Locating and Capturing an Evader in a Polygonal Environment

Volkan Isler[1*], Sampath Kannan[1**], and Sanjeev Khanna[1***]

University of Pennsylvania, Department of Computer and Information Science
Philadelphia, PA, USA
{isleri,kannan,sanjeev}@cis.upenn.edu

Abstract. This paper contains two main results: First, we revisit the well-known *visibility based pursuit-evasion* problem and show that, in contrast to deterministic strategies, a single pursuer can locate an unpredictable evader in any simply-connected polygonal environment using a randomized strategy. The evader can be arbitrarily faster than the pursuer and it may know the position of the pursuer at all times but it does not have prior knowledge of the random decisions made by the pursuer. Second, using the randomized algorithm together with the solution of a known lion and man problem [12] as subroutines, we present a strategy for two pursuers (one of which is at least as fast as the evader) to quickly capture an evader in a simply-connected polygonal environment.

1 Introduction

Pursuit-evasion games are among the fundamental problems studied by Robotics researchers. In a pursuit-evasion game, one or more pursuers try to capture an evader who, in turn, tries to avoid capture indefinitely. A typical example is the homicidal chauffeur game where a driver wants to collide with a pedestrian and the goal is to determine conditions under which he can (not) do so. Among the numerous applications of this game are collision avoidance and air traffic control.

Recently, there has been increasing interest in capturing intelligent evaders with certain sensing capabilities who are contaminating a complex environment. Aside from its obvious applications in search-and-rescue and surveillance, this problem provides a natural test-bed for many mobile robot algorithms.

Perhaps the most well-understood game in this context is the *visibility-based pursuit-evasion* where one or more pursuers try to locate an evader in a polygonal environment [13,10,2]. In this game, the evader is very powerful: it has unbounded speed and global visibility, meaning that it knows the location

* Supported in part by NSF-IIS-0083209, NSF-IIS-0121293 and MURI DAAH-19-02-1-03-83.

** Supported in part by NSF Grants CCR0105337 and CCR9820885.

*** Supported in part by an Alfred P. Sloan Research Fellowship and an NSF Career Award CCR-0093117.

of the pursuers at all times. In [15,4] the authors study a similar game in a probabilistic framework and propose a greedy algorithm.

The main ingredient of a pursuit-evasion game is the presence of an adversarial evader who actively avoids capture. Due the this aspect, the solutions to pursuit-evasion problems are game theoretic and distinguish themselves from their target finding counterparts (e.g. [11,14]) where the evader's motion is independent from the pursuer's.

In our present work, we propose randomized pursuer strategies for the visibility based pursuit-evasion problem. Randomization is a powerful technique which allows us to solve many problems that are not solvable by deterministic algorithms and has found wide-spread applications in many areas ranging from computational geometry to cryptography.

As we show in the following sections, it turns out that randomization provides a drastic increase in the power of the pursuers. For example, it is known that there are simply-connected environments where $\Theta(\log n)$ pursuers are required [2] in order to locate the evader with deterministic strategies. In contrast, we show that a single pursuer can locate the evader in any simply-connected environment with high probability, even if the evader knows the pursuer's location at all times and has unbounded speed (Theorem 1). The power of randomized strategies comes from the fact that the evader has no prior knowledge of the random decisions inherent in such strategies. It is worth noting that the randomized strategies work against any evader strategy and require no prior information about the strategy of the evader.

We also address the harder task of capturing the evader. For this problem we present a strategy for two pursuers, one of which is at least as fast as the evader. The strategy is based on the randomized strategy to locate the evader and the solution of a known lion and man problem [12] which is reviewed in Section 3.1. The same strategy can be used to capture the evader while protecting a door. This problem was introduced in [7] to model scenarios where the goal is to locate the evader which may leave the polygonal area through a door and win the game.

The two-pursuer strategy can be modified so that a single pursuer can also capture the evader. However, the expected time to capture in this case, though finite, may be significantly longer than the expected time to capture with two pursuers.

1.1 Randomized strategies

The power of randomization in the context of pursuit-evasion games is nicely illustrated by the example in Figure 1. A similar example can be found in [2].

In this example, a single pursuer can never locate the evader using a deterministic strategy: Let us distinguish four points A, B, C and D as shown in the figure. We will slightly abuse the notation and use these points to refer to the maximal convex region inside the polygon that contains these points as well. Now suppose the pursuer has a deterministic strategy of visiting those

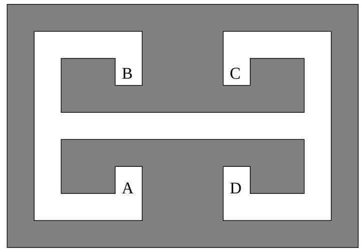

Fig. 1. A single pursuer can not capture an evader using deterministic strategies.

points in the order A, B, C, D. In this case, the evader can first hide at B and escape to D while the pursuer is visiting A. Afterwards, it can repeat the same strategy and escape to B when the pursuer is at C. If the pursuer visits the points in a different order, it is easy to see that the evader can find a similar strategy and avoid the pursuer. Therefore, in this polygon one pursuer can never locate the evader.

Now consider the following randomized strategy: Instead of committing to a deterministic strategy, the pursuer selects one of the regions $\{A, B, C, D\}$ uniformly at random and visits that point. It is easy to see that if the pursuer guesses the region where the evader is located, then the evader can not escape and the probability of this desired event is $\frac{1}{4}$. The crucial observation is that since the evader does not know which region the pursuer will visit, it can not choose a strategy based on the order of points visited by the pursuer.

One might suspect that the desired event of locating the evader is not guaranteed since the pursuer can keep guessing a wrong corridor forever. However, this probability can be made arbitrarily small by repeating the same strategy a few times. This is because, if k is the number of trials, the probability of missing in all k trials is only $(\frac{3}{4})^k$ in this example and this probability decreases exponentially in k. In general, if the probability of capture is p, the expected number of rounds to capture is $\frac{1}{p}$. Note that each round is independent. We can obtain the expected time to locate the evader as follows: Since the length of a round is bounded by the time to travel between two furthest points in the polygon (say T), the expected time to capture is $\frac{T}{p}$. We can convert the expected time to a high probability argument by repeating the experiment roughly $\frac{1}{p} \log \frac{1}{p}$ times using the Chernoff bound. For details of this technique the reader is referred to [8].

1.2 Preliminaries

Let P be the input polygon including its interior and V be the set of vertices of P. Unless stated otherwise, n denotes the number of vertices of the polygon. Two points $u, v \in P$ can *see* each other if the line segment uv lies entirely in P.

We use $d(u, v)$ to denote the length of the shortest path from u to v that remains inside P. The shortest path has the following property.

Property 1. The shortest path between any two points u and v inside a polygon P is a polygonal path whose inner vertices are vertices of P.

The *shortest path tree from a point x in P* is defined as $\cup_{v \in V(P)} d(x, v)$, where $V(P)$ is the set of vertices of P.

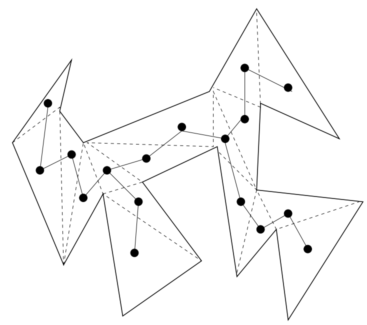

Fig. 2. Triangulation of a polygon and its dual tree.

The *diameter* of the polygon P, denoted $diam(P)$, is defined as $\max_{u,v \in P} d(u, v)$. Without loss of generality, we assume that the units are normalized so that the time it takes to travel the diameter is also $diam(P)$. A polygon is *simply-connected* if any simple closed curve inside the polygon can be shrunk to a point. All the polygons considered in this paper are simply-connected.

The *triangulation* of a polygon is a decomposition of the polygon into triangles by a maximal set of non-intersecting diagonals (see Figure 2). The dual of a triangulation is a graph whose vertices correspond to the triangles. There is an edge between two vertices if the corresponding triangles share a side. It is well known that the triangulation of a simply-connected polygon has exactly $n - 2$ triangles where n is the number of vertices of the polygon. In addition, the dual of the triangulation is a tree [9].

2 Locating the evader

In this section, we present a randomized algorithm for the visibility-based pursuit-evasion problem [2]:

An evader whose location is unknown to the pursuer is hiding inside a polygon P. The evader has global visibility (i.e. it knows the location of the pursuer at all times) and can be arbitrarily faster than the pursuer. Moreover, the evader is unpredictable – that is, no prior information about its strategy is available. The pursuer's goal is to locate the evader by establishing line-of-sight visibility: He wins the game if he ever sees the evader. The evader wins the the game if it can avoid being located forever.

Next, we show that for any simple polygon P, the pursuer can locate the evader in $O(n \cdot diam(P))$ expected time.

2.1 The pursuer strategy

Given polygon P, the pursuer first triangulates the polygon. The pursuer's strategy is divided into rounds of length at most $diam(P)$. Let T be the triangulation tree (see Figure 2) rooted at the triangle that contains the pursuer's initial location at the beginning of a round. For any triangle t let $t_1, .., t_k$, $k \leq 3$ be the children of t. We use the notation $T(t)$ to denote the subtree of T rooted at the triangle t. Figure 3 is provided for quick reference to the notation used in this section.

The pursuer's strategy relies on the following observation: Suppose the pursuer is inside triangle t and the evader is located inside a triangle contained in $T(t_j)$ for some j. Then, while the pursuer is located at t, the evader can not enter any triangle contained in $T(t_i)$, $i \neq j$ without being seen by the pursuer. This is because the triangle t is a separator for the subtrees $T(t_i)$. Moreover, this property is preserved if the pursuer moves to the triangle t_j.

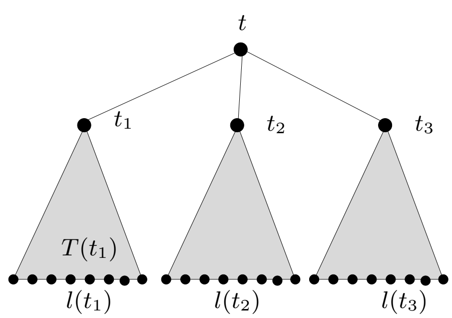

Fig. 3. Notation used for Lemma 1. Each vertex of the tree corresponds to a triangle in the triangulation tree.

Therefore, had the pursuer known the subtree that contains the evader, he could gradually move towards it while preventing the evader to move from one subtree to another. This process guarantees that the pursuer can enter the triangle which contains the evader and this clearly implies that the evader would be located. Of course, the pursuer does not know where the evader is. This is where we will utilize randomization.

The pursuer will guess the subtree that contains the evader according to the following rule:

Let $l(t)$ denote the number of leaves of the subtree $T(t)$. Suppose the pursuer is located in triangle t and let $t_1, .., t_k$ be the children of t (see Figure 3).

Let $L = \sum_{i=1}^{k} l(t_i)$. With probability $\frac{l(t_i)}{L}$, the pursuer picks the child t_i and moves there. The round is over whenever the pursuer arrives at a leaf of T.

Next, we show that using this guessing strategy, the pursuer efficiently locates the evader.

Lemma 1. *Let T be the triangulation tree rooted at the triangle that contains the pursuer's initial location in the beginning of the round. At each round, if the pursuer follows the guessing strategy described above, he can locate the evader with probability at least $\frac{1}{L}$.*

Proof. The lemma is proven by induction on the height of T. The basis, where the height of t is 0, corresponds to the case where the input polygon is a triangle. The pursuer trivially locates the evader with probability 1 in this case.

Let $p(t)$ be the probability that the evader is located within a round, starting from triangle t and inductively assume that the lemma is true for all trees of height less than or equal to j.

Given a triangulation tree of height $j+1$, the probability of success starting from the root t is:

$$p(t) \geq \min\left\{\frac{l(t_1)}{L}p(t_1), \ldots, \frac{l(t_k)}{L}p(t_k)\right\} \tag{1}$$

Note that all the subtrees $T(t_i)$ have height at most j, therefore by the inductive hypothesis we have $p(t_i) \geq \frac{1}{l(t_i)}$ for all i and the lemma follows.

Clearly, the number of leaves of any triangulation tree is less than the number of vertices of the polygon, therefore at each round the evader is located with probability at least $\frac{1}{n}$. Moreover, since the length of a round is $diam(P)$, we have the main result of this section:

Theorem 1. *In any simply connected environment P, against any evader strategy, the expected time to locate the evader with a single pursuer is at most $n \cdot diam(P)$ where n is the number of vertices and $diam(P)$ is the diameter of the polygon.*

Remark 1. Any simply-connected polygon can be partitioned into a minimum number of disjoint convex polygons in polynomial time [5,1]. It is easy to see that the dual of such a partition will be a tree. Therefore, instead of using a triangulation dual, the pursuer can execute the strategy described above using the dual of the convex partition. However, in general this does not improve the expected capture time. For example, for the polygon shown in Figure 5, the number of leaves of the triangulation dual is equal to the number of leaves of a minimum convex partition.

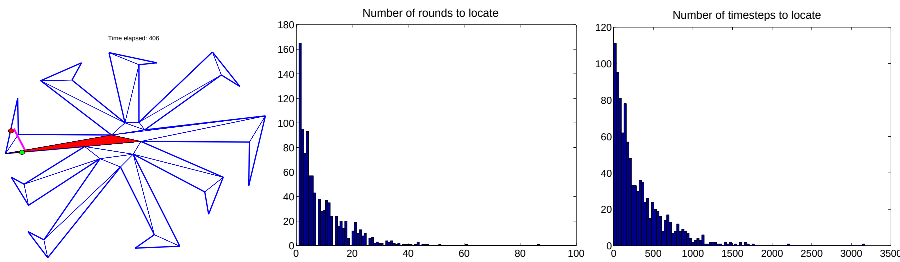

Fig. 4. **Left:** An instance of the simulator showing the triangulation of the environment as well as the hiding location of the evader. **Middle:** The histogram of the number of rounds required to locate evader in 1000 simulations. **Right:** The histogram of the time-steps required to locate evader in 1000 simulations.

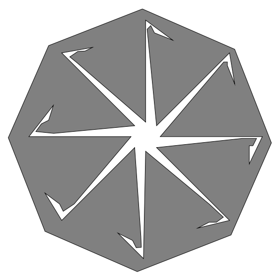

Fig. 5. For any randomized pursuer strategy, the expected time to capture the evader in this star with hooks is $O(n \cdot diam(P))$.

2.2 Lower bounds

One might suspect that the expected time to locate an evader can be improved using a more sophisticated strategy. Unfortunately, this is not possible: The polygon in Figure 5 is a $k-$star with hooks attached at the end of each spike (in the figure $k = 8$). The evader's strategy is to choose a hook at random and hide there until the end of the game. In order to locate the evader, the expected number of spikes searched by the pursuer is $\frac{k}{2}$ and it takes $diam(P)$ steps to travel from one spike to another. Since the number of vertices is a constant multiple of k, the time it takes to locate the evader is $\Omega(n \cdot diam(P))$. In fact, using the well-known technique due to Yao, this argument can be extended to show that the expected time to capture the evader for *any* randomized pursuer strategy is $\Omega(n \cdot diam(P))$ (see [8] for details).

We present the results of a simulation of the pursuer and the evader strategy for this environment in Figure 4.

3 Capturing the evader with two pursuers

In this section, we move on to the more challenging task of capturing the evader, defined as moving to the same point as the evader. We assume that there are two pursuers in the game. Later, we will show how to modify their strategy for the case of a single pursuer (at the expense of increasing the expected capture time). We make the following assumptions:

- The pursuers can communicate with each other at all times. [1]
- The evader has global visibility. It is unpredictable, i.e., no prior information about its strategy is available.
- Both pursuers have only line of sight visibility. One pursuer is at least as fast as the evader.
- The game is played in discrete time. In order to simplify presentation, we assume that the players move in turns: first the evader moves, followed by the pursuers.

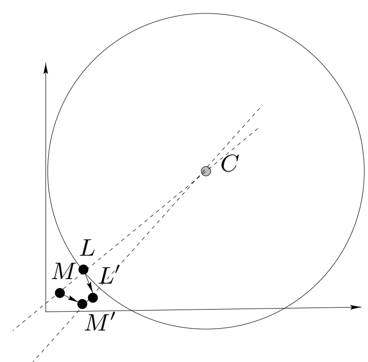

Fig. 6. Lion's strategy

The strategy of one of the pursuers is based on the solution to a problem known as the *lion and man* problem [12]. We present an extension of this strategy in the case of a (possibly non-convex) polygonal environment. One of the major difficulties for our pursuers is that the evader may not be visible at all times, in which case the lion's strategy is not well-defined. The second pursuer will use the strategy presented in the previous section to tackle this difficulty.

We start with a review of the lion's strategy.

[1] This assumption can be relaxed to line-of-sight communication, i.e. the pursuers can communicate only if they see each other.

3.1 The lion and man problem

The *lion and man* problem with discrete time in the nonnegative quadrant of the plane is attributed to David Gale [3]. Let the initial positions of the lion and man be $L_0 = (x_0, y_0)$ and $M_0 = (x'_0, y'_0)$, respectively. In each round, first the man moves to any point in the quadrant at distance at most 1 from his current position, and then the lion does the same. The lion wins if he moves to the current position of the man. The man wins if he can keep escaping for infinitely many rounds. In [12], Sgall proves that, when both $x'_0 < x_0$ and $y'_0 < y_0$, the lion always catches the man in a finite number of rounds. The number of moves required is bounded by a quadratic function in x_0, y_0 and the slope (or its inverse) of the line segment L_0M_0.

3.2 Lion's strategy

Let the initial positions of the lion and man be $L_0 = (x_0, y_0)$ and $M_0 = (x'_0, y'_0)$, respectively. In the beginning of the game, the lion finds a point C on the line M_0L_0 such that L_0 is inside the segment M_0C and the circle with center C, radius $|CL_0|$ and passing through L_0 intersects both axes. Among all possible such circles, it chooses the one whose center is closest to the origin. C remains fixed throughout the game.

Let L and M denote the current positions of the lion and the man respectively (see Figure 6). Let M' denote the point the man moves to, $|MM'| \leq 1$. If $|LM'| \leq 1$, the lion catches the man. Otherwise, it moves to a point L' on the line $M'C$ such that $|L'L| = 1$. There are two such points, it chooses the one closer to the man.

Definition 1. We will refer to this move as the *lion's move from L with respect to C and M'.*

The lion's move maintains the following:

Lemma 2 ([12]). *If the lion does not catch the man in the current move then*

(i) M' has both coordinates strictly smaller than C,
(ii) L' is inside the segment $M'C$, and
(iii) $|L'C|^2 \geq 1 + |LC|^2$.

Proof. See [12].

3.3 The strategy to capture the evader

Let $p_1(t), p_2(t)$ and $e(t)$ denote the locations of the pursuers and the evader, respectively, at time t. In the beginning of the game the two pursuers move together and search for the evader using the strategy described in the previous

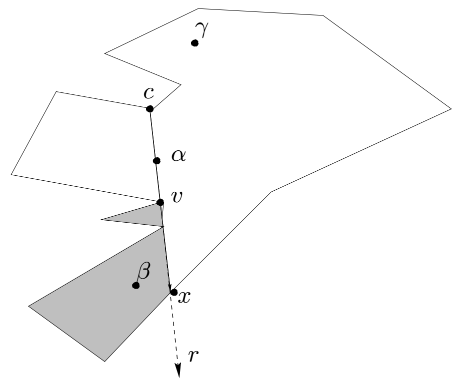

Fig. 7. Pocket with respect to c and v

section. Without loss of generality, we assume that the game starts at $t = 0$ where $p_1(0) = p_2(0) = o$ and $e(0)$ is visible from o. We will sometimes refer to point o as the *origin*. The origin will be fixed until the evader is captured. Let $d_1(t) = d(p_1(t), o)$, $d_2(t) = d(p_2(t), o)$, and $d_e(t) = d(e(t), o)$.

Definition 2. Suppose $e(t)$ is visible from $p_1(t)$ but $e(t+1)$ is not visible from $p_1(t)$. This means that the shortest path P from $p_1(t)$ to $e(t+1)$ is composed of at least two line segments (Property 1). The first vertex on the path from $p_1(t)$ to $e(t+1)$ is called a *pseudo-blocking vertex.*

Let r be the ray starting from a vertex c and passing from another vertex v that is not adjacent to c. In the sequel, c will be the center of the circle for the lion's move and v will be the pseudo-blocking vertex. Consider the first time the ray r leaves the polygon P after it passes through v and let x be the point on $r \cap P$ just before this happens (see Figure 7). The line segment vx splits the boundary of the polygon into two chains. The chain which does not contain the point c, together with the line segment vx defines a polygon. We will refer to this polygon as *the pocket with respect to* c *and* v. The line segment vx is referred to as the *entrance of the pocket.*

We will utilize the following properties of pockets:

Property 2. Let α be a point on the line segment cv and β be a point in the pocket with respect to c and v. The line segment αv is contained in the shortest path from α to β (Figure 7).

Property 3. Let R be a pocket with respect to c and v inside a polygon P. Any path from $\beta \in R$ to $\gamma \in P - R$ crosses the entrance of the pocket (Figure 7).

Looking ahead, let us describe how we will utilize these properties: Suppose pursuer p_1 is moving towards the evader and the evader disappears. Let v be the current pseudo-blocking vertex. If p_1 moves towards v, Property 2

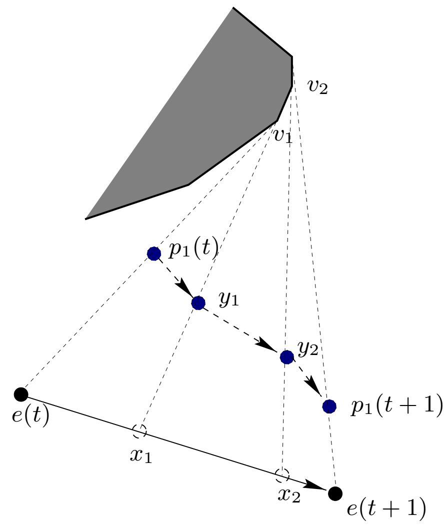

Fig. 8. The extended lion's move

implies that it is still moving on the shortest path from the evader to the origin. If the evader becomes visible before p_1 reaches v, Property 3 implies that it must cross the entrance of the pocket and p_1 can continue its strategy (described in the next section) as if the evader has not disappeared.

If the evader is not visible when p_1 arrives at v, then v becomes a *blocking vertex*. At this point, the second pursuer will enter the game.

Next, we present the details of the strategies of p_1 and p_2.

3.4 Strategy of Pursuer p_1

As stated earlier, we assume that pursuer p_1 is at least as fast as the evader. At time step t, p_1 moves according to the following strategy:

If the evader is visible, he performs a *extended lion's move* which is defined as follows: Let τ be the shortest path from $e(t)$ to $e(t+1)$. Without loss of generality, p_1 will pretend that the evader followed τ. As a point x moves from $e(t)$ to $e(t+1)$ along τ, the vertices on the shortest path from x to the origin o may change. However, the number of changes is at most n: The first vertex on the shortest path from x to o must be one of the vertices of the polygon. Since τ is the shortest path from $e(t)$ to $e(t+1)$, each vertex of the polygon can be this first vertex for at most one contiguous sub-path in τ. Let $x_1, \ldots, x_{k-1}$ correspond to the points on τ where such changes occur, we define $x_0 = e_t$ and $x_k = e(t+1)$. Let v_i be the first vertex on the shortest path from x_i to o. The extended lion's move consists of $k-1$ phases. During Phase i, $i = 1, \ldots, k$, pursuer p_1 performs the lion's move with respect to v_i and x_i (see Figure 8). Note that the time spent by the pursuer in Phase i is equal to the time spent by the evader in traveling from x_{i-1} from x_i.

If the evader was visible in the previous time step, but is not visible any more, let v be the pseudo-blocking vertex. Pursuer p_1 moves towards v until he reaches it. If the evader becomes visible before p_1 arrives at v, he continues with the lion's move. Otherwise, v becomes a blocking vertex.

If the evader is still not visible after p_1 reaches the blocking vertex, he waits for p_2 to report the location of the evader. Let R be the current pocket defined with respect to the blocking vertex and the current center. There are two possibilities.

1) The evader reveals itself to p_1. Then, by Property 3, this must happen before the evader crosses the entrance of R. In this case p_1 continues the game with the lion's move.

2) Pursuer p_2 finds the evader located at e. Let v' be the first vertex on the shortest path from v to e and R' be the pocket with respect to v and v'. In this case, v' becomes a pseudo-blocking vertex, R' becomes the new pocket and p_1 continues his strategy by moving towards v'.

3.5 Strategy of Pursuer p_2

The task of pursuer p_2 is to search for the evader when it is not visible to p_1. When the evader disappears from the sight of p_1, pursuer p_2 waits until p_1 reaches the blocking vertex. Afterwards, p_2 locates the evader using the strategy described in the previous section and reports the location of the evader to p_1.

3.6 Properties of Pursuer p_1's strategy

Lemma 3. *For all times t, pursuer p_1 maintains the following invariants until the evader is caught:*

(I1) $p_1(t)$ is on the shortest path from o to $e(t)$.

(I2) $d_1(t+1)^2 \geq d_1(t)^2 + \frac{1}{n}$ if $p_1(t) \neq p_1(t+1)$.

Proof of Invariant I1: We prove the invariant by induction. Assume that it holds at time t.

First consider the case where p_1 can see e at time t. Let the first vertex on the shortest path from $e(t)$ to o be u. It follows that $p(t)$ is in the line segment joining u to $e(t)$, since if $p(t)$ is between o and u on the shortest path, he would not be able to see $e(t)$.

Let x denote the evader's position at an arbitrary time in the time interval $[t, t+1)$. Suppose when the evader is at x, the first vertex on the shortest path from x to o changes from u to v. Note that p_1 can see the evader until this point. Then, the shortest path from x to o passing through u and the shortest path from x to o passing through v have the same length. This implies that u, v, and x have to be collinear. For otherwise, a shorter path from x to o can be found in the interior of the polygon formed by these two presumed shortest paths from x to o, which is a contradiction.

This implies that either u is an ancestor or a descendant of v in the shortest path tree rooted at o. If u is an ancestor, at the point x where the switch occurs, p_1 could either be on the segment vx in which case it can continue the lion's move in the next phase or p_1 is on the segment uv, in which case e will become invisible to p_1 after x. In this case, p_1 must be either moving towards a pseudo-blocking vertex or waiting at a blocking vertex. In both cases, the invariant is maintained by Property 2. If u is a descendant of v, then p_1 is already on the segment ux and hence on the segment vx. Hence it can continue the lion's move in the next phase. The invariant is therefore maintained as a corollary of Lemma 2.

Otherwise, if p_1 does not see the evader at time t, he must be either waiting at a blocking vertex or moving towards a pseudo-blocking vertex. In both cases, the invariant is maintained by Property 2.

Proof of Invariant I2:

If p_1 is moving towards a pseudo-blocking vertex, his distance to the origin is increasing by 1 and the invariant is maintained.

Next, we show that the extended lion's move maintains the invariant: Suppose the lion's move has $k \leq n$ phases and consider phase i of the extended lion's move where the evader moves from the point x_{i-1} to x_i. Suppose, during this phase the pursuer p_1 moved from point y_{i-1} to y_i (see Figure 8) and let v_i be the center of the circle for the lion's move during this phase.

Let $\alpha_i = d(o, y_i) - d(o, y_{i-1})$.

As a corollary of Lemma 2 we have

$d(y_i, o)^2 \geq d(y_{i-1}, o)^2 + \alpha_i^2$.

Summing up over all phases we get the total progress to be $\sum_{i=1}^{k} \alpha_i^2$.

This expression when subject to $\sum_{i=1}^{k} \alpha_i = 1$ is minimized when all $\alpha_1 = \ldots = \alpha_k = \frac{1}{k}$. Therefore we have $d_1(t+1)^2 \geq d_1(t)^2 + \frac{1}{k}$ which implies the invariant I2. □

The combined strategy of the two pursuers can be viewed as follows: Pursuer p_1 moves only when it knows the shortest path from the evader to the origin o. Performing the lion's move is equivalent to growing a disk inside the polygon whose center is at the origin o and passes through the current location of p_1. By invariant I1, the evader can never enter the disk. Further, the disk is still protected if p_1 does not move. Invariant I2 implies that, whenever p_1 moves, the disk monotonically grows and the evader is eventually squeezed between p_1 and the polygon boundary.

Pursuer p_2 moves only when p_1 does not know the evader's path to the origin. It locates the evader using the randomized strategy given in the previous section and reports its location to p_1 so that p_1, in turn, can keep growing the disk and eventually capture the evader.

3.7 Expected time to capture

Let T_1 be the time it takes the faster pursuer (who performs the lion's move) to travel the diameter of the polygon. By Invariant I2 (Lemma 3), this pursuer will capture the evader in nT_1^2 steps. However, in the meantime, the other pursuer may have to search for the evader. Each such search ends in time $T_2 \cdot n \cdot \log n$ with high probability, where T_2 is the time for pursuer p_2 to travel the diameter of the polygon. Moreover, once a vertex becomes a blocking vertex, it will never become a blocking vertex again (it will be included in the ball defined by the origin and p_1), therefore at most n such searches will be necessary. In conclusion, the expected time to capture the evader is $O(nT_1^2 + T_2 \cdot (n^2 \log n))$ with high probability.

3.8 Capturing the evader with a single pursuer

Suppose we have only pursuer p_1. In this case, instead of waiting for p_2 to find the evader, p_1 can guess the first vertex on the shortest path from the evader to his current location and move there.

Consider the shortest path tree T from the origin o to the vertices of the polygon. For each vertex v, let $l(v)$ be the number of leaves of the subtree $T(v)$ of T rooted at the vertex v. Then the probability that the pursuer's guess will be successful if he is located at v is at least $\frac{1}{l(v)}$. If the guess is correct and the evader is visible, the pursuer continues with the lion's move. However, in case of a wrong guess the evader may end up in an advantageous location and move towards the origin o, in which case the pursuer must restart the game. Further, if all the guesses are correct, no vertex can be a blocking vertex more than once. Continuing this way we can obtain a worst-case lowerbound on the probability of success. Unfortunately, this bound can be possibly exponentially small in the number of reflex vertices in the environment. However, the expected time to capture the evader is still finite for any simply connected environment and this strategy may still be practical for simple settings.

One might suspect that an analysis similar to the one in Section 2 can be applied to prove that the expected time to capture is polynomial. The reason such an analysis does not apply directly is that even if the pursuer and the evader are co-located in a leaf triangle, the capture game still continues and the evader can move to another triangle in the tree. Therefore the number of guesses may exceed the depth of the tree, resulting in a possibly exponential capture time. This poses an interesting trade-off between the pursuer's visibility and the capture time. If the pursuer can somehow track the evader at all times (perhaps using a satellite), then Lemma 3 implies that he could capture the evader in time $O(n \cdot diam(P)^2)$. If this is not possible though, he can either use a second pursuer for locating the evader and still capture it in polynomial time or simultaneously search and capture which results in a much longer capture time.

3.9 Polygonal rooms with a door

In [7], Lee et al. studied the following variant of the pursuit-evasion problem: The input is a pair (P, d) where P is the polygonal room the game is played in and d is a *door*, a point marked on the boundary of P. The goal is to devise a strategy for the pursuer to eventually see the evader, in such a way that the evader can not escape through the door. The authors presented a characterization of polygons where a single pursuer with very narrow visibility (represented by a single ray) can locate the evader before it reaches the door.

In a similar scenario, the two pursuer algorithm presented in Section 3 can be used to *capture* an evader before it exits through the door. The only modification necessary is the following: Initially, pursuer p_1 is located at the door d and waits until pursuer p_2 locates the evader. Afterwards, he continues with the lion's move with respect to d. This ensures that the evader can never enter the disk whose origin is d and passes through the current location of p_1. Therefore, the door is always protected until the evader is captured.

4 Conclusion and Future Work

In this paper, we studied the visibility-based pursuit evasion game and showed that using a randomized strategy a single pursuer can locate an unpredictable evader in any simply-connected polygonal environment. The evader may be arbitrarily faster than the pursuer and it may know the location of the pursuer at all times.

The randomized strategy has some desirable properties: First, as shown in [2], there are polygonal environments which require an arbitrary number of pursuers if they are restricted to deterministic strategies. Therefore on such environments, a randomized strategy is mandatory for locating the evader with a single pursuer. Moreover, even if the polygon is deterministically searchable by a single pursuer, it is known that some of these polygons require revisiting parts of the polygon $\Omega(n)$ times [2]. Hence, the expected time to capture with a randomized strategy is comparable to the time to capture with a deterministic strategy. However, the randomized strategies may be preferable to the deterministic strategies, as they do not require complicated data structures and costly preprocessing.

Second, the randomized strategy to locate the evader does not require an exact map of the environment: It is based on the dual graph of the triangulation, therefore it is insensitive to errors in the map of the environment. An interesting research direction is to incorporate the navigation strategies in [14] which require a minimal representation of the environment.

Another interesting extension is the case of non-polygonal environments. The randomized strategy can be used to locate the evader in non-polygonal, simply-connected environments. For example, this could be done by replacing the triangulation tree (Lemma 1) with the decomposition studied in [6].

We have also studied the more challenging problem of capturing the evader. For this problem, we presented a strategy for two pursuers (one of which is as fast as the evader) to capture the evader in expected time polynomial in the number of vertices and the diameter of the environment. The strategy can be modified for a single pursuer, however it is not clear whether the expected time to capture remains a polynomial in the number of vertices. We leave this as a future research direction.

References

1. B. Chazelle and D. Dobkin. Computational geometry. *Optimal convex decompositions, G. Toussaint (editor)*, pages 63–133, 1985.
2. L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani. A visibility-based pursuit-evasion problem. *International Journal of Computational Geometry and Applications*, 9(4/5):471–, 1999.
3. R. Guy. Unsolved problems in combinatorial games. In *Proc. Symp. Applied Mathematics*, pages 183–189, 1991.
4. J. P. Hespanha, G. J. Pappas, and M. Prandini. Greedy control for hybrid pursuit-evasion games. In *Proceedings of the European Control Conference*, pages 2621–2626, Porto, Portugal, September 2001.
5. M. Keil. Decomposing a polygon into simpler components. *SIAM J. Computing*, 14:799–817, 1985.
6. S. M. LaValle and J. Hinrichsen. Visibility-based pursuit-evasion: The case of curved environments. *IEEE Transactions on Robotics and Automation*, 17(2):196–201, Apr. 2001.
7. J.-H. Lee, S.-M. Park, and K.-Y. Chwa. Searching a polygonal room with one door by a 1-searcher. *International Journal of Computational Geometry and Applications*, 10(2):201–220, 2000.
8. R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
9. J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1998.
10. S.-M. Park, J.-H. Lee, and K.-Y. Chwa. Visibility-based pursuit-evasion in a polygonal region by a searcher. *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, 2076, 2001.
11. A. Sarmiento, R. Murrieta-Cid, and S. Hutchinson. An efficient strategy for rapidly finding an object in a polygonal world. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Optimal Systems (IROS)*, 2003.
12. J. Sgall. Solution of David Gale's lion and man problem. *Theoret. Comput. Sci.*, 259(1-2):663–670, 2001.
13. I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM Journal on Computing*, 21(5):863–888, 1992.
14. B. Tovar, S. M. LaValle, and R. Murrieta. Optimal navigation and object finding without geometric maps or localization. In *Proc. IEEE International Conference on Robotics and Automation*, 2003.
15. R. Vidal, O. Shakernia, J. Kim, D. Shim, and S. Sastry. Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation. *IEEE Transactions on Robotics and Automation*, 18:662–669, 2002.