Randomized Pursuit-Evasion in a Polygonal Environment

Volkan Isler, *Student Member, IEEE,* Sampath Kannan, and Sanjeev Khanna

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 to a problem called the "lion and man problem" [2] 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. We show how this strategy can be extended to obtain a strategy for (i) a polygonal room with a door, (ii) two pursuers who have only line-of-sight communication, and (iii) a single pursuer (at the expense of increased capture time).

Index Terms

Pursuit-evasion games, path planning, randomized algorithms

CONDITIONALLY ACCEPTED PAPER
Log Number: W04-228

Corresponding Author is Volkan Isler.
Email: isler@eecs.berkeley.edu
Tel: +1-510-642-7512
Fax: +1-510-642-1800
Mailing Address:
CITRIS, 475 HMMB, Mail Code 1764,
University of California,
Berkeley, CA, 94720, USA

Sampath Kannan
Department of Computer and Information Science,
3330 Walnut St. Levine Hall Philadelphia, PA,19104, USA
Email: kannan@cis.upenn.edu

Sanjeev Khanna
Department of Computer and Information Science,
3330 Walnut St. Levine Hall Philadelphia, PA,19104, USA
Email: sanjeev@cis.upenn.edu

Volkan Isler is with the Center for Information Technology Research in the Interest of Society (CITRIS) at University of California, Berkeley. Sampath Kannan and Sanjeev Khanna are with the Computer and Information Science Department of the University of Pennsylvania. V. Isler is supported in part by MURI DAAH-19-02-1-03-83. S. Kannan is supported in part by NSF Grant CCR0105337 and ARO Grant DAAD19-01-1-0473. S. Khanna is supported in part by an Alfred P. Sloan Research Fellowship and an NSF Career Award CCR-0093117.

Corresponding Author is Volkan Isler. Email: isler@eecs.berkeley.edu Tel: +1-510-642-7512 Fax: +1-510-642-1800 Mailing Address: CITRIS, 475 HMMB, Mail Code 1764, University of California, Berkeley, CA, USA

A preliminary version of this paper appeared in the Proceedings of the Workshop on Algorithmic Foundations of Robotics (WAFR 2004) [1]

Randomized Pursuit-Evasion in a Polygonal Environment

I. 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 developing pursuit strategies (which incorporate sensing limitations) to capture intelligent evaders contaminating a complex environment [3], [4], [5]. The main ingredient of a pursuit-evasion game is the presence of an adversarial evader who actively avoids capture. Due to this aspect, a pursuit strategy is usually different from a search strategy where the target's motion is independent of the pursuer's (e.g., [6], [7]). Obtaining such pursuit strategies is important for surveillance applications where we would like to locate, and perhaps, capture intruders who may be adversarial. Another application is a search-and-rescue operation where we would like to save a victim. In this setting, even though the victim is not adversarial, a pursuit strategy is still desirable as it guarantees a rescue regardless of the victim's actions.

To model the adversarial nature of the game, pursuit-evasion games are usually studied in a game theoretic framework [8], [9]. The conditions under which the pursuer can capture the evader are obtained by studying a Hamilton-Jacobi-Isaacs equation which brings together the system equations of the pursuer and the evader. This approach has the advantage of yielding a closed-form solution of the game. Unfortunately, as the environments get complicated, solving Hamilton-Jacobi-Isaacs equations become intractable. Therefore solutions of pursuit-evasion games in complex environments are usually algorithmic.

Perhaps the most well-understood game in this context is the *visibility-based pursuit-evasion game* where one or more pursuers try to locate an evader in a polygonal environment [10], [11], [12]. In this game, the evader is very powerful: it has unbounded speed and global visibility, meaning that it knows the location of the pursuers at all times. In [4], the authors study a similar game in a probabilistic framework (where the evader performs a random walk) and propose a greedy algorithm.

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 [12] in order to locate the evader with deterministic strategies. Here, n denotes the number of vertices of the polygon. 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 2). 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 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 known solution to a problem called the lion and man problem [2] which is reviewed in Section III-A. The same strategy can be used to capture the evader while protecting a door. This problem was introduced in [13] 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.

Organization of the paper

We start the paper with a motivating example for randomized strategies (Section I-A). We present preliminary concepts and definitions in Section I-B. In Section II, we address the problem of locating a fast, unpredictable evader with global visibility.

Next, in Section III, we address the task of capturing the evader in a simply-connected environment. For this problem we present a randomized strategy for two pursuers, who can communicate at all times, to quickly capture the evader. We show how this strategy can be modified for a single pursuer at the expense of increasing the capture time in Section IV-A. We also present extensions of the basic two-pursuer strategy for the case where the pursuers have limited communication (Section IV-B) and for a scenario where the polygonal room has a door through which the evader can escape (Section IV-C).

A. 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 [14].

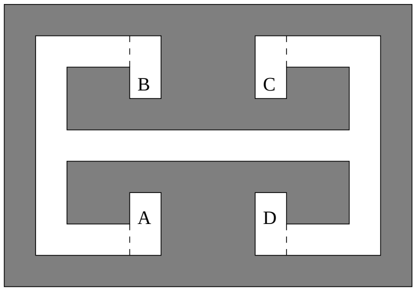


Fig. 1. A single pursuer can not capture an evader using deterministic strategies.

In this example, a single pursuer $\mathcal{P}$ can never locate the evader using a deterministic strategy: Let us distinguish four regions A, B, C and D as shown in the figure. Now suppose the pursuer has a deterministic strategy of visiting these regions in the order A, B, C, D. In this case, the evader $\mathcal{E}$ 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 while $\mathcal{P}$ is at C. If $\mathcal{P}$ visits the regions in a different order, it is easy to see that $\mathcal{E}$ can find a similar strategy to avoid $\mathcal{P}$. Therefore, in this polygon one pursuer can never locate the evader.

An alternative interpretation of this situation is the following. Suppose the polygon in Figure 1 is contaminated with many evaders executing all possible evader strategies. There is no deterministic pursuer strategy that guarantees that all the evaders will be caught; for any given deterministic pursuer strategy, there will be at least one evader which can avoid being located forever.

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

The probability of locating the evader can be made arbitrarily small by repeating the same strategy a few times. If k is the number of trials, the probability of missing in all k trials is $(\frac{3}{4})^k$ in this example which decreases exponentially with 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 $\frac{T}{p}$. By repeating the experiment roughly $\frac{1}{p} \log \frac{1}{p}$ times, we can show (using the Chernoff bound) that the pursuer has a high probability of locating the evader. For details of this analysis the reader is referred to [15].

B. Preliminaries

Let P be the input polygon including its interior and V be the set of vertices of P. The letter 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 $\cup_{v \in V} \pi(x, v)$ where $\pi(x, v)$ denotes the shortest path from x to v. A polygon is *simply-connected* if any simple closed curve inside the polygon can be shrunk to a point. In other words, a simply-connected polygon does not contain any “holes”. All the polygons considered in this paper are simply-connected.

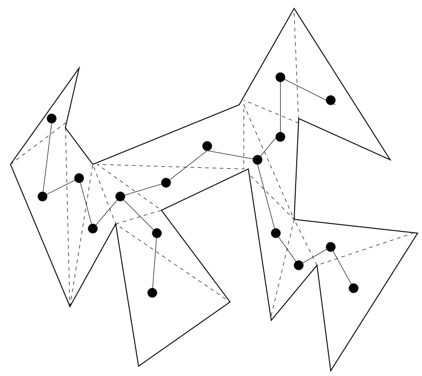


Fig. 2. Triangulation of a polygon and its dual tree.

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. In addition, the dual of the triangulation is a tree [16].

Game formulations

In this paper, we study two pursuit-evasion games with different objectives. Both games take place in a simply-connected polygon P which is known to all players.

The first game, which we call the *locating game*[1] is defined as follows.

It is played between an evader and a single pursuer. An *evader trajectory* is a continuous function $e : [0, \infty) \to P$ such that $e(t)$ denotes the evader’s position at time t. The *pursuer trajectory*, $p(t)$, is defined similarly. The pursuer moves with unit speed so that $\|\dot{p}\| = 1$. The *diameter* of the polygon P,

[1]The general version of this game is known as the visibility-based pursuit evasion game [12].

denoted $diam(P)$, is defined as $\max_{u,v \in P} d(u,v)$. Since the pursuer moves with unit speed, the diameter is also equal to the maximum amount of time it takes the pursuer to travel between two points in P. The evader can be arbitrarily faster than the pursuer but it must move continuously.

After the game starts, the players can observe their surroundings continuously. Further, at any given time t, the pursuer's location $p(t)$ is revealed to the evader. Therefore, the evader's strategy is a function of both the environment and the pursuer's trajectory. Since the pursuer does not observe the evader until the end of the game, a pursuer strategy is a function of only the environment. The pursuer wins the game if, in finite time t^*, he can reach a position such that $p(t^*)$ sees $e(t^*)$. The evader wins the game otherwise, i.e., if, for any given pursuer strategy, there exists a strategy for the evader to avoid being seen by a pursuer forever.

We assume that in both games, the evader knows the strategy of the pursuer(s) before the game starts. However, it does not have access to the outcome of the random coin tosses during the execution of the pursuer's strategy. The pursuer, on the other hand, knows nothing about the evader's strategy.

The second game is called the *capture game* and is defined as follows. Let $e(t)$ denote the evader's and $p_i(t)$ denote the i^{th} pursuer's trajectories as before. Instead of finding the evader, the pursuers win the capture game if in finite time t^*, they can reach a position such that there exists an i with $p_i(t^*) = e(t^*)$.

In this game, one of the pursuer's is as fast as the evader. Without loss of generality, we assume that $\|\dot{e}\| = \|\dot{p_1}\| = 1$. Similar to the locating game, the players observe their surroundings continuously and at any given time t, the pursuers' location $p_i(t)$ is revealed to the evader.

Unlike the previous game, we assume that the players move in discrete time intervals and in turns: the evader first, followed by the pursuers. It is easy to see that a pursuit strategy that captures the evader in this formulation can be modified to a pursuit strategy which guarantees that a pursuer p_i can reach a point within unit distance from the evader in a game where the players move continuously and simultaneously. It is interesting to note that for 25 years, the basic lion's strategy for the discrete-time formulation (Section III-A) was believed to be sufficient to capture the evader in the continuous formulation as well. However in 1952, Besicovitch showed that this is incorrect and that the evader can escape. In [17], Littlewood shows how this result can be generalized and that it is not possible to capture the evader in the continuous formulation (see also [18]).

II. Locating the evader

In this section, we study the locating game and show that for any simple polygon P, the pursuer can locate the evader in $O(n \cdot diam(P))$ expected time.

A. The pursuer strategy

The pursuer strategy to locate the evader utilizes the acyclic structure of the triangulation dual of a simply connected polygon. Intuitively, it relies on the the following observation. Let Δ be the triangle that contains the pursuer's current location (see Figure 3) and suppose Δ is non-leaf, i.e., has more than one neighbor. Since the triangulation dual is a tree, it is easy to see that Δ is a separator – removing it from the polygon results in smaller, disconnected polygons called components. This implies that the evader can not move from one component to another while the pursuer is located at Δ without revealing itself.

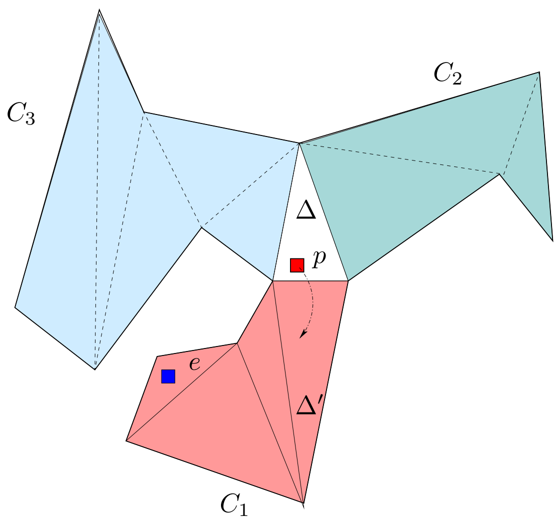


Fig. 3. Intuitive explanation of the pursuer strategy. Each component is displayed with a different color. While the pursuer is at Δ the evader can not move from one component to another. Further, when the pursuer moves to Δ', he can restrict the game to a smaller polygon.

The second observation is that the pursuer can not only prevent the evader from moving between components but also restrict the game to a smaller polygon. While the pursuer is at Δ, let C_1, C_2 and C_3 be the components such that the evader is located in C_1. Let Δ' be the neighbor of Δ in C_1. If the pursuer moves to Δ', he can restrict the game to C_1 as the evader will not be able to move to any triangle not contained in C_1.

Therefore, had the pursuer known the subtree that contains the evader, he could gradually move towards it – trapping the evader in smaller and smaller components. This process guarantees that the pursuer can enter the triangle which contains the evader and this clearly implies that the evader would be located.

The main difficulty in implementing the strategy above is that the pursuer does not know the component that contains the evader. In what follows, we show how the pursuer can utilize randomization to tackle this difficulty.

The pursuer strategy is divided into rounds. Each round lasts at most $diam(P)$ time units[2]. The pursuer will start the round at a leaf triangle and end the round at another leaf triangle[3].

Let Δ_0 be pursuer's triangle at the beginning of the round and T be the triangulation tree (see Figure 2) rooted at Δ_0. Within the round, the pursuer will visit triangles $\Delta_0, \Delta_1, \ldots$ where Δ_{i+1} is chosen among the children of Δ_i as follows. When the pursuer is at Δ_i, let $\Delta_i^1, \ldots, \Delta_i^k$ be the immediate

[2]Note that the pursuer has unit speed.

[3]When the game starts, if the pursuer is located at a non-leaf triangle, he moves to an arbitrary leaf before starting the first round.

children of Δ_i (see Figure 4). For each child Δ_i^j, let l_i^j denote the number of leaves of the subtree rooted at Δ_i^j. Let $L = \sum_{j=1}^{k} l_i^j$. The next triangle, Δ_{i+1} is chosen randomly among the children Δ_i^j according to the following distribution: the probability that Δ_i^j is chosen for Δ_{i+1} is $\frac{l_i^j}{L}$. After choosing the next triangle Δ_{i+1} the pursuer moves there. If he arrives at a leaf triangle, the round is over. Otherwise, the pursuer continues the round by picking one of the children of Δ_{i+1} as described above.

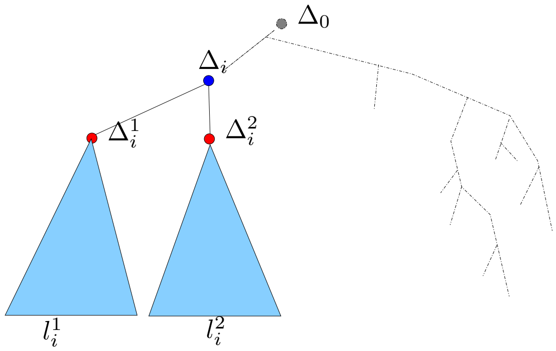


Fig. 4. Choosing the next triangle within a round (Lemma 1).

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

Lemma 1: At each round, if the pursuer follows the guessing strategy described above, he can locate the evader with probability at least $\frac{1}{N}$ where N is the total number of leaves of the triangulation tree T.

Proof: The lemma is proven by induction on the height of T. The basis, where the height 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 $p(\Delta)$ be the probability that the evader is located within a round, after the pursuer visits the triangle Δ. We inductively assume that the lemma is true for all trees of height less than or equal to i.

Given a triangulation tree of height $i+1$, the probability of success starting from the root Δ_0 is:

$$p(\Delta_0) \geq \min\left\{\frac{l_0^1}{N}p(\Delta_0^1), \ldots, \frac{l_0^k}{N}p(\Delta_0^k)\right\} \quad (1)$$

Note that for all j, the subtrees rooted at the immediate children Δ_0^j have height at most i, therefore by the inductive hypothesis we have $p(\Delta_0^j) \geq \frac{1}{l_0^j}$ for all j 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 2: In any simply connected polygonal 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.

The strategy for finding the evader is presented in Table I.

LocateTheEvader($\mathcal{T}$: a triangulation of the environment)

Go to an arbitrary leaf triangle *(initialization)*
while the evader is not found
  $\Delta_0 \leftarrow$ current triangle of the pursuer
  $T \leftarrow \mathcal{T}$ rooted at Δ_0
  $i \leftarrow 0$
  repeat
    $C_i \leftarrow \{\Delta_i^j : \Delta_i^j$ is a child of Δ_i in $T\}$
    $\Delta_{i+1} \leftarrow$ randomly chosen triangle from C_i where
        Δ_i^j is chosen with probability $\frac{l_i^j}{\sum_j l_i^j}$
    move from Δ_i to Δ_{i+1}
    $i \leftarrow i+1$
  until Δ_i is a leaf triangle

TABLE I

THE PURSUER'S STRATEGY FOR LOCATING THE EVADER. PLEASE REFER TO FIGURE 4 AND LEMMA 1 FOR THE NOTATION.

Remark 3: Any simply-connected polygon can be partitioned into a minimum number of disjoint convex polygons in polynomial time [19], [20]. The dual of such a partition will also 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 the dual of a minimum convex partition.

Remark 4 (Multiply-connected environments): The strategy presented in this section requires the triangulation dual to be a tree and therefore it does not hold for multiply-connected environments. To obtain an upper bound on the number of pursuers required for deterministic strategies, in [12] the following technique is presented: For an environment with h holes, $O(\sqrt{h})$ pursuers are utilized to reduce the environment into simply-connected components. An additional $O(\log n)$ pursuers are used to deterministically clear each simply-connected component. This yields an upper bound of $O(\sqrt{h} + \log n)$ pursuers. Using the same technique, we can establish an improved $O(\sqrt{h}+1)$ bound for randomized strategies. Using $O(\sqrt{h})$ pursuers, we partition the environment into K simply-connected polygons $P_1, \ldots, P_K$. After partitioning the environment, we use an extra pursuer to locate the evader. The strategy of this pursuer is as follows: Pick a simply-connected polygon $\tilde{P}$ among $P_1, \ldots, P_K$ uniformly at random. Execute one round of the randomized strategy on $\tilde{P}$. The probability of success is easily seen to be $\frac{1}{K} \cdot \frac{1}{n} \geq \frac{1}{n^2}$ and therefore the expected time to capture the evader is $n^2 \cdot diam(P)$.

B. 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

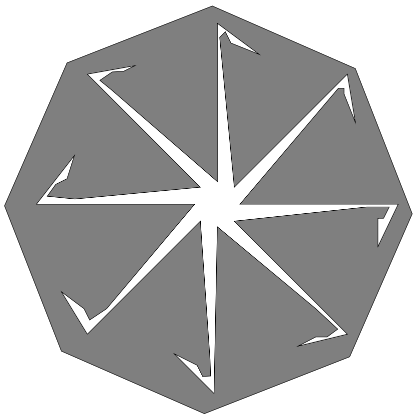


Fig. 5. For any randomized pursuer strategy, the expected time to capture the evader in this star with hooks is $O(n \cdot diam(P))$.

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 locate the evader for *any* randomized pursuer strategy is $\Omega(n \cdot diam(P))$ (see [15] for details).

We present the results of a simulation of the pursuer and the evader strategy for such an environment in Figure 6. For the simulation, the visibility-based pursuit evasion game was played 1000 times. The average number of rounds for locating the evader is 8.9. This is in agreement with Lemma 1 since the number of the leaves of the triangulation dual is 9.

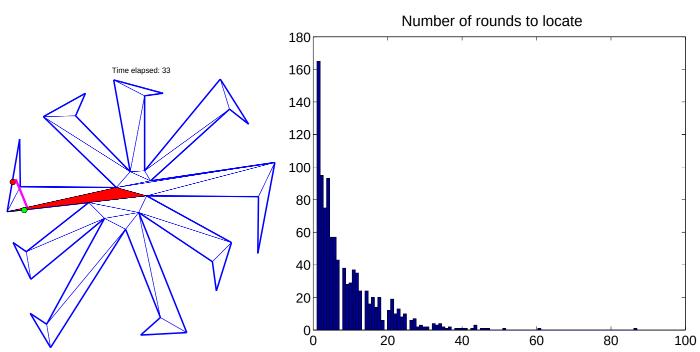


Fig. 6. **Left:** An instance of the simulator showing the triangulation of the environment as well as the hiding location of the evader. **Right:** The histogram of the number of rounds required to locate evader in 1000 simulations. The mean μ and the standard deviation σ of the number of rounds was $\mu = 8.8960$ and $\sigma = 9.0479$.

On the other hand, the randomized strategy may not be optimal for some environments. The simplest example of such an environment is a star-shaped polygon such as the one shown in Figure 7. In this environment, the optimal strategy is to go to a point (e.g., x in the figure) from where the entire polygon (hence, the evader) will be visible.

III. 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 start by presenting a pursuit strategy for two pursuers (who can communicate at all times) to capture the evader. Later, we will show how to modify this strategy to obtain a strategy for i) two pursuers who have only line-of-sight communication ii) a single pursuer (at the expense of increasing the expected capture time).

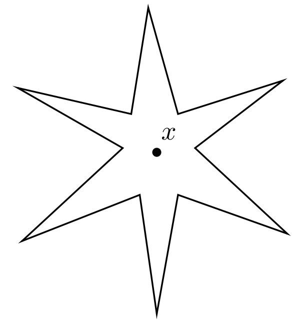


Fig. 7. A star-shaped polygon. The optimal strategy is to go to a point (e.g., x in the figure) from where the entire polygon is visible.

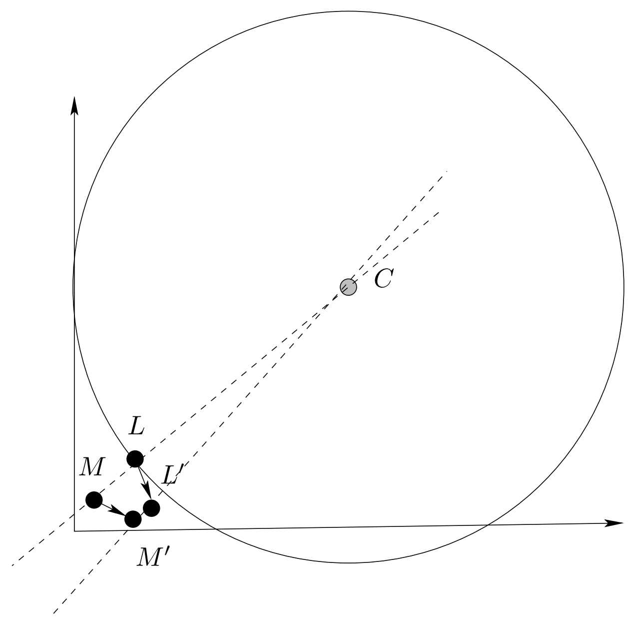


Fig. 8. Lion's strategy

The strategy of one of the pursuers is based on the solution to a problem known as the *lion and man* problem [2]. 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.

A. The lion and man problem

The *lion and man* problem with discrete time in the non-negative quadrant of the plane is attributed to David Gale [21]. 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 [2], 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 (in remaining cases, the man wins the game). 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.

B. 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 8). 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 5: We will refer to this move as the *lion's move from L with respect to C and M'* (Figure 8).
The lion's move maintains the following:

Lemma 6 ([2]): 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 [2]. ■

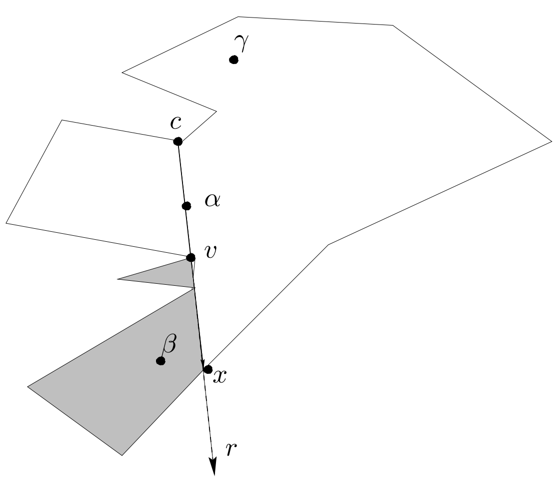


Fig. 9. Pocket with respect to c and v

C. 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 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 7: 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 through 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 9). 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 9).

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 9).

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 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.

D. 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 an *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 $x_k = e(t+1)$.

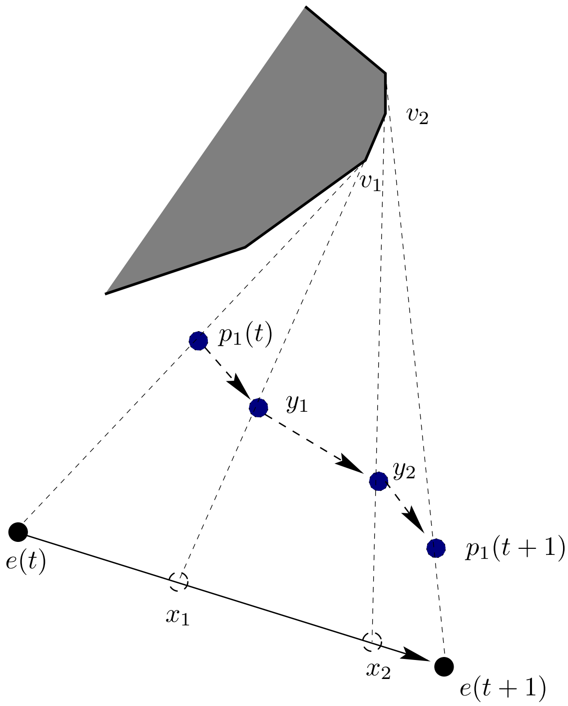


Fig. 10. The extended lion's move

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 10). 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 while before the evader is crossing 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'.

An illustration of different modes of the pursuer's strategy is presented in Figure 11.

E. 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.

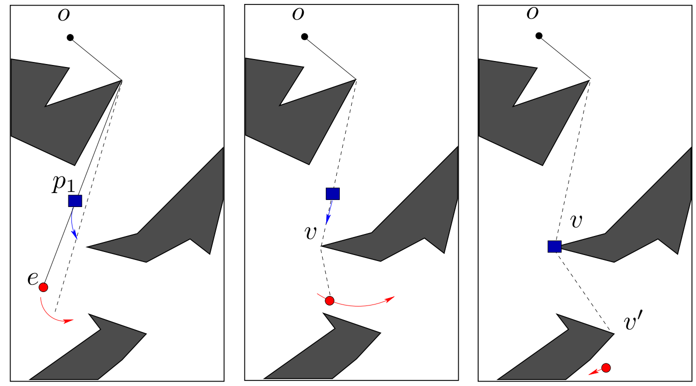


Fig. 11. A summary of pursuer p_1's strategy: **Left:** The evader is visible to p_1 who proceeds with the extended lion's move. **Middle:** The evader disappears for the first time. It is known to be behind the pseudo-blocking vertex v. Pursuer p_1 moves towards v. **Right:** Pursuer p_1 can not see the evader after arriving at v. The vertex v becomes a blocking vertex and p_1 sends out p_2 to search for the evader. After finding the evader, p_2 will report v', the new pseudo-blocking vertex.

F. Properties of Pursuer p_1's strategy

Lemma 8: 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 6.

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 10) and let v_i be the center of the circle for the lion's move during this phase.

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

As a corollary of Lemma 6 we have

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

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

This expression when subject to $\sum_{i=1}^{k} \omega_i = 1$ is minimized when all $\omega_1 = \ldots = \omega_k = \frac{1}{k}$. 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.

G. Expected time to capture

Let $T_1 = diam(P)$ be the time it takes pursuer 1 (who performs the lion's move) to travel the diameter of the polygon. By Invariant I2 (Lemma 8), this pursuer will capture the evader in nT_1^2 steps. However, in the meantime, pursuer 2 may have to search for the evader.

The number of searches is bounded by the number of vertices. This is because, once a vertex becomes a blocking vertex, it can never become a blocking vertex again. Next, we bound the length of each search. Recall that the probability of capturing the evader within a round is at least $\frac{1}{n}$ (Lemma 1). Using the inequality $(1+x) \leq e^x$, it can be easily shown that after $2n \ln n$ rounds, the probability of not finding the evader is at most $\frac{1}{n^2}$. Using the union bound, the probability of failure in any of the n searches is bounded by $n \cdot \frac{1}{n^2} = \frac{1}{n}$. Therefore with probability $1 - \frac{1}{n}$, all n searches finish in total time $T_2 \cdot 2 \cdot n^2 \cdot \ln n$ with high probability, where T_2 is the time for pursuer p_2 to travel the diameter of the polygon.

In conclusion, the expected time to capture the evader is $O(nT_1^2 + T_2 \cdot (n^2 \ln n))$ with probability arbitrarily close to one.

Our main result is summarized by the following theorem.

Theorem 9: In any simply-connected polygon, two pursuers p_1 and p_2 can capture an evader (whose speed is bounded by the speed of p_1) with probability arbitrarily close to one.

IV. EXTENSIONS OF THE TWO-PURSUER STRATEGY

In this section, we present three extensions of the two-pursuer strategy presented in the previous section. In Section IV-A, we show how a single pursuer can implement the same strategy at the expense of increased capture time. In Section IV-B, we show how the global communication requirement can be relaxed. Finally, in Section IV-C, we show that two pursuers can capture the evader even if the polygon has a door through which the evader can escape and win the game.

A. 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 II 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 8 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.

B. Relaxing the global communication requirement

When pursuer 1 arrives at a blocking vertex v, suppose the evader is not visible from v. Therefore pursuer 2 starts searching for the evader and finds it at time t.

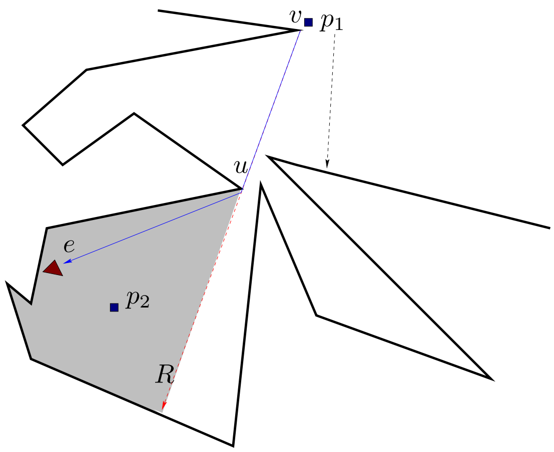


Fig. 12. Pursuer p_1 is waiting at the current blocking vertex v. Pursuer p_2 finds the evader in the pocket with respect to v and u (the shaded region). The vertex u will become the new pseudo-blocking vertex.

In the previous section, we assumed that the two pursuers can communicate all the time. Hence, after finding the evader at time t, pursuer 2 can compute the first vertex u on the shortest path from $p_1(t)$ to $e(t)$ and report u to pursuer 1. Afterwards, u becomes the pseudo-blocking vertex and pursuer 1 starts moving toward u (Figure 12).

The two pursuers can implement this strategy even if they have only line-of-sight communication. That is, the pursuers can communicate if and only if they can see each other. The only modification required in the strategy is the following.

After finding the evader and computing the pseudo-blocking vertex u at time t, if pursuer 1 and pursuer 2 can not see each other, pursuer 2 can not report the vertex u (see Figure 12). In this case, pursuer 2 starts moving towards pursuer 1 along the shortest path from $p_2(t)$ to $p_1(t)$. Let R be the pocket with respect to the current blocking vertex v and the pseudo-blocking vertex u (the shaded area in Figure 12). Since pursuer 1 will eventually see pursuer 2, there are two possibilities based on whether the evader becomes visible to pursuer 1 in the meantime.

If pursuer 1 sees the evader before seeing pursuer 2, this means that the evader is leaving R (Property 3). Further, at this time, pursuer 1 will be on the evader's shortest to the origin by Property 2. Therefore, in this case, pursuer 1 proceeds with the extended lion's move – maintaining both invariants $I1$ and $I2$.

Otherwise, if pursuer 1 sees pursuer 2 first, the evader must be in the pocket with respect to v and u. In this case, pursuer 2 reports the new pseudo-blocking vertex u and pursuer 1 proceeds as before by moving towards u. Note that pursuer 2 reports only the pseudo-blocking vertex, not the precise location of the evader.

C. *Polygonal rooms with a door*

In [13], 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 III 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.

V. 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 [12], 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 [12]. In such polygons, 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 and as long as the structure of the triangulation dual is preserved, it is insensitive to errors in the map of the environment. An interesting research direction is to incorporate the navigation strategies in [7] 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 [22]. Another alternative is to use the medial axis (c.f. [16] for a definition and related properties) of a simply-connected polygon instead of the triangulation tree.

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] V. Isler, S. Kannan, and S. Khanna, "Locating and capturing an evader in a polygonal environment," in *Proc. of 6th Workshop on Algorithmic Problems in Robotics (WAFR'04)*, 2004, pp. 351–367.

[2] J. Sgall, "Solution of David Gale's lion and man problem," *Theoret. Comput. Sci.*, vol. 259, no. 1-2, pp. 663–670, 2001.

[3] J. Hespanha, H. J. Kim, and S. Sastry, "Multiple-agent probabilistic pursuit-evasion games," in *In Proc. of the 38th Conf. on Decision and Contr*, 1999, pp. 2432–2437.

[4] 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*, vol. 18, pp. 662–669, 2002.

[5] J. P. Hespanha, G. J. Pappas, and M. Prandini, "Greedy control for hybrid pursuit-evasion games," in *Proceedings of the European Control Conference*, Porto, Portugal, September 2001, pp. 2621–2626.

[6] 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, pp. 1153–1158.

[7] 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, pp. 464–470.

[8] R. Isaacs, *Differential Games*. Dover, 1965.

[9] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. SIAM, 1998.

[10] I. Suzuki and M. Yamashita, "Searching for a mobile intruder in a polygonal region," *SIAM Journal on Computing*, vol. 21, no. 5, pp. 863–888, 1992.

[11] 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)*, vol. 2076, pp. 456–468, 2001. [Online]. Available: citeseer.nj.nec.com/park01visibilitybased.html

[12] 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*, vol. 9, no. 4/5, pp. 471–, 1999. [Online]. Available: citeseer.nj.nec.com/guibas96visibilitybased.html

[13] 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*, vol. 10, no. 2, pp. 201–220, 2000. [Online]. Available: citeseer.nj.nec.com/lee00searching.html

[14] S. M. LaValle, D. Lin, L. J. Guibas, J.-C. Latombe, and R. Motwani, "Finding an unpredictable target in a workspace with obstacles," in *In Proc. IEEE International Conference on Robotics and Automation*, 1997, pp. 737–742.

[15] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge University Press, 1995.

[16] J. O'Rourke, *Computational Geometry in C*. Cambridge University Press, 1998.

[17] J. E. Littlewood, *A Mathematician's Miscellany*. London: Methuen & Co. Ltd., 1953.

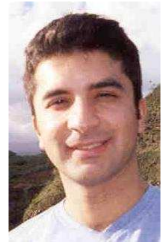
[18] L. Alonso, A. S. Goldstein, and E. M. Reingold, ""Lion and man": upper and lower bounds," *ORSA J. Comput.*, vol. 4, no. 4, pp. 447–452, 1992.

[19] M. Keil, "Decomposing a polygon into simpler components," *SIAM J. Computing*, vol. 14, pp. 799–817, 1985.

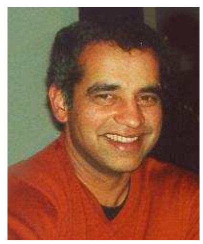
[20] B. Chazelle and D. Dobkin, "Computational geometry," *Optimal convex decompositions, G. Toussaint (editor)*, pp. 63–133, 1985.

[21] R. Guy, "Unsolved problems in combinatorial games," in *Proc. Symp. Applied Mathematics*, 1991, pp. 183–189.

[22] S. M. LaValle and J. Hinrichsen, "Visibility-based pursuit-evasion: The case of curved environments," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, pp. 196–201, Apr. 2001.



Volkan Isler Volkan Isler is a post-doctoral researcher at the Center for Information Technology Research in the Interest of Society (CITRIS) at the University of California, Berkeley. He received his MSE and PhD degrees in Computer and Information Science from the University of Pennsylvania and BS degree in Computer Engineering from Bogazici University in Istanbul, Turkey. His research interests are in robotics (pursuit-evasion, exploration, motion planning), sensor-networks (deployment, target tracking and localization) and computer vision (tele-immersion, model reconstruction and segmentation).



Sampath Kannan Sampath Kannan is a professor at the Computer and Information Science Department of the University of Pennsylvania. He obtained his MS degree from Princeton University (1984) and PhD degree from University of California, Berkeley (1989) – both in Computer Science. His research interests are algorithms, computational biology, and formal methods for software reliability. He is currently an editor of ACM Transactions on Algorithms.

Sanjeev Khanna Sanjeev Khanna is an associate professor in the Computer and Information Science Department of the University of Pennsylvania. He received a PhD in Computer Science from Stanford University and a BS degree in Computer Science from Birla Institute of Technology, India. His research interests are in approximation algorithms, combinatorial optimization, computational biology, and databases.