontinuities ($k \geq 0$) at times $t_1, t_2 \ldots, t_k$, such that $T(t)$ is continuous the rest of the time, and every discontinuity represents a horizontal jump from right to left over X_n. Let also $T(0) = \lim_{t\to 0^+} T(t)$ be the beginning of the path, such that, $T(0) = \langle p, p\rangle$ for some vertex $p \in \partial P$ and $T(t)$ approaches $T(0)$ from above or from the right. (This means that the path starts from a configuration in which the entire boundary is contaminated.) We define $T(t)$ to be a* ***legal path****. Also, if $T(t)$ is a legal path in which $T(1) = \lim_{t\to 1^-} T(t)$ is the end of the path, such that, $T(1) = \langle q, q\rangle$ for some vertex $q \in \partial P$ and $T(t)$ approaches $T(1)$ from below or from the left (this means that the path ends in a configuration in which the entire boundary is clear), we say that $T(t)$ is a* ***winning legal path*** *or simply* ***winning path****.*

For example, the winning path representing the solution described in Figure 1(a) is plotted as the rectilinear path of solid arrows in Figure 1(b) starting from the letter 'S' in the lower right corner, wrapping around[2] and ending at the letter 'F'.

From the discussion above it is clear that there is a one-to-one correspondence between a (winning) pursuer strategy and a (winning) legal path. This leads to the following statement.

Proposition 2.3 *Let P be a polygon with a VOD X. P is 1-searchable if and only if there exists a winning path in X.*

Note that $T_1(t)$ and $T_2(t)$ are treated differently. Since the pursuer moves continuously on the boundary, then $T_1(t)$ is also continuous. On the other hand, since the endpoint of the flashlight can jump over invisible intervals of ∂P, then $T_2(t)$ is **piecewise continuous**: we allow $T_2(t)$ to make horizontal jumps from right to left over X_n. This is the main difference between the problem of clearing a polygon with a single 1-searcher as compared to the problem of clearing a polygon with two guards [IK92]. The two-guards problem can also be represented as a search in the VOD, with the only difference that $T_2(t)$ and, therefore, $T(t)$ have to be continuous, i.e., the recontamination move is disallowed.

If for a polygon P there exists a winning path which does not contain a recontamination move, we say that P **can be cleared without recontamination**. On the other hand, if all of the legal winning paths for P contain a recontamination move, we say that **clearing P requires recontamination**. (For a more detailed example of a polygon, which requires recontamination, see Figure 7, discussed in detail in Section 5.)

[2]The circularity of ∂P implies that there is a vertical and horizontal wraparound along each of the axes, making X into a torus.

2.4 Compact encoding of a solution

In the previous section we showed that in order to check whether a polygon is 1-searchable, it suffices to verify whether there exists a winning path in its VOD. However, if in addition we would like our algorithm to describe the winning path (i.e., output a winning strategy of the pursuer) we have to take into account several configurations. If a polygon is 1-searchable, then the number of existing winning paths is infinite. They may vary by shape or length, so, intuitively, we would like to output a relatively simple (shorter, smoother, with less detours) path $T(t)$. Even in this case, it is clear that we cannot expect the algorithm to print all of the configurations $T(t)$ in the time interval $t \in [0, 1]$. This implies that instead, we have to print a finite sequence

$$T(t_0 = 0), T(t_1), T(t_2), \ldots T(t_{k-1}), T(t_k = 1) \quad ,$$

which conforms to the following two requirements:

- the length of the sequence, k, is as small as possible, (ideally, k is bounded by a polynomial in n, since we are looking for an algorithm which runs in polynomial time)
- each move from configuration $T(t_i)$ to configuration $T(t_{i+1})$, $0 \leq i < k$ belongs to a finite set of "elementary moves": these moves should be easy to describe, and straightforward to implement in a real-world application.

Note that the two requirements might be conflicting. Next, we present a reasonable tradeoff between them, by carefully defining of the set of "elementary moves".

2.4.1 The naive approach: rectilinear paths in X_v

The most straightforward, yet flawed, approach to choosing a set of elementary moves is to assume that during any elementary move either the pursuer is stationary or the flashlight illuminates the same point. Thus the motions of the pursuer and the rotations of the flashlight can be represented by vertical or horizontal directed segments in X as follows:

1. Flashlight rotation in clockwise direction over visible points and a stationary pursuer corresponds to a horizontal segment in X_v directed from left to right. In Figure 1, the rotation from point 1 to point 5 in the beginning of the solution is represented by the first (horizontal) arrow in the path.
2. Flashlight rotation in counterclockwise direction over visible points and a stationary pursuer corresponds to a horizontal segment in X_v directed from right to left.
3. Motion in counterclockwise direction of the pursuer with the flashlight illuminating the same point corresponds to a vertical segment in X_v directed up. In Figure 1, the move of the pursuer from point 14 to point 13 while illuminating point 5 is represented by the second (vertical) arrow in the path.
4. Motion in clockwise direction of the pursuer with the flashlight illuminating the same point corresponds to a vertical segment in X_v directed down.
5. Recontamination move

One advantage of choosing the five moves above as elementary ones is that they are simple and easy to describe. However, the examples in Figure 2 suggest, that there is a serious drawback if we

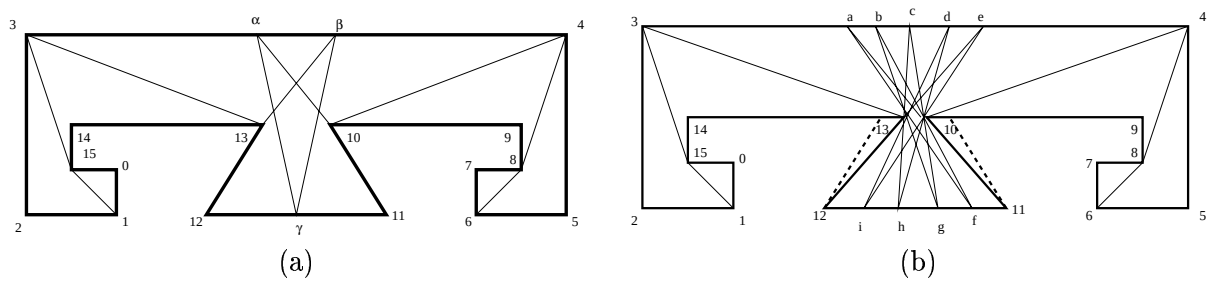


Figure 2: The size of a rectilinear legal path is not bounded by n

use a rectilinear moves to describe a winning strategy for the pursuer even in a relatively simple polygon. Detailed explanation follows.

Consider first the polygon in Figure 2(a). It can be searched, for example by the following rectilinear winning path:

$$\langle 0,0\rangle, \langle 15,1\rangle, \langle 15,3\rangle, \langle 13,3\rangle, \langle 13,\beta\rangle, \langle \gamma,\beta\rangle, \langle \gamma,\alpha\rangle, \langle 10,\alpha\rangle, \langle 10,4\rangle, \langle 8,4\rangle, \langle 8,6\rangle, \langle 7,7\rangle$$

While this is not the only rectilinear winning path, an important property of the polygon is that in order to get from configuration $\langle 13,3\rangle$ to configuration $\langle 10,4\rangle$, we need to introduce intermediate stopping points α, β and γ.

Let us now change slightly the shape of the polygon. We move vertices 13 and 10 horizontally towards each other, leaving the rest of the vertices intact, see Figure 2(b). The original position of the edges $\overline{10,11}$ and $\overline{12,13}$ is shown as a dashed line. How does a rectilinear winning path for the new polygon look like? It will have at least 9 intermediate stopping points:

$$\langle 0,0\rangle, \langle 15,1\rangle, \langle 15,3\rangle, \langle 13,3\rangle, \langle 13,e\rangle, \langle i,e\rangle, \langle i,d\rangle, \langle h,d\rangle, \langle h,c\rangle,$$
$$\langle g,c\rangle, \langle g,b\rangle, \langle f,b\rangle, \langle f,a\rangle, \langle 10,a\rangle, \langle 10,4\rangle, \langle 8,4\rangle, \langle 8,6\rangle, \langle 7,7\rangle$$

Note that the polygons in Figure 2(a) and (b) are very similar. In particular, they have the same number of edges, n, and furthermore, the visibility relation between edges and vertices is the same in the two polygons. Intuitively, the length of the minimal rectilinear winning paths should be the same. However, as we have shown, the second polygon requires a longer rectilinear winning path. Since we can use a similar construction to build a polygon in which the vertices 10 and 13 are arbitrarily close, it follows that we can increase the length of its rectilinear winning path arbitrarily while keeping n fixed. Thus, the length of the path and hence the time to describe a solution, does not depend on the number of edges n; therefore, the proposed set of elementary moves is unsuitable for a compact encoding of a solution.

2.4.2 Encoding a solution with length $O(n^2)$

Given that several other papers dealing with 1-searchability [SY92, LPC00] had to formally describe a strategy of the pursuer, how did they encode a solution? While a detailed definition can be found in [SY92, LPC00], we summarize the approach, which we call **conservative**, as follows. The set of elementary moves contains the recontamination move, and four other elementary moves that allow simultaneous pursuer motion and flashlight rotation, as long as those do not cross a vertex of ∂P. It can be seen that this approach limits the maximal length of a solution to $\Theta(n^2)$; therefore, the time required for an algorithm to print a solution is $\Omega(n^2)$.

There is a minor drawback to using the conservative set of instructions to describe a solution. Consider the following example from Figure 2(a). Suppose that the pursuer is at point 15 and is illuminating point 1. If the pursuer starts to rotate the flashlight in clockwise direction until it illuminates point 3, this intuitively represents a single move: $\langle 15, 1\rangle \rightarrow \langle 15, 3\rangle$. However, in the conservative approach, we have to break the rotation of the flashlight at every vertex that we illuminate, thus in this case, the rotation will be described as: $\langle 15, 1\rangle \rightarrow \langle 15, 2\rangle \rightarrow \langle 15, 3\rangle$. An extreme example would be the case of a regular n-sided polygon in which every pair of points is visible. Clearly, this is the simplest possible instance of the problem and can be cleared with a single rotation of the flashlight. On the other hand, if we use the conservative instructions, the single rotation has to be broken into a series of n instructions, one for each vertex that is illuminated by the flashlight.

Intuitively, it is not necessary to break up moves of the pursuer or rotations of the flashlight only because they cross a vertex. In the next section we show a set of elementary instructions which combines the advantages of the previous two approaches and allows us to encode every solution with $O(m)$ instructions, where m, the number of concave regions in the polygon is defined in Section 3.1.

2.4.3 Atomic moves

In this section we define the five atomic moves, which will be the basic building blocks for any legal path. One of the five moves is the recontamination move. The other four moves resemble the four conservative moves from Section 2.4.2: the pursuer is moving while simultaneously rotating the flashlight. We call the moves *northeast*, *northwest*, *southeast* and *southwest* since these are the directions of the respective paths in X.

Definition 2.4 *Let $p_1, p_2, q_1, q_2 \in \partial P$ be in clockwise order on the boundary, such that, $c' = \langle p_2, q_1\rangle, c'' = \langle p_1, q_2\rangle \in X_v$. Let also $R \in X - X_d$ be the rectangle with lower left corner c' and upper right corner c''. We say that there is an* ***atomic northeast move*** *from c' to c'' if either of the following conditions holds:*

(i) $R \subseteq X_v$, i.e., every configuration in R is feasible

(ii) For every feasible configuration $c \in R \cap X_v$, there is an atomic northeast move from c' to c and an atomic northeast move from c to c''.

Note that the condition recursively divides in move into two submoves. Condition (i) represents the base of this recursion. An atomic northeast move can be intuitively be considered as a chain of diagonal motions in the northeast direction in X.

Case (i) can be illustrated by the following example, see Figure 2(a). The move from configuration $\langle 14, 3\rangle$ to configuration $\langle 13, \beta\rangle$ is a northeast one. To show this, we choose $p_1 = 14$, $p_2 = 13$, $q_1 = 3$ and $q_2 = \beta$. Clearly, every point in $[q_1, q_2]$ is visible from every point in $[p_1, p_2]$. Thus the rectangle R, as defined above, contains only feasible configurations and the case (i) holds.

A move from $\langle 15, 1\rangle$ to $\langle 13, \beta\rangle$ is an example for an atomic northeast move, case (ii).

The atomic **southwest**, **southeast** and **northwest** moves are defined similarly.

An immediate observation is that each of the five rectilinear elementary moves defined in Section 2.4.1 is an atomic one. Furthermore, if we now use the atomic moves, a solution (i.e., winning path) for the polygon in Figure 2(a) can be represented as:

$$\langle 0, 0\rangle \longrightarrow \langle 15, 1\rangle \stackrel{NE}{\longrightarrow} \langle 13, \beta\rangle \stackrel{NW}{\longrightarrow} \langle 10, \alpha\rangle \stackrel{NE}{\longrightarrow} \langle 8, 6\rangle \longrightarrow \langle 7, 7\rangle \ ,$$

where NE and NW stand for the atomic northeast and northwest moves. An important observation is that Figure 2(b) has almost identical solution:

$$\langle 0,0\rangle \longrightarrow \langle 15,1\rangle \stackrel{NE}{\longrightarrow} \langle 13,e\rangle \stackrel{NW}{\longrightarrow} \langle 10,a\rangle \stackrel{NE}{\longrightarrow} \langle 8,6\rangle \longrightarrow \langle 7,7\rangle.$$

This illustrates the clear advantage of the set of atomic instructions compared to the rectilinear ones defined in Section 2.4.1.

Also, a solution for the case of a n-sided regular polygon can be described with a single north-eastern move, which is an advantage as compared with the conservative moves from Section 2.4.2.

3 Shelters and skeleton of the search space

In Section 3.1 we use simple geometric conditions to introduce "critical points", which provide the basis for determining the alternating order of the moves of the pursuer and the rotations of the flashlight. In Section 3.2 we use the critical points to construct a more compact, "skeleton" representation of the VOD.

3.1 Critical points

Vertex $p_i \in \partial P$ is a **reflex vertex** if the angle formed by incident edges e_{i-1} and e_i, in the interior of P, is greater that 180° (i.e., points p_{i-1}, p_i, and p_{i+1}, form a left turn). Otherwise, p_i is a **non-reflex vertex**. A maximal subinterval of ∂P of the form $C = (p_i, p_{j+1})$, in which all of the vertices $p_{i+1}, \ldots, p_j$ are reflex vertices, forms a **concave region**. Obviously, two concave regions cannot overlap and must be separated by non-concave regions (each of which contains at least one non-reflex vertex). For example, Figure 1(a) shows two concave regions labeled A and B.

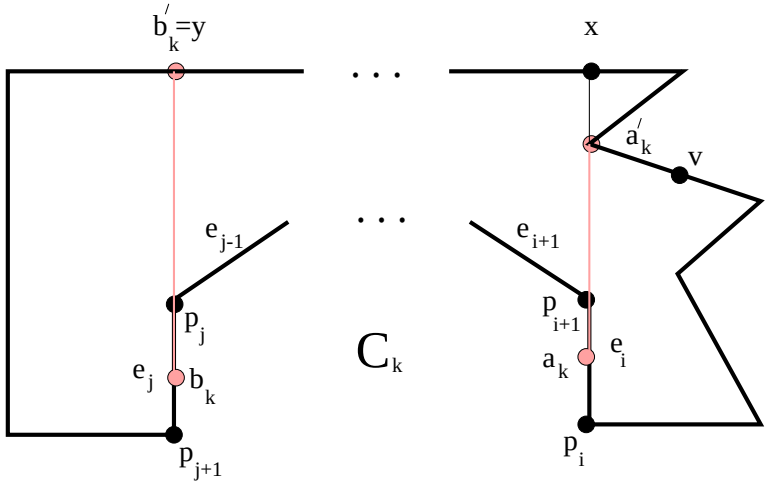


Figure 3: An illustration of critical points

Inasmuch as the concave regions of P represent the obstructions to visibility, their extreme edges represent the best hiding places for the evader (in view of his unbounded speed). Consider the k-th concave region $C_k = (p_i, p_{j+1})$ of P, see Figure 3. Define **shelters** of C_k to be the two points a_k and b_k, the midpoints of the edges e_i and e_j, respectively. Shoot a ray [CEG+94, HS95] starting at point a_k through p_{i+1} and let x be the point of ∂P where the ray leaves P for the first time. Define the **threshold** of a_k, denoted by a'_k, to be the point in $\overline{a_k x} \cap \partial P$ which is farthest from

a_k in positive direction along the boundary. Similarly, shoot a ray starting at point b_k through p_j and let y be the point of ∂P where the ray leaves P for the first time. Define the **threshold** of b_k, b'_k, to be the point in $\overline{b_k y} \cap \partial P$ which is farthest from b_k in negative direction along the boundary. Given the points a_k and a'_k define $\mathcal{H}(a_k) = (a_k, a'_k] \subset \partial P$. Note that a_k is not visible from any point in $\mathcal{H}(a_k)$. Thus, intuitively, whenever the pursuer stays in $\mathcal{H}(a_k)$, the evader can hide at a_k and is guaranteed not to be detected by the pursuer. Similarly, we can define $\mathcal{H}(b_k) = [b'_k, b_k) \subset \partial P$.

We have defined, for each concave region C_k of P, four types of important points: two shelters, a_k and b_k, and two corresponding thresholds, a'_k and b'_k. These are collectively called the **red points** of C_k and the sets of all such points, for the various concave regions of P, are denoted, according to types, by A, B, A', and B' respectively:

$$A = \{a_0, \ldots, a_{m-1}\}, \quad A' = \{a'_0, \ldots, a'_{m-1}\}$$
$$B = \{b_0, \ldots, b_{m-1}\}, \quad B' = \{b'_0, \ldots, b'_{m-1}\} .$$

Lemma 3.1 *Let P be a polygon with n edges and m concave regions. The critical points of P and their order along ∂P can be found in time $O(n + m \log n)$.*

Proof: The points in A and B can be found in time $\Theta(n)$. To find the points in A' and B' we can use the ray-shooting algorithms of Chazelle et al [CEG+94]. After a preprocessing time of $O(n)$, we can find each threshold by a single ray-shooting query in time $O(\log n)$. Thus we perform $2m$ queries for a total time $O(n + m \log n)$. ∎

3.2 Skeleton of X_n

One of the main ideas of our algorithm is that instead of trying to determine (and use) the exact shape of X_v, we will use the information obtained from the critical points to build an "equivalent" search space that is computationally much more convenient. The leads to a skeletal representation of the X_n in the VOD, and will allow us to construct a solution in time $O(m^2 + m \log n + n)$, where m is the number of concave regions of P. A detailed description of the new search space follows.

For any shelter $a_i \in A$ define the **horizontal wall** α_i^h and **vertical wall** α_i^v to be segments in X_n:

$$\alpha_i^h = \{\langle a_i, q\rangle \mid q \in \mathcal{H}(a_i)\}, \qquad \alpha_i^v = \{\langle p, a_i\rangle \mid p \in \mathcal{H}(a_i)\} .$$

Similarly, each shelter $b_i \in B$ induces the horizontal and vertical walls β_i^h and β_i^v respectively:

$$\beta_i^h = \{\langle b_i, q\rangle \mid q \in \mathcal{H}(b_i)\}, \qquad \beta_i^v = \{\langle p, b_i\rangle \mid p \in \mathcal{H}(b_i)\} .$$

Note that all walls are subsets of X_n.

Definition 3.2 *For a polygon P define the **skeleton** $\mathcal{S} \subset X_n$ to be*

$$S = \mathcal{D} \cup \mathcal{A}^h \cup \mathcal{B}^h \cup \mathcal{A}^v \cup \mathcal{B}^v, \text{ where:}$$

- $\mathcal{D} = \{\langle p, p\rangle \mid p \in \partial P\}$ *is the **diagonal wall**, or simply the **diagonal**,*
- $\mathcal{A}^h = \bigcup_{k=0}^{m-1} \alpha_k^h$, $\mathcal{B}^h = \bigcup_{k=0}^{m-1} \beta_k^h$, *are the two sets of **horizontal walls**,*
- $\mathcal{A}^v = \bigcup_{k=0}^{m-1} \alpha_k^v$, $\mathcal{B}^v = \bigcup_{k=0}^{m-1} \beta_k^v$, *are the two sets of **vertical walls**.*

Let α_i^h (resp. α_i^v) be a horizontal (resp. vertical) wall. The **tip** of α_i^h (resp. α_i^v) is the configuration $\langle a_i, a_i{}' \rangle \in X$ (resp. $\langle a_i{}', a_i \rangle \in X$). The tips of the walls β_i^h and β_i^v are defined similarly. Note that a tip (of a wall) need not belong to X_v. However, as the next lemma shows, for every tip τ (of a horizontal or vertical wall) there is a feasible neighborhood $C \subset X_v$ such that τ belongs to the (topological) closure of C.

Lemma 3.3 *Let $\langle a'_k, a_k \rangle$ be the tip of α_k^v. There exists a sufficiently small rectangle Q with left upper corner $\langle a'_k, u_1 \rangle$ and lower right corner $\langle v, u_2 \rangle$, where $a_k \in (u_1, u_2)$ and $a'_k \in (a_k, v)$, such that the interior of Q lies entirely in X_v. A similar statement holds for the tips of the other three types of walls.*

Proof: The proof follows directly from the definitions of the critical points a_k and a'_k. Let a_k lie on the edge $e_i = \overline{p_i p_{i+1}}$ of ∂P; refer to Figure 3 in Section 3.1. Let $u_1 = p_i$ and $u_2 = p_{i+1}$. Also, let $v \in (a'_k, a_k)$ be sufficiently close to a'_k, such that the entire edge e_i is visible from every point in the interval $(a'_k, v]$. Clearly, the interior of Q lies in X_v. ∎

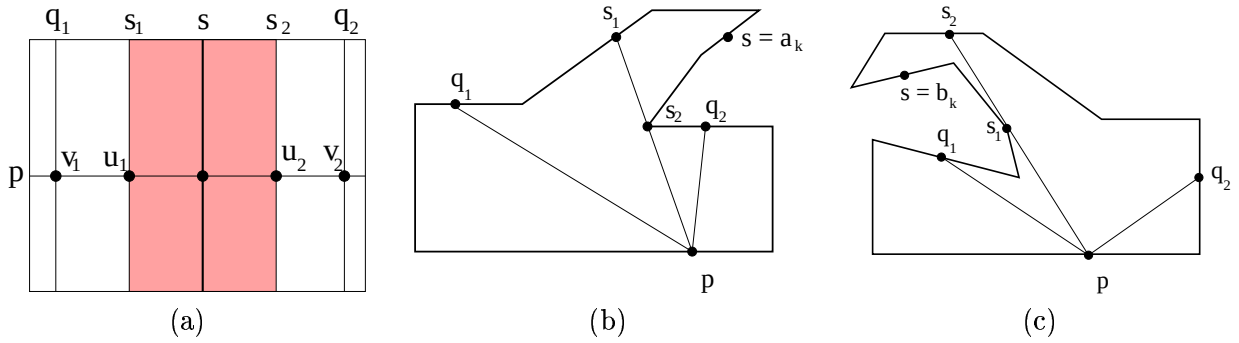


Figure 4: Illustration for the proof of Lemma 3.4

The next lemma establishes that all horizontal and vertical walls in $\mathcal{S}$ touch the diagonal.

Lemma 3.4 *Let $\overline{v_1 v_2}$ be a horizontal (resp. vertical) segment, such that $v_1, v_2 \in X_v$ and $\overline{v_1 v_2} \cap \mathcal{D} = \emptyset$. If $\overline{v_1 v_2} \cap X_n \neq \emptyset$, then $\overline{v_1 v_2} \cap (\mathcal{A}^v \cup \mathcal{B}^v) \neq \emptyset$ (resp. $\overline{v_1 v_2} \cap (\mathcal{A}^h \cup \mathcal{B}^h) \neq \emptyset$).*

Proof: The two cases being similar, we assume the hypotheses for a horizontal segment $\overline{v_1 v_2}$, where $v_1 = \langle p, q_1 \rangle$, $v_2 = \langle p, q_2 \rangle$, see Figure 4(a). By assumptions, there exists a point $s \in (q_1, q_2)$, such that $u = \langle p, s \rangle \in X_n$. Then there exist configurations $u_1 = \langle p, s_1 \rangle$ and $u_2 = \langle p, s_2 \rangle$, such that $\overline{u_1 u_2}$ is the maximal horizontal subsegment of $\overline{v_1 v_2}$ which contains u and whose interior lies in X_n. Since none of the points in (s_1, s_2) are visible from p, there must exist a concave region C_k which obstructs the view from p. Moreover, exactly one of the points s_1 and s_2 lies on C_k and is visible from p, while the other point is not visible from p and does not lie on C_k, see Figure 4 (b,c). We consider both cases:

- If $s_2 \in C_k$ and $\langle p, s_2 \rangle \in X_v$, then $s = a_k \in (s_1, s_2)$ and $p \in \mathcal{H}(a_k)$, see Figure 4(b). Thus $\overline{u_1 u_2} \cap \mathcal{A}^v \neq \emptyset$.
- If $s_1 \in C_k$ and $\langle p, s_1 \rangle \in X_v$, then $s = b_k \in (s_1, s_2)$ and $p \in \mathcal{H}(b_k)$, see Figure 4(c). Thus $\overline{u_1 u_2} \cap \mathcal{B}^v \neq \emptyset$.

From both cases it follows that $\overline{u_1u_2} \cap (\mathcal{A}^v \cup \mathcal{B}^v) \neq \emptyset$, and since $\overline{u_1u_2} \subset \overline{v_1v_2}$, the lemma follows. ∎

Lemma 3.3 and Lemma 3.4 illustrate the intuition behind the term "skeleton" because all horizontal and vertical walls in $\mathcal{S}$ span from the diagonal to a horizontal or vertical extrema (tips) of X_n. Thus, in a sense, these lines form a supporting frame for X_n. For example, Figure 5(a), shows the VOD together with the skeleton $\mathcal{S}$ for the polygon in Figure 1.

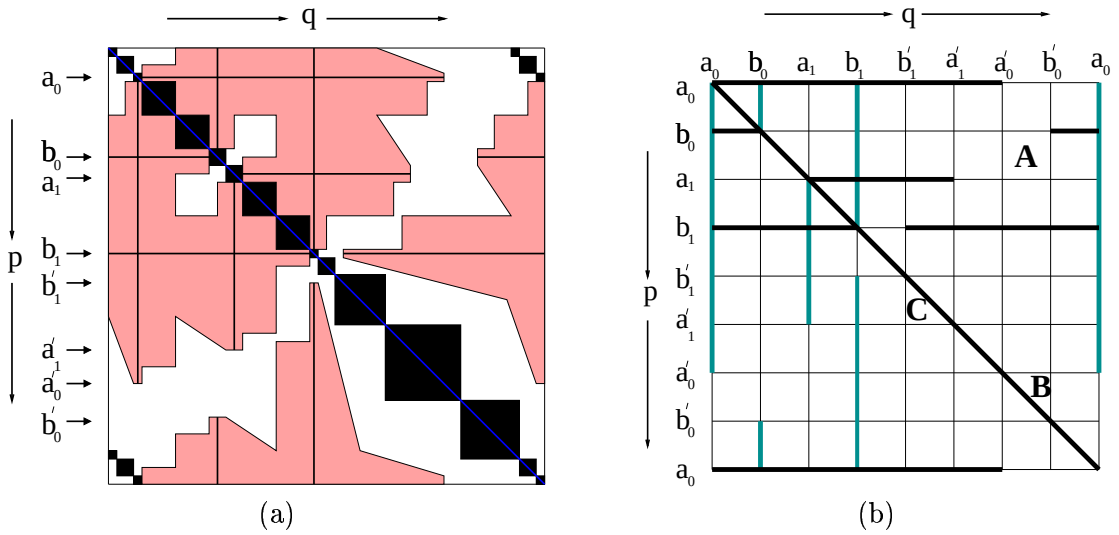


Figure 5: Skeleton (a) and square grid (b) of the polygon from Figure 1

Next we show that if we relax somewhat the restrictions that the regions X_n and X_d impose on legal paths, we can build a simpler representation of X, which, as we shall see, will allow us to find a solution in time $O(m^2 + m \log n + n)$. The idea is to replace the regions X_n and X_d by the skeleton $\mathcal{S}$. The new search space is called **skeletal obstruction diagram (SOD)** and is illustrated in Figure 5(b). It represents the skeleton $\mathcal{S}$ of the same polygon, drawn over the square grid formed by plotting the critical points along the axes. For convenience we assume that the configuration $\langle a_0, a_0 \rangle$ is in the upper left corner of the grid. Note that the thin lines which form the grid are added for reference only and are not part of the skeleton. The thick grey and black lines in the figure correspond to the vertical and horizontal walls of $\mathcal{S}$.

Definition 3.5 *A **relaxed path** in X is a continuous rectilinear path which does not cross the diagonal $\mathcal{D}$ or a horizontal wall and does not cross a vertical wall from left to right. A relaxed path which starts immediately above $\mathcal{D}$ and ends immediately below $\mathcal{D}$ is a **winning relaxed path**.*

Clearly, every legal path is also relaxed. The following theorem shows that clearing P is equivalent (up to some transformation) to finding a relaxed path in its corresponding SOD.

Theorem 3.6 *(i) Every winning legal path is a winning relaxed path. (ii) Every winning relaxed path can be transformed into a winning legal path.*

Proof: Since $\mathcal{S} \subset X_n \cup X_d$, the restrictions to a relaxed path are a subset of the restrictions to a legal path, and the correctness of (i) follows immediately. The proof of (ii) is rather lengthy and quite involved. A detailed argument is presented in the appendix. ∎

4 Finding a solution

4.1 Finding a relaxed path in time $O(m^2 + m \log n + n)$

The grid in Figure 5(b) is useful to illustrate compactly the restrictions imposed by the skeleton. Intuitively, we can think about a winning relaxed path as a path in a maze with horizontal and (one-way) vertical walls. For clarity of exposition we will present the search for a winning relaxed path in a directed graph $\mathcal{G}_P$ that represents (the restrictions on) the connectivity between neighboring squares of the grid. Suppose the sequence $\langle r_0, r_1, \ldots, r_{4m-1} \rangle$ consists of all of the critical points in A, A', B, and B', such that $r_0 = a_0$ and the elements of the sequence are ordered in the positive direction along ∂P.

$V(\mathcal{G}_P)$, the set of vertices in the graph consists of:

(v1) **non-diagonal vertices**, one for each square away from the diagonal; e.g., the square labeled with **A** in Figure 5(b),

(v2) **starting vertices**, one for each half-square, a triangle immediately above the diagonal; e.g., the half-square labeled with **B** in Figure 5(b), and

(v3) **goal vertices**, one for each half-square, a triangle immediately below the diagonal; e.g., the half-square labeled with **C** in Figure 5(b).

We define $E(\mathcal{G}_P)$, the edges in the graph as the maximum subset of $V(\mathcal{G}_P) \times V(\mathcal{G}_P)$ satisfying the following rules:

(e1) there can be edges only between neighboring (i.e., sharing a common grid line segment) squares,

(e2) there is no edge across the diagonal,

(e3) there is no edge across a horizontal wall,

(e4) there is no edge from left to right across a vertical wall, and

(e5) there is no edge from right to left across a vertical wall α^v (resp. β^v) if there are no horizontal walls crossing α^v (resp. β^v) between the edge and the tip of α^v (resp. β^v).

Rule (e1) ensures that a path in the graph represents a continuous relaxed path. Rules (e2), (e3) and (e4) are required by the definition of a relaxed path. Finally, rule (e5) guarantees that a relaxed path avoids unnecessary recontamination, i.e., we disallow a recontamination move between two squares if there is a path between them which does not require recontamination.

Finding a winning relaxed solution is equivalent to finding a path in the graph $\mathcal{G}_P$ from a starting to a goal vertex. Finding a path in $\mathcal{G}_P$ can be done by a breadth-first search and it takes time linear in the size of the graph.

Theorem 4.1 *There is an algorithm that, given a simple polygon P with n edges and m concave regions, decides whether P can be cleared by a 1-searcher, and if so, outputs a winning relaxed path in time $O(n + m \log n + m^2)$.*

Proof: An outline of an algorithm which, given a simple polygon, finds a winning relaxed path if one exists, is presented in Figure 6. The correctness of the algorithm follows immediately from Theorem 3.6 and the definition of the graph $\mathcal{G}_P$.

```
    FIND_RELAXED_SOLUTION()
1   Input a polygon as a sequence of points ⟨p_0, p_1, ... p_{n-1}⟩.
2   Determine the shelter points {a_0, ... a_{m-1}} and {b_0, ... b_{m-1}}.
3   Use ray shooting to find the threshold points a'_i, b'_i for i ∈ Z_m.
4   Sort the critical points into a sequence ⟨r_0, r_1, ..., r_{4m-1}⟩.
5   Construct the graph G_P.
6   Using BFS in G_P find a non-zero length path from D to D.
7   Output the path or a message if one does not exist.
```

Figure 6: Outline of the algorithm for finding a winning relaxed path

We have to show that the algorithm runs in time $O(n + m \log n + m^2)$. Steps (1) and (2) of the algorithm take time $\Theta(n)$. To perform step (3) we apply from Lemma 3.1, which states that after a preprocessing of $\Theta(n)$, we need to perform $2m$ ray shooting queries, each of which can be performed in time $O(\log n)$, thus the total time for step (3) is $O(m \log n + n)$. Step (4) takes time $O(m \log m)$. Steps (5) and (6) can be completed in time $O(m^2)$ since the graph has m^2 vertices and the outdegree of each vertex is bounded by 4.

Note that the number of concave regions cannot exceed the number of edges in the polygon, so the number of critical points, $4m$, is $O(n)$. Thus the total running time is $O(n+m \log n+m \log m+ m^2) = O(n + m \log n + m^2)$, which in the worst case, when $m = \Theta(n)$, is equivalent to $O(n^2)$. ∎

4.2 Transforming a relaxed path into a legal path

Note that the search in $\mathcal{G}_P$ will find a relaxed, but not necessarily legal, path. In time $O(m^2 + m \log n+n)$ we can transform a relaxed path into an encoding of a winning path using $O(m)$ atomic instructions. We do this in two different stages.

The first stage is to find the places where recontamination occurs in the relaxed path, i.e., where it crosses a vertical wall from right to left. Now we have to find a pair of configurations $c_0, c_1 \in X_v$, representing the corresponding recontamination move from c_0 to c_1, see Lemma A.4. Such a pair can be found in time $O(n_0+n_1+\log n+m)$, where n_0 and n_1 are the number of vertices in the respective concave regions. In order to compute the total time for finding all of the recontamination moves, we note that there can be at most two recontamination moves across a given concave region (one per hiding place in a region) and thus at most $O(m)$ recontamination moves. Thus, the total time is $O(n)$ (since the vertices of P are counted at most twice) $+O(m) \cdot O(\log n+m)) = O(n+m \log n+m^2)$.

Once we have broken the relaxed path into subpaths which do not contain any recontamination, we have to transform each path into a sequence of the other four atomic moves. This can be done iteratively in a procedure similar to the induction in Proposition A.5, for a total time of $O(m^2)$.

Thus, the total time for generating a description of a winning path consisting of atomic instructions is $O(n + m \log n + m^2)$. Note that in the worst case, when $m = \Theta(n)$ this is equivalent to $\Theta(n^2)$. However, in the cases in which m is much smaller than n, our algorithm is much faster than $O(n^2)$. Consider, for example, approximating curved planar environments [LH99] by simple polygons. In a sequence of arbitrarily fine approximations, n will tend to infinity; however, the number of concave regions, m, remains finite (m is equal to one-half of the number of inflection points along the curved boundary).

On the other hand, if we need a more detailed description of the sequence of vertices visited by the pursuer or illuminated by the flashlight, this can be done in time $O(n^2)$ by computing in $O(n)$ time the visibility polygon for each of the vertices and critical points as identified by the algorithm.

5 Implementation and comparison with previous work

The single-pursuer algorithm was implemented using GNU C++ and the Library of Efficient Data Types and Algorithms (LEDA), and experiments were performed on a Pentium III 500Mhz PC running Linux. The algorithm was determined to be efficient enough for practical use real environments. Figure 7 provides an example of the program output. Note that the position of the pursuer on the boundary is designated with a small white circle.

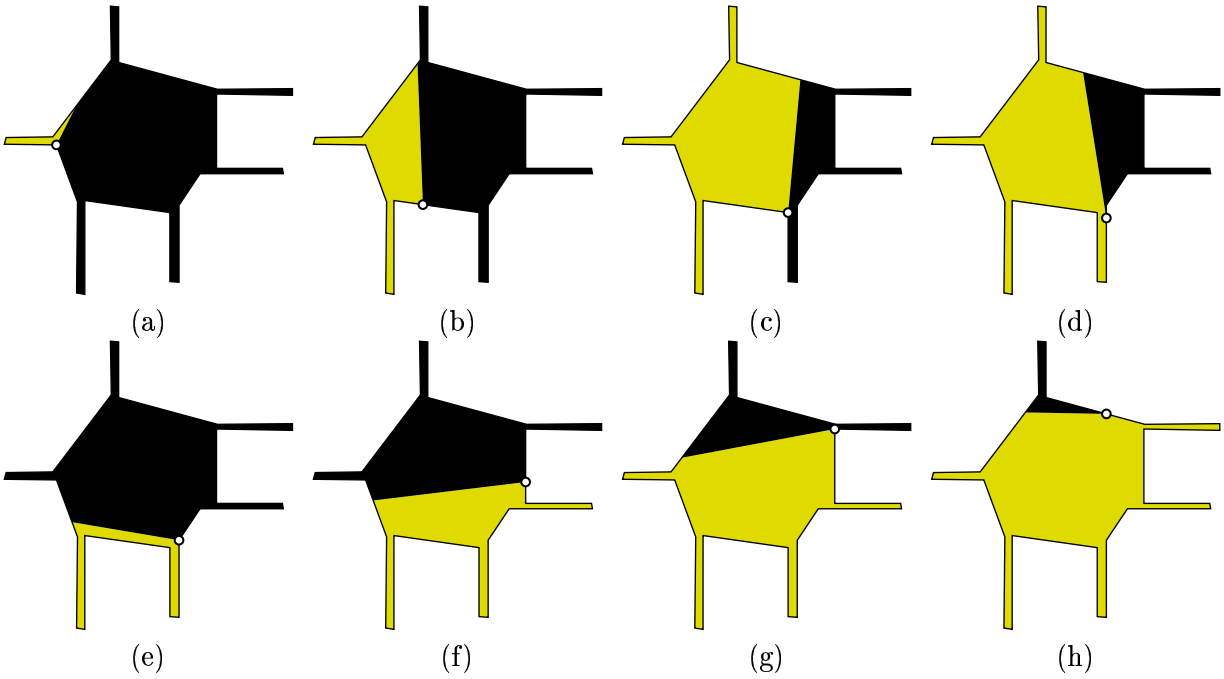


Figure 7: A polygon which requires recontamination (between frames (d) and (e)).

The polygon in Figure 7 is a simplified version of one from [SY92] and is interesting beyond a mere illustration of the algorithm implementation. It represents a polygon which **requires** a recontamination in order to be searched successfully. The recontamination happens between frames (d) and (e): the endpoint of the flashlight **jumps** over an interval of ∂P, thus some of the area which was already cleared is contaminated again after the pursuer rotates the beam to the left.

It is important to note that a schedule for a 1-searcher which contains a recontamination move cannot be simulated by a corresponding search schedule with two guards [IK92, Hef96]. The reason is that the two guards cannot maintain visibility and stay on ∂P while attempting to simulate a jump of the flashlight. Therefore, there are polygons that can be cleared by a 1-searcher but cannot be cleared by two guards. Since every schedule for two guards is a schedule for a 1-searcher as well, it follows that the set of polygons that can be cleared by a 1-searcher is a strict superset of the polygons that can be cleared by two guards. Also, note that while the polygon in Figure 7 can be cleared by a chain of three guards using an algorithm by Efrat et al [EGHP$^+$00] (this is generalization of the two guards problem to a chain of k guards), this is not equivalent to finding a solution for a 1-searcher.

Similarly, it is not hard to show that every point in the polygon in Figure 7 is contaminated at some time during any successful 1-searcher schedule. Thus there is no point d such that the room $\langle P, d\rangle$ can be cleared by a 1-searcher as described in the $O(n^2)$ algorithm of Lee et al [LPC00].

6 Conclusion

In this paper we have presented an $O(m^2 + m \log n + n)$ algorithm which, given a simple polygon with n edges and m concave regions, decides whether the polygon can be cleared by a 1-searcher and if so, outputs a search schedule. The algorithm is a nontrivial generalization of the two-guard search algorithm and solves a (rather longstanding) problem left open in [SY92]. The most interesting extension of the result in the current paper would be an algorithm which, given a polygon and an integer k, decides whether the polygon can be cleared by k 1-searchers, and ideally, returns a search schedule. Note that the problem does not impose any restrictions on the mutual visibility between the 1-searchers which, we showed in Section 5, allows clearing of a strictly greater set of polygons as compared to a chain of $k + 1$ guards. Finally, while the proposed problem for k 1-searchers is NP-hard for polygonal regions which contain holes [YUSK97], little is known about the complexity of the problem for simple polygons.

Acknowledgments

We are grateful for the funding provided in part by NSF CAREER Award IRI-9875304 (LaValle).

References

[CEG+94] B. Chazelle, H. Edelsbrunner, M. Grigni, L. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12:54–68, 1994.

[CSY95] 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 Journal of Computational Geometry and Applications*, 5(4):397–412, 1995.

[EGHP+00] A. Efrat, L. J. Guibas, S. Har-Peled, D. C. Lin, J. S. B. Mitchell, and T. M. Murali. Sweeping simple polygons with a chain of guards. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2000. to appear.

[GLL+97] L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani. Visibility-based pursuit-evasion in a polygonal environment. In F. Dehne, A. Rau-Chaplin, J.-R. Sack, and R. Tamassia, editors, *WADS '97 Algorithms and Data Structures (Lecture Notes in Computer Science, 1272)*, pages 17–30. Springer-Verlag, Berlin, 1997.

[Hef96] P.C. Heffernan. An optimal algorithm for the two-guard problem. *International Journal of Computational Geometry and Applications*, 6:15–44, 1996.

[HS95] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *Journal of Algorithms*, 18(3):403–431, May 1995.

[IK92] C. Icking and R. Klein. The two guards problem. *International Journal of Computational Geometry and Applications*, 2(3):257–285, 1992.

[LH99] S. M. LaValle and J. E. Hinrichsen. Visibility-based pursuit-evasion: The case of curved environments. In *Proceedings, IEEE International Conference on Robotics and Automation (ICRA)*, pages 1677–1682, Detroit, MI, USA, May 1999.

[LPC00] J.-H. Lee, S.-M. Park, and K.-Y. Chwa. Searching a polygonal room with a door by a 1-searcher. *International Journal of Computational Geometry and Applications*, 2000. to appear.

[LSC99] J.-H. Lee, S. Y. Shin, and K.-Y. Chwa. Visibility-based pursuit-evasion in a polygonal room with a door. In *Proceedings, ACM Symposium on Computational Geometry (SCG)*, pages 281–290, Miami Beach, FL, USA, June 1999.

[SY92] I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal on Computing*, 21(5):863–888, 1992.

[THL98] L. H. Tseng, P. Heffernan, and D. T. Lee. Two-guard walkability of simple polygons. *International Journal of Computational Geometry and Applications*, 8(1):85–116, February 1998.

[YUSK97] M. Yamashita, H. Umemoto, I. Suzuki, and T. Kameda. Searching for mobile intruders in a polygonal region by a group of mobile searchers. In *Proceedings, ACM Symposium on Computational Geometry (SCG)*, pages 448–450, Nice, France, June 1997.

A Proof of Theorem 3.6, part (ii)

In this section we provide the proof that a winning relaxed path can be transformed into a winning legal path.

In general, X_v consists of a finite number of maximal connected regions. These are called **conservative regions**, because any path within a region preserves the left invariant[3]. For example, there are three (two small and one large) conservative regions in Figure 5(a).

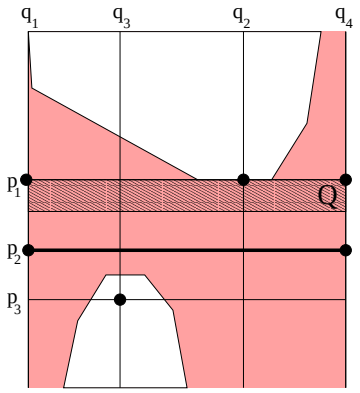


Figure 8: Horizontal separation of conservative regions in the proof of Lemma A.1.

The following lemma states that conservative regions which do not overlap horizontally are separated by horizontal walls.

Lemma A.1 *Let C be a conservative region, and let the configuration $\langle p_1, q_2\rangle \in X$ be a local minimum of the boundary of C. For $q_1, q_4 \in \partial P$, such that $q_2 \in (q_1, q_4)$, suppose that $c_1 = \langle p_1, q_1\rangle$ and $c_2 = \langle p_1, q_4\rangle$ are configurations from X_n which do not lie on the boundary of C. Let Q be a sufficiently small rectangle (possibly a segment) with an upper edge $\overline{c_1c_2}$, such that all interior points of Q lie in X_n. For $q_3 \in (q_1, q_4)$, if $\langle p_3, q_3\rangle \in X_v - Q$, then it follows that there is a horizontal wall which separates p_1 and p_3, i.e., there exists $p_2 \in \partial P$, $p_2 \in (p_1, p_3)$ such that the horizontal segment between $\langle p_2, q_1\rangle$ and $\langle p_2, q_4\rangle$ is a part of a horizontal wall.*

Proof: The proof relies on an observation of Guibas et al [GLL+97] that when the pursuer moves along the boundary, an interval of visible points appears/disappears exactly when a point on a bitangent line is crossed[4]. Note that in the lemma, the existence of a local minimum $\langle p_1, q_2\rangle$ for the conservative region C means that when a pursuer is moving in the positive direction along the boundary, a visible interval neighboring point q_2 disappears exactly when the pursuer crosses point p_1. Therefore, p_1 must be a point where a bitangent intersects ∂P. Assume that the pursuer continues to move clockwise on the boundary past point p_1. How soon can point q_3 become visible? No point $x \in (q_1, q_4)$, including q_3, is visible before the pursuer crosses another point p_1' of the bitangent. Since there is a point $p_2 \in A \cup B$ such that $p_2 \in (p_1, p_1')$ and $(q_1, q_4) \subset \mathcal{H}(p_2)$, it follows that there is a corresponding horizontal wall which includes the segment between the configurations $\langle p_2, q_1\rangle$ and $\langle p_2, q_4\rangle$. ∎

Since describing the exact shape of the conservative regions is rather complicated, we would

[3]The conservative regions in our paper have a meaning close to the conservative cells defined by Guibas et al [GLL+97].

[4]These are the "green points" defined in [GLL+97].

like to enclose every region in a rectilinear boundary which will be easier to construct and explore.

Definition A.2 *Given a configuration $c \in X - \mathcal{S}$, the set of all configurations reachable from c without crossing the skeleton is called a* ***visibility tile*** *(of c).*

Note that every conservative region of X is contained in some visibility tile, and a tile is a simply-connection region that has a rectilinear boundary. If a tile does not contain a conservative region, it is an **empty tile**. Otherwise it is **nonempty**.

Lemma A.3 *Every tile contains at most one conservative region. Thus, for a nonempty tile T, if $c_0, c_1 \in X_v \cap T$, then there is a path between c_0 and c_1 entirely within $X_v \cap T$.*

Proof: The claim is trivial for an empty tile, so assume that T is a nonempty tile. Let c_0, c_1 be two configurations, such that $c_0, c_1 \in X_v \cap T$. Since $c_0, c_1 \in T$, then there is a finite rectilinear path π within T from c_0 to c_1. We will prove by induction on k, the number of segments in π that there is a rectilinear path from c_0 to c_1 entirely within $X_v \cap T$, i.e., along feasible configurations.

The basis of the induction includes two cases: $k = 1$ and $k = 2$. The case for $k = 1$ follows immediately from Lemma 3.4. The case for $k = 2$ follows from Lemma A.1. Assume that the statement is true for all paths of length less than k, $k \geq 3$.

Consider a rectilinear path $\pi = \langle c_0 = \pi_0, \pi_1, \ldots, \pi_k = c_1 \rangle$ within T. The inductive step includes separate arguments for monotonous[5] and nonmonotonous rectilinear paths.

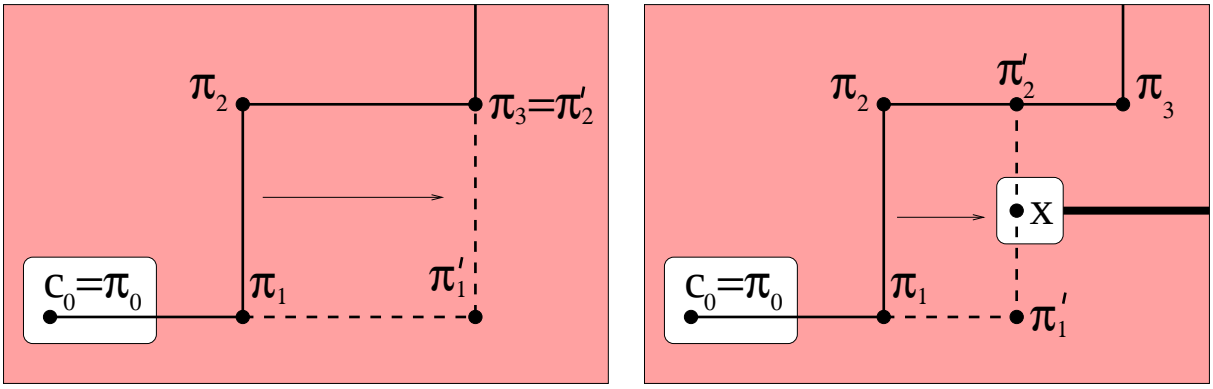


Figure 9: Inductive hypothesis for a monotonous path, Lemma A.3

First, let us assume that π is monotonous. Without loss of generality, assume that the segments $\overline{\pi_0\pi_1}$, $\overline{\pi_1\pi_2}$ and $\overline{\pi_2\pi_3}$ are left-to-right, upward, and left-to-right, respectively, see Figure 9. (The rest of the cases are similar.) Let us start to translate the segment $\overline{\pi_1\pi_2}$ horizontally from left to right to the segment $\overline{\pi'_1\pi'_2}$, where one of the next two events will occur:

- $\pi'_2 = \pi_3$, see Figure 9(a).

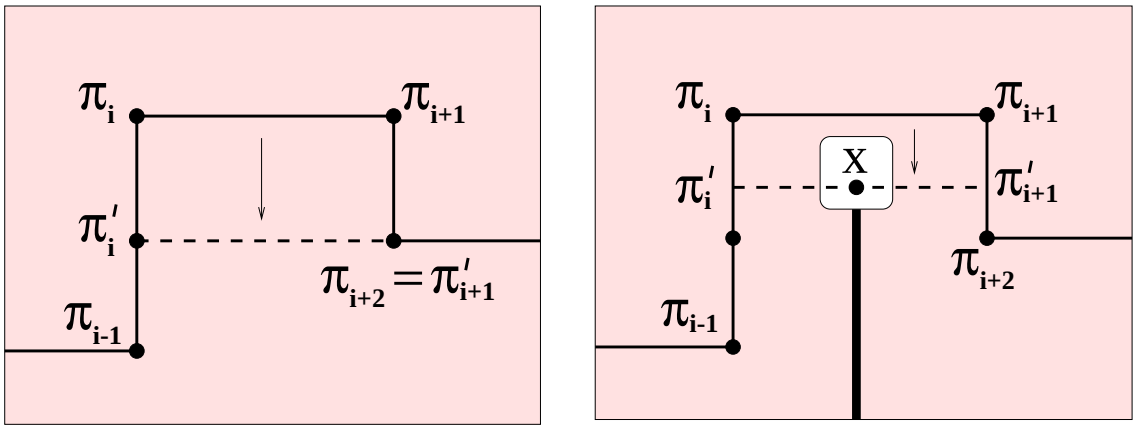
  Then the path $\langle c_0 = \pi_0, \pi'_1, \pi'_2 = \pi_3, \ldots, \pi_k = c_1 \rangle$ has length less than k and the starting points are in X_v so the induction hypothesis applies.

[5] We define a "monotonous rectilinear path" as a rectilinear path, the projection of which on any of the horizontal (resp. vertical) axis is either a nondecreasing or a nonincreasing horizontal (resp. vertical) path.

- $\overline{\pi'_1\pi'_2}$ will enter the white tip of a horizontal or vertical wall, see Figure 9(b)

  That is, there exists a point $x \in X_v \cup \overline{\pi'_1\pi'_2}$. Then we can apply the inductive hypothesis to the paths $\langle c_0 = \pi_0, \pi'_1, x\rangle$ and $\langle x, \pi'_2, \pi_3, \ldots, \pi_k = c_1\rangle$.

We have shown that, if the path has length k and is monotonous, the statement holds.



(a) (b)

Figure 10: Inductive hypothesis for a nonmonotonous path, Lemma A.3

Second, we consider the case in which the path π is not monotonous. This implies, that there exists i such that the segments $\overline{\pi_{i-1}\pi_i}$ and $\overline{\pi_{i+1}\pi_{i+2}}$ are in opposite directions.

Without loss of generality, assume that $\overline{\pi_{i-1}\pi_i}$, $\overline{\pi_i\pi_{i+1}}$ and $\overline{\pi_{i+1}\pi_{i+2}}$ are upward, left-to-right and downward, respectively, see Figure 10. (The rest of the cases are similar.) Let us start to translate the segment $\overline{\pi_i\pi_{i+1}}$ downward to the segment $\overline{\pi'_i\pi'_{i+1}}$, where one of the next three events will occur:

- $\pi'_{i+1} = \pi_{i+2}$, see Figure 10(a).

  Then the path $\langle c_0 = \pi_0, \ldots, \pi_{i-1}, \pi'_i, \pi'_{i+1} = \pi_{i+2}, \ldots, \pi_k = c_1\rangle$ has length less than k and the starting points are in X_v so the induction hypothesis applies.

- $\pi'_i = \pi_{i-1}$.

  Then the path $\langle c_0 = \pi_0, \ldots, \pi_{i-1} = \pi'_i, \pi'_{i+1}, \pi_{i+2}, \ldots, \pi_k = c_1\rangle$ has length less than k and the starting points are in X_v so the induction hypothesis applies.

- $\overline{\pi'_i\pi'_{i+1}}$ will enter the white tip of a vertical wall, see Figure 10(b).

  That is, there exists a point $x \in X_v \cup \overline{\pi'_i\pi'_{i+1}}$. Then we can apply the inductive hypothesis to the paths $\langle c_0 = \pi_0, \ldots, \pi_{i-1}, \pi'_i, x\rangle$ and $\langle x, \pi'_{i+1}, \pi_{i+2}, \ldots, \pi_k = c_1\rangle$.

We have shown that, if the path has length k and is not monotonous, the statement holds.

Combining the proofs of the cases for monotonous and nonmonotonous paths, we conclude that the lemma is true. ∎

Let T_0 and T_1 be two nonempty tiles, and let p, q_0, q_1 be points in ∂P, such that $q_0 \in (p, q_1)$ and the configurations $d_0 = \langle p, q_0\rangle$ and $d_1 = \langle p, q_1\rangle$ are interior points in T_0 and T_1, correspondingly. If the line segment $\overline{d_0d_1}$ crosses neither the diagonal nor any nonempty tiles other than T_0 and T_1,

we say that T_0 is a **left neighbor** of T_1, see Figure 11. We assume that neither of the tiles borders the diagonal line. We can make the assumption since our goal with the relation "left neighbor" is to represent the recontamination move of the pursuer, and recontamination from or to a tile which borders the diagonal is not useful. If there is recontamination to a nonempty tile which is immediately above the diagonal, then we can ignore all of the previous moves, since this tile is reachable directly from the diagonal. Similarly, if there is recontamination from a nonempty tile which is immediately below the diagonal, we can ignore the rest of the moves, since the diagonal is reachable before the recontamination.

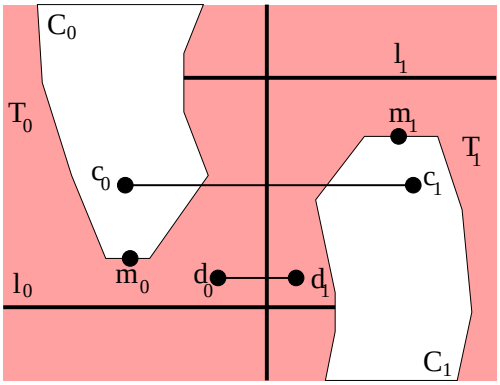


Figure 11: Position of C_0, C_1, T_0, and T_1 in the proof of Lemma A.4.

Lemma A.4 *If T_0 and T_1 are two nonempty tiles such that T_0 is a left neighbor of T_1, then the corresponding conservative regions, C_0 and C_1, overlap horizontally, i.e., there exist $c_0 = \langle p, q_0 \rangle \in C_0$ and $c_1 = \langle p, q_1 \rangle \in C_1$. Also, the segment $\overline{c_0 c_1}$ does not cross any conservative regions other than C_0 and C_1.*

Proof: Note that if one of the two tiles T_i is unbounded (on torus), then the corresponding conservative region C_i is unbounded and it overlaps horizontally with the whole vertical axis, which proves the claim. Therefore, it suffices to consider the case in which both tiles are bounded from above and below by horizontal walls. Let $L_0 = \langle l_0, \cdot \rangle$ (respectively, $L_1 = \langle l_1, \cdot \rangle$) be the lowest (resp., highest) horizontal wall which borders T_0 (resp., T_1). Similarly, let $m_0 = \langle m_0', m_0'' \rangle$ (resp., $m_1 = \langle m_1', m_1'' \rangle$) be the global minimum (resp., maximum) on the boundary of the conservative region of T_0 (resp., T_1), see Figure 11.

Since T_0 is a left neighbor of T_1, there exists a horizontal edge $\overline{d_0 d_1}$ which extends from T_0 to T_1. Without loss of generality, assume that $\overline{d_0 d_1}$ lies between lines L_1 and L_0. (The case in which $\overline{d_0 d_1}$ lies between lines L_0 and L_1 is similar, we just have to redefine the lines L_0 and L_1 to the highest and lowest horizontal lines of the corresponding regions.) If we assume that $m_0' \notin (l_1, l_0)$ or $m_0' \notin (l_1, l_0)$, we can apply Lemma 3.3 to show that there are nonempty tiles between T_0 and T_1. This would contradict the hypothesis; therefore, it follows that $m_0', m_1' \in (l_1, l_0)$. For the sake of contradiction, assume that the white regions do not overlap horizontally, i.e., $m_0' \in (l_1, m_1')$. We can apply Lemma A.1 by choosing m_0 for the local minimum in the lemma, and any point in X_v sufficiently close to m_1 for the configuration in X_v. It follows that there is a horizontal wall which crosses the rectangle with left upper corner m_0 and lower right corner m_1. This implies that the horizontal wall lies between the lines L_1 and L_0, so either L_0 is not a bordering line of T_0, or L_1 is not a bordering line of T_1. We have reached a contradiction; therefore, our assumption that $m_0' \in (l_1, m_1')$ is false. It follows that C_0 and C_1 overlap horizontally, i.e., there exist $c_0 = \langle p, q_0 \rangle \in C_0$ and $c_1 = \langle p, q_1 \rangle \in C_1$.

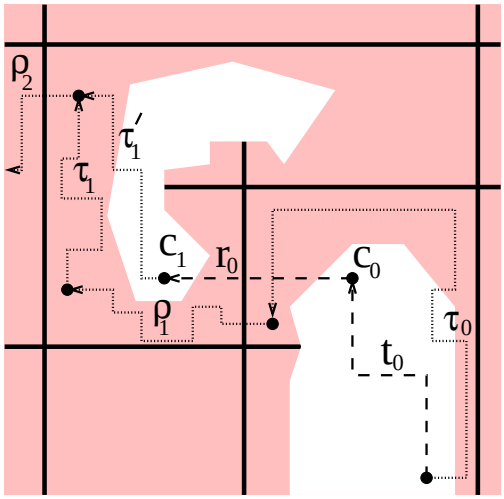


Figure 12: Inductive step in Proposition A.5.

Using Lemma 3.3 again, we can show that there are no nonempty tiles (other than T_0 and T_1) in the rectangle with left lower corner m_0 and right upper corner m_1. It follows that the horizontal segment $\overline{c_0 c_1}$ does not cross any conservative regions other than C_0 and C_1. ∎

Every relaxed path π which starts and ends at a nonempty tile can be partitioned into a sequence of subpaths τ_0, ρ_1, $\tau_1, \ldots, \rho_k$, τ_k such that every subpath τ_i lies completely inside a nonempty tile and every subpath ρ_i starts in the tile of τ_{i-1}, ends in the tile of τ_i, and crosses no other nonempty tiles. On the other hand, a legal path p can be partitioned into a concatenation of subpaths $t_0, r_1, t_1, \ldots, r_k, t_k$ where each t_i is a legal path inside the same conservative region and r_i is a horizontal path which starts from the conservative region of t_{i-1}, ends in the conservative region of t_i, and does not cross other conservative regions. Intuitively, a path t_i corresponds to a sequence of the four reversible moves of the pursuer as defined in section 2.2. A horizontal segment r_i (always right-to-left) corresponds to the fifth, recontamination move: stationary pursuer, rotating the flashlight counterclockwise across an interval of invisible points.

We will show that we can transform every relaxed path π into a legal path p, by replacing every subpath τ_i and ρ_i with legal subpaths t_i and r_i. A formal description follows. For convenience, we use $s(\pi)$ and $f(\pi)$ do denote the start and the end of a path π (same for path p).

Proposition A.5 *Every relaxed path $\pi = \tau_0 \cdot \rho_1 \cdot \tau_1 \ldots \rho_k \cdot \tau_k$ with $s(\pi), f(\pi) \in X_v$ and starts and ends in X_v can be transformed into a legal path $p = t_0 \cdot r_1 \cdot t_1 \ldots r_k \cdot t_k$, where τ_i, ρ_i, t_i and r_i are defined as above and $s(\pi) = s(p)$ and $f(\pi) = f(s)$.*

Proof: The proof is by induction on k, the number of times the path π switches between nonempty tiles.

[Basis, k = 0]. If $\pi = \tau_0$, this implies that π starts and ends in the same conservative region and never leaves its original tile. From Lemma A.3 it follows that there exists a path $p = t_0$, such that $s(p) = s(\pi)$, $f(p) = f(\pi)$, and t_0 lies entirely within X_v. Thus p is a legal path.

[Hypothesis, k < n]. Assume that the proposition is true for all $k < n$.

[Inductive step, k = n]. Let $\pi = \tau_0 \cdot \rho_1 \cdot \tau_1 \ldots \rho_n \cdot \tau_n$ be as described in the proposition. We have to show that it can be transformed into a legal path $p = t_0 \cdot r_1 \cdot t_1 \ldots r_n \cdot t_n$. Let T_0 and T_1 be

the first two nonempty tiles that π visits, i.e., $\tau_0 \subset T_0$ and $\tau_1 \subset T_1$. From Lemma A.4 it follows that there exists a pair of configurations $c_0 \in X_v \cap T_0$ and $c_1 \in X_v \cap T_1$ such that $r_0 = \overline{c_0 c_1}$ is a horizontal segment in X, see Figure 12. Also, by the assumption, we have $s(\tau_0) = s(\pi) \in T_0 \cap X_v$, and since $c_0 \in T_0 \cap X_v$, from Lemma A.3 it follows that there exists a legal path $t_0 \subset T_0 \cap X_v$ with $s(t_0) = s(\tau_0)$ and $f(t_0) = c_0$. Finally, there is a path $\tau_1' \subset T_1$ such that $s(\tau_1') = c_1$ and $f(\tau_1') = f(\tau_1) = s(\rho_2)$.

We apply the inductive hypothesis to the relaxed path $\tau' = \tau_1' \cdot \rho_2 \cdot \tau_2 \ldots \rho_n \cdot \tau_n$ to get the legal path $p' = t_1 \cdot r_2 \cdot t_2 \ldots r_n \cdot t_n$. We define the legal path $p = t_0 \cdot r_0 \cdot p'$ thus proving the inductive step. ∎

Lemma A.6 *Every winning relaxed path can be transformed into a winning legal path.*

Proof: Follows immediately from Proposition A.5 and also from the fact that, for every nonempty tile T which borders the diagonal, the corresponding conservative region C also borders the diagonal. ∎