Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 Keller Hall
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 15-013

A Pursuit-Evasion Toolkit

Narges Noori, Andrew Beveridge, Volkan Isler

July 24, 2015

Revised

A Pursuit-Evasion Toolkit

Narges Noori, Andrew Beveridge and Volkan Isler

***Abstract*— This tutorial contains tools and techniques for designing pursuit and evasion strategies. The material targets a diverse audience including STEM educators as well as robotics researchers interested in applications of pursuit-evasion games.**

We start with a simple "lion and man" game in a square environment which should be accessible to anyone with a high-school level background on geometry and trigonometry. We then visit various versions of this game with increasing complexity. Rather than surveying specific results for specific environments, the tutorial highlights broadly applicable techniques and strategies. It also includes exercises for STEM educators as well as open problems for robotics researchers.

I. INTRODUCTION

A pursuit-evasion game takes place between two players. The pursuer is charged with capturing the evader while the evader tries to avoid getting caught. Many robotics applications such as search, tracking and surveillance can be modeled as pursuit-evasion games. Equally importantly, these games can be modeled as fun mathematics problems to inspire new-comers to the field of robotics. The authors of the paper have witnessed this first-hand during summer Research Experiences for Undergraduates (REU) programs at the Institute for Mathematics and its Applications (IMA), located at the University of Minnesota. The subject is accessible, with many open problems that require creativity, insight and strong algorithmic thinking. Our summer students digested the basics of the field and developed results that evolved into research publications [1], [2].

The purpose of this article is to provide a "toolkit" so as to make pursuit-evasion games accessible to a broader audience. We focus on a classical game known as the *lion and man game*, where the lion pursues the man, moving with equal speed. Rather than a traditional survey of literature on the lion and man game, the paper is organized in a tutorial fashion. We start from simple motivating examples whose solutions should be accessible to anyone with a high-school level background on geometry and trigonometry. Our journey takes us to open variants such as pursuit-evasion on surfaces. Along the way, we introduce tactics which can be used as building blocks in different settings. The material in the earlier sections of the paper provides a starting point for STEM educators looking for an engaging robotics problem accessible to high-school and undergraduate students. Simultaneously, we provide an introduction for researchers who would like to tackle one of the most challenging path planning problems. We conclude with open problems for the researchers and exercises to engage students. Let us dive right in!

A. Overview

The original version of the lion and man game takes place in a circular arena. Intuitively, the lion should win the game: if it moves directly toward the man, the man has to step back in the same direction to maintain the seperation between the players. He can not do this forever, since he will eventually hit the boundary. It turns out that the analysis of this simple "greedy" strategy is not straightforward since the turn angle can be arbitrarily small. In fact, when time is continuous and capture requires colocation, man can avoid capture indefinitely by following a gently spiraling path [3]. To avoid this pathology, robotics researchers focus on the turn-based version of the game, where the players move alternately. We assume that the players have the same maximum step-size; we employ a unit step size, which is both standard and convenient. The pursuer captures the evader when their distance is within than a fixed *capture radius* r. We encounter $r = 0$ and $r > 0$ with equal frequency in the literature, and we will consider both variants below. Current research in robotics focuses on games which take place in more complex environments (e.g., non-convex polygons, environments with polygonal obstacles) and sometimes consider players subject to sensing limitations.

We consider the full-visibility version of the game, where the players know the environment and one another's locations at all times. In order to capture the evader, a general pursuit strategy consists of two main phases. First, the evader is expelled from a subset of the environment, and this subset is protected thereafter, meaning that the evader cannot re-contaminate it without being caught. Second, the protected subset is gradually grown until the whole environment is cleared and the evader is captured. These two phases are referred to as **guarding** and **making progress** respectively.

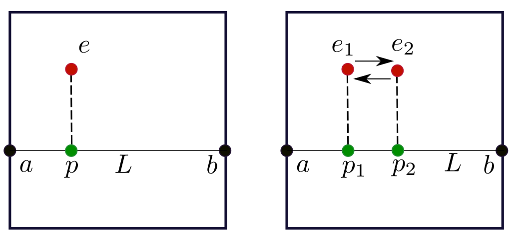
As a demonstrative example, suppose that this turn-based game is played in a square region; the pursuer can observe the exact location of the evader, and the capture radius is zero $r = 0$. A first intuitive idea for guarding is to locate the pursuer on the line segment L between two arbitrary boundary endpoints, and prevent the evader from crossing it. If we can also push L towards the evader the progress goal would be achieved as well.

The pursuer can guard L by positioning itself on the *vertical projection* of the evader onto L (Figure 1(a)). The vertical projection has the property that it is closer to all

Narges Noori and Volkan Isler are with the Department of Computer Science and Engineering, University of Minnesota. Andrew Beveridge is with the Department of Mathematics, Statistics and Computer Science at Macalester College. Emails: {noori, isler}@cs.umn.edu, abeverid@macalester.edu. This work is supported in part by NSF grants #1111638 and #0917676.

the points along L than the evader itself. As a result, if the evader tries to cross L, the pursuer (which is on the evader's projection) will capture it during its next move.

Although the vertical projection idea succeeds in restricting the evader to one side of L for the rest of the game, the pursuer fails to achieve its second goal: shrinking the evader's region. Indeed, suppose that the evader takes a full unit step parallel to L, moving from e_1 to e_2 with $|e_1 e_2| = 1$ (Figure 1(b)). Consequently, the evader's vertical projections onto L, denoted by p_1 and p_2 respectively, satisfy $|p_1 p_2| = 1$. Therefore, the pursuer exhausts its movement budget just to keep up with the evader. Now, if the evader reverses its direction after each step, the pursuer is forced to move between the two fixed points p_1 and p_2. Consequently, the pursuer cannot move L towards the evader, and the evader can escape forever.



(a) (b)

Fig. 1. **(a)** The pursuer can prevent the evader from crossing L by positioning itself on the vertical projection of the evader. **(b)** Since $e_1 e_2$ is parallel to L and $|e_1 e_2| = 1$, we have $|p_1 p_2| = 1$. Thus, the pursuer is stuck on L between the two points p_1 and p_2.

There are three approaches to tackle the issue above and achieve progress.

Approach 1. The Lion's Strategy: In the first approach, known as the **lion's strategy**, the pursuer can simultaneously guard and make progress. In this strategy, the pursuer (lion) starts at an arbitrary center $p^0 = c$. In round $t \geq 1$, the pursuer makes a **lion's move**, meaning that it moves from p^{t-1} to the point $p^t \in B(p^{t-1}, 1)$ on the line segment connecting c to e^t such that p^t is closest to the evader (Figure 2). As discussed in Section III, eventually the evader will be squeezed between the pursuer and the boundary which results in capture.

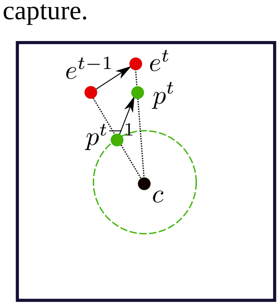
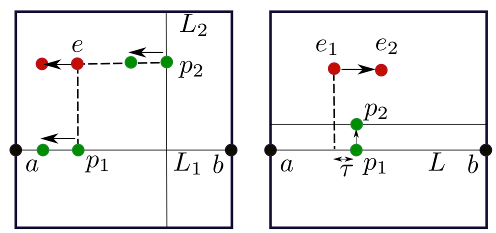


Fig. 2. Lion's strategy in a square region.

Approach 2. Multiple Guards: In the second approach, we add a pursuer to guard another line segment. In our square example, suppose that we have two pursuers, each of which guards a line by positioning itself on the evader projection. The first pursuer p_1 guards a horizontal line segment L_1, while the second pursuer p_2 guards the vertical line segment L_2 (Figure 3(a)). This traps the evader in one quadrant of the square. If the evader moves horizontally, then p_1 mimics this move while p_2 makes one unit of progress by advancing its guarded line. Likewise, if the evader moves vertically, then p_2 keeps pace while p_1 advances its guarded line. More generally, the Pythagorean theorem shows that at least one of the pursuers can advance its guarded line by $\sqrt{2}/2$ units in each turn. This means that the evader will be cornered by the pursuers after a finite number of moves, after which one of the pursuers has a capture move.

In both of the approaches above, capture is guaranteed for zero capture radius. Pursuit games with capture radius $r > 0$ are also common in the literature. Our third approach takes advantage of the non-zero capture radius to overcome the stalemate in Figure 1(b).



(a) (b)

Fig. 3. **(a)** Two pursuers are guarding L_1, L_2 by staying on the projection of the evader. One of them can make progress after each move. **(b)** A single pursuer can push L towards the evader if it stays behind the projection of the evader.

Approach 3. The Rook's Strategy: As mentioned above, this approach requires a non-zero capture radius. To build intuition, consider a chess endgame for a black rook and black king versus a solitary white king. Before reading on, we encourage our readers to develop a strategy for the black king to win the game.

The **rook's strategy** works as follows: One row at a time, the black rook reduces the area available to the white king. This is achieved with support of the black king who trails the projection of the white king onto the row guarded by the rook. Specifically, suppose the black rook is at row i, column 1, which guards row i. The white king is at row $i' < i$ and column j, and the black king is at row $i + 1$, column $j - 1$. From this position, the black king trails the white king until the column separation of the black rook and black king is at least two (Figure 4(a)). After achieving this separation, the black player can make progress as follows. If the white king moves back on to column $j - 1$, then the black player can move the rook to row $i - 1$ and push the white king further down (Figure 4(b))[1]. If the white king moves away

[1]This is because the white king cannot move to row i since the reachable cells are being guarded by the black king. Furthermore, in the extreme case where $i' = i - 1$, the white king cannot stay in row i' because of the two column separation between the black king and the black rook.

from the black king, then the black king keeps trailing him. Eventually, the white king is trapped in a single row, where black can checkmate.

In rook's strategy, the pursuer guards a horizontal line which is analogous to the row guarded by the rook. The pursuer plays the role of the black king and trails the projection. Whenever the evader moves "back" toward the projection, the pursuer can make progress by pushing the line forward. Eventually, the evader will be squeezed between the guarded path and the boundary, where it will be caught. In Section VI, we show how to generalize the rook's strategy to general environments.

In the rest of the paper, we present various techniques to guard a frontier as well as strategies to make progress by expanding the frontier towards the evader. In Section III we overview an analysis of the lion's strategy as well as an extension of it. In Section IV and Section V we introduce general projection mappings and their applications in designing capture strategies. In Section VI we elaborate on the rook's strategy and its applications for capturing the evader. We mostly focus on the full-visibility variants of the lion-and-man game where the assumption is that the players know one another's location at all times. Designing strategies for limited visibility models is an active research problem. We will study a limited-visibility version in Section VI as an application of the rook's strategy. In Section VII we present a set of exercises that can be used as handouts to encourage and warm-up the students. Finally, we conclude the paper with some open problems in Section VIII.

II. Notation

In this section, we present the notation used throughout the paper. The game environment is denoted by $\mathcal{S}$ with boundary $\partial\mathcal{S}$. We refer to the subset of $\mathcal{S}$ that the evader cannot enter without being captured as the **cleared** region. The remaining part of $\mathcal{S}$ is referred to as the **contaminated** region. We refer to a shortest path in $\mathcal{S}$ between points x and y by $\Pi(x, y)$. The shortest distance between x and y, i.e. the length of $\Pi(x, y)$, is denoted by $d(x, y)$. When we require that a distance is measured within a subset, such as to $Q \subseteq \mathcal{S}$, we write $d_Q(x, y)$. We use $B(x, r) = \{y \in \mathcal{S} \mid d(x, y) \leq r\}$ to denote the ball of radius r centered at x.

III. Lion's Strategy

The lineage of the lion and man game traces back to Rado's classic version from the 1930s [3]. This game takes place in a circular arena, and both the lion and the man have equal speed. Lion wins if it becomes colocated wtih the man in finite time. The lion's strategy of staying on the man's radius was generally accepted in folklore [3]. The capture time of this turn-based strategy is $O(R^2)$ where R is the radius of the environment. Sgall [4] uses a similar strategy to show finite time capture when the game takes place in the non-negative quadrant of the plane. Alonso et al. [5] proposed a more sophisticated strategy which guarantees capture in $O(R \log \frac{R}{r})$ steps where r is the capture radius.

Simple trigonometric arguments can be used to show that the lion's strategy captures the man. The proof exhibits the two essential ingredients for verifying that a pursuit strategy succeeds in finite time. First, we establish an **invariant** that the pursuer(s) maintain throughout the game. For lion's strategy, this invariant is that p was located on the radius between the center and the evader. Second, we need a **measure of progress** to show that the game ends in finite time. Let d and d' denote the distance between the center c and the lion before and after a move respectively (Figure 5(b)). Let α be the angle between ce_1 and p_1p_2. We have $d'^2 = (d + \cos\alpha)^2 + \sin^2\alpha = d^2 + 2d\cos\alpha + 1$ (Note that $p_1p_2 = 1$). We first show that there exists a point p_2 on ce_2 such that $\alpha \leq \frac{\pi}{2}$. This is because $e_1e_2 \leq 1$, and hence $p_1q \leq 1$ where q is a point on ce_2 such that qp_1 is prependicular to ce_1 (Figure 5(a)). As a result of $\alpha \leq \frac{\pi}{2}$ we have $d'^2 \geq d^2 + 1$. When coupled with the invariant, this guarantees capture after at most R^2 rounds.

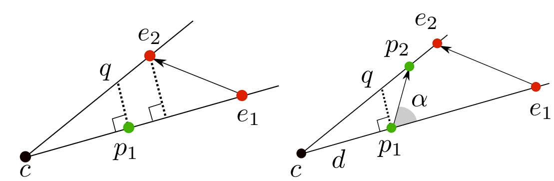


Fig. 5. In lion's strategy, the pursuer can increase its distance from the center by at least $d' - d \geq \frac{1}{2R}$. This is because: **(a)** The length of p_1q is less than one. As a result, there exists a point p_2 on ce_2 which is closer to e_2 than q, and moreover is at distance one from p_1. **(b)** In fact, $\alpha < \frac{\pi}{2}$.

Isler et al. [6] adapted lion's strategy for pursuit in a simply connected polygon P. First, the pursuer starts at point c in the polygon, which is typically a boundary vertex. Thereafter, the pursuer always moves onto the shortest path between c and the evader, getting as close to e as possible. Note that this shortest path could interact with the boundary of the polygon, in which case it will be a piecewise linear path (Figure 6). The extended lion's strategy uses the same invariant (being on the shortest path between c and e) and the same measure of progress (increasing $d(c, p)$ at a constant rate) as lion's strategy. For a polygon P with n vertices and diameter $\text{diam}(P) = \max_{u,v\in P} d(u, v)$, Isler et al. [6] proved that the capture time is $O(n \cdot \text{diam}(P)^2)$. Recently, Beveridge and Cai [7] gave a streamlined analysis that improves the capture time bound to $O(\text{diam}(P)^2)$.

IV. Guarding Shortest Paths

In this section, we begin our exploration of environments with obstacles. Obstacles create an advantage for the evader. For example, a single pursuer cannot catch an evader when there is one large obstacle in the environment. Indeed, the evader can start from a point on the boundary of the obstacle and then loop around the obstacle, moving away from the pursuer thereafter. In this section, we describe how a shortest path can be guarded by a pursuer even in the presence of obstacles. This capability turns out to be a powerful subrou-

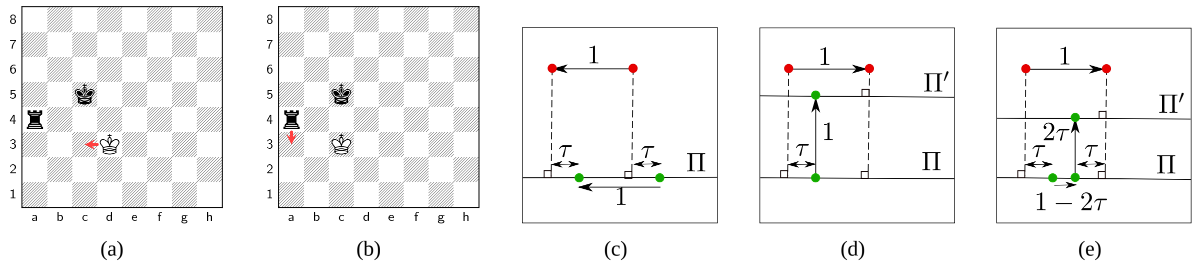


Fig. 4. The rook-and-king chess endgame is shown in (a) and (b). **(a)** The black rook is one row above the white king. The black king is (at least) two columns to the right and one row above the black rook, and just to the left of the white king. **(b)** After the white king moves leftward, the black rook makes progress. Illustration of the rook strategy when $\tau < 1/2$ is shown in (c), (d) and (e). **(c)** If the evader moves to the left, the pursuer moves in the same direction for one unit. **(d)** If the evader moves to the right, the pursuer pushes Π to Π'. When $1 - \tau \leq r$, the distance between Π and Π' is one unit. **(e)** When $1 - \tau > r$, the distance between Π and Π' is 2τ. Here notice that since $-\tau \geq -r$ and $1 - \tau > r$ we have $1 - 2\tau > 0$.

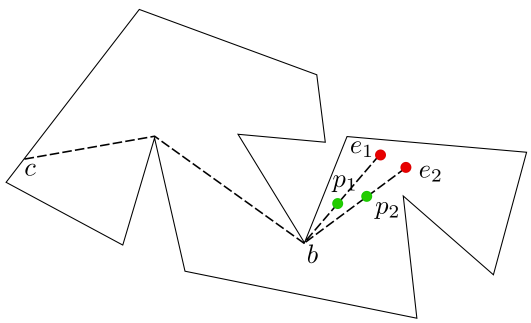


Fig. 6. Lion's strategy in a polygon.

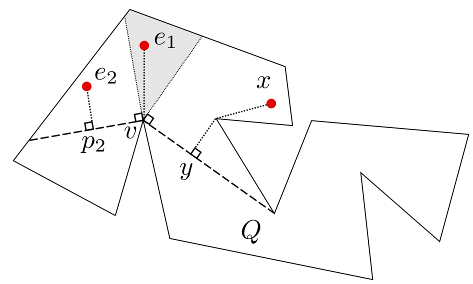


Fig. 7. The closest point projection is illustrated in a simply connected polygon.

tine for solving the lion and man game in environments more complex than simply-connected polygons.

We say that a pursuer **guards** a path Π when p can immediately respond with a capture move whenever the evader steps onto or across Π.

To begin, let's return to our square example in Figure 1(a). Let Π be a horizontal line segment connecting two boundary points. For any point $e \in \mathcal{S}$, let $\pi(e)$ be the **vertical projection** of e onto Π. Basic geometry shows that if e moves to a point e' then $d(\pi(e), \pi(e')) \leq d(e, e')$. This means that if $p = \pi(e)$ then it can move to $\pi(e')$, so the pursuer can maintain this property indefinitely. Furthermore, if e steps onto or across Π then p can respond with capture. In summary, a pursuer positioned at $\pi(e)$ can guard Π, restricting the evader to a subset of the environment. Moreover, it is clear that a pursuer can achieve $p = \pi(e)$ in finite time: start at the left endpoint of Π and then walk rightward at full speed until reaching $\pi(e)$.

We are now ready for a formal definition of projection mappings.

Definition 1 (Projection Map): A projection $\pi : \mathcal{S} \rightarrow Q$ is a function such that (1) if $x \in Q$ then $\pi(x) = x$, and (2) for all $x, y \in \mathcal{S}$, we have $d_Q(\pi(x), \pi(y)) \leq d_{\mathcal{S}}(x, y)$.

Suppose that the pursuer is located at $p = \pi(e) \in Q$ where $\pi : \mathcal{S} \rightarrow Q$ is a projection. It can be shown that if the evader moves from e to e' then the pursuer can respond by moving to $p' \in Q$ satisfying $p' = \pi(e')$. In particular, if $e' \in Q$ then the pursuer responds by capturing the evader [8].

As with lion's strategy, this idea can be adapted for a simply connected polygon P. Given two boundary points $u, v \in \partial P$, let Π be the unique shortest path between them. Define the **closest point projection** $\rho : \mathcal{S} \rightarrow \Pi$ to be the mapping that takes $x \in P$ to the point $y \in \Pi$ that is closest to x, see Figure 7. The vertical projection above is just a special case of the closest point projection. As in the square region, a pursuer can establish a guarding position on Π and maintain it thereafter, trapping the evader in a sub-polygon.

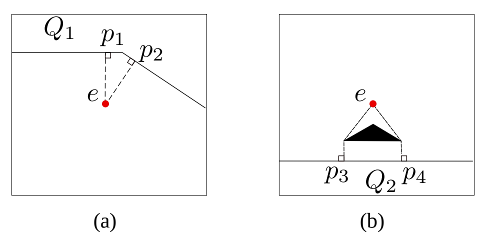


Fig. 8. Two examples where the closest point mapping is not a projection mapping. In these examples, the closest point is not unique. **(a)** In the first example, since Q_1 is not convex, there are two points that are closest to e in Q_1 (p_1 and p_2). **(b)** In the second example, due to the obstacle in $\mathcal{S} \setminus Q_2$, the point e has two closest points in Q_2 (p_3 and p_4). Therefore, the evader can move such that its closest point moves faster. Thus, the pursuer cannot stay on the closest point of the evader.

The situation changes once we introduce obstacles to the environment: the closest point mapping becomes ill-defined when there are obstacles between e and Π, see Figure 8. Bhadauria et al. [8] introduce an alternate type of projection that is less intuitive, yet robust in the presence of obstacles. Let $a, b \in \partial\mathcal{S}$ and let Π be a shortest (a, b)-path, which we refer to be **anchored** at a. The **path projection** of the point $x \in \mathcal{S}$ is the point $y \in \Pi$ such that $d(a, y) = d(a, x)$. If $d(a, x) > d(a, b)$, then we simply define $\pi(x) = b$. See Figure 9 for an example. The path projection remains unique, even when there are obstacles in the environment. A pursuer on the path projection $\pi(e)$ can guard Π, meaning that $d(\pi(e), \pi(e')) \leq d(e, e')$, and that the pursuer can capture the evader whenever it crosses Π [8].

Verifying that a path projection satisfies $d(\pi(x), \pi(y)) \leq d(x, y)$ is straight-forward, though there are multiple cases to consider depending on the distance of $x, y \in \mathcal{S}$ from the anchor a. The validity of this projection means that a single pursuer can guard a path and thereby restrict the sub-environment available to the evader, even in the presence of obstacles. Next, we show how guarding shortest paths leads to winning capture strategies in two families of environments.

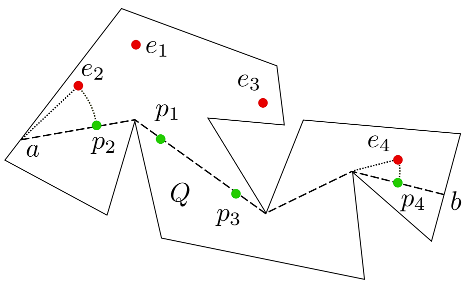


Fig. 9. The path projection anchored at a.

A. Applications

Path guarding is a powerful subroutine for designing pursuit strategies. For example, consider the lion and man game in a polygon P with obstacles. The shortest path $\Pi(a, b)$ between boundary points a, b splits P into two or more regions. Guarding $\Pi(a, b)$ confines the evader to one of these regions (Figure 10(a)). This process suggests a divide-and-conquer strategy for multiple pursuers where the pursuers constrain the evader into smaller and smaller regions and finally capture it. The **shortest path strategy** of Bhadauria et al. [8] exploits this idea and guarantees capture with only three pursuers in any polygonal environment with obstacles. (They also provide an alternative **minimal path strategy** that we will discuss in Section V.) Beveridge and Cai [7] adapted the shortest path strategy to more general environments in the plane. Klein and Suri [10] showed that an analogous surround-and-contract pursuit strategy works on polyhedral surfaces using four pursuers. Recently, Noori and Isler [9] proved that for positive capture radius $r > 0$, three pursuers are sufficient on polyhedral surfaces. We note that the classic Aigner and Fromme result that three pursuers can catch an evader in a planar graph [11] employs a similar split-and-guard strategy. While Aigner and Fromme do not use the term “projection,” their pursuit strategy employs path projections to guard a sequential family of shortest paths on a graph.

Next, we present a brief description of the above geometric capture strategies. The main idea is the same for polygons and polyhedra. The pursuers’ strategy is divided into rounds. In each round, two pursuers, namely p_1 and p_2, guard two shortest paths Π_1 and Π_2 respectively by locating themselves on the path projection of the evader (Figure 10(b)). The evader is then restricted to a subset of the environment that is bounded by these two shortest paths and perhaps part of the boundary of the environment: this subset is the contaminated region. The third pursuer p_3 guards a third shortest path Π_3 inside the contaminated region. The path Π_3 splits the contaminated region into two smaller subsets. The evader is now restricted to one of these subsets. Without loss of generality, suppose that the evader is between Π_1 and Π_3 (Figure 10(b)). Crucially, the path Π_3 is selected such that p_2 can be released. Therefore, in the next round, the contaminated region is guarded only by p_1 and p_3, so p_2 can continue the splitting process.

We now describe the splitting step in more details. We may assume that $\Pi_1 = \Pi(a_1, b_1)$ and $\Pi_2 = \Pi(a_2, b_2)$ are internally disjoint paths (otherwise, the evader is actually trapped in a smaller region). If these paths are completely disjoint, then either $\Pi(a_1, b_2)$ or $\Pi(a_2, b_1)$ can be used for splitting (Figure 10(b)). Next, suppose $a_1 = a_2$ but $b_1 \neq b_2$. In this case, we can pick a point c along the portion of the boundary from b_1 to b_2 and use the path $\Pi(a_1, c)$ to make progress.

The remaining case is when $a_1 = a_2$ and $b_1 = b_2$, so that Π_1, Π_2 are distinct shortest (a_1, b_1)-paths. We handle this case differently in polygons and polyhedra. In polygons, multiple shortest paths between two points a_1, b_1 occur only when they both touch obstacles (Figure 10(c)). In this case, the contaminated region is disconnected and we can make progress by removing components that do not contain the evader. In particular, if there is an obstacle that touches both Π_1 and Π_2, then the evader is actually trapped by this obstacle as well, so we can replace Π_1, Π_2 with shorter paths that only share one endpoint. Finally, if there is an obstacle that only touches Π_1, then we can choose $\Pi_3 \neq \Pi_1$ to be a shortest path that circumvents this obstacle.

The situation is more subtle for a polyhedral surface. Obstacles are not required for multiple shortest paths: the hills and valleys of the surface contribute a variety of shortest paths (Figure 10(d)). Without guaranteed obstacles, the evader can move freely in the region in between Π_1 and Π_2. This complicates the choice of the splitting path (Figure 10(d)). In particular, after removing Π_1 and Π_2, the next shortest path between a_1 and b_1 can be infinitesimally close to either Π_1 or Π_2. Let us refer this candidate path as $\Pi_{\min}$. If we choose Π_3 as $\Pi_{\min}$, then the splitting procedure may go on forever and we cannot have finite time capture. On the other hand, if we choose any other path as Π_3, then p_3 cannot guard Π_3 because $\Pi_{\min}$ is shorter. Klein and Suri [10] resolve this situation by employing a fourth pursuer, while Noori and Isler [9] exploit a non-zero capture radius to achieve capture with three pursuers.

The four-pursuer strategy on polyhedral surfaces proposed by Klein and Suri [10] takes advantage of the following observation: When there are distinct shortest paths Π_1, Π_2

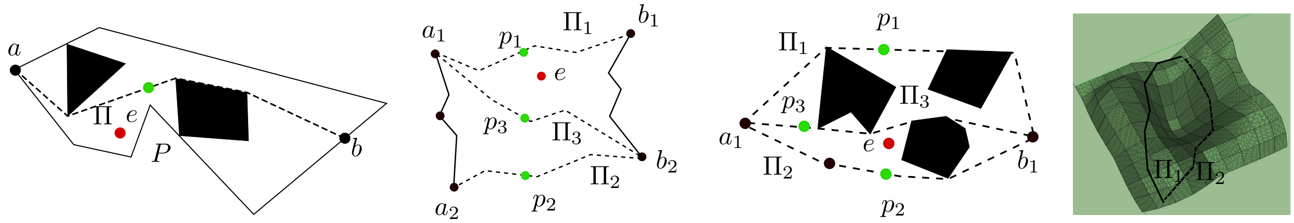


Fig. 10. **(a)** The shortest path Π divides the polygon into at least two regions in which one of them contains the evader. **(b)** Shortest paths Π_1, Π_2, and Π_3 are shown with dashed lines and $\partial\mathcal{S}$ is shown with solid lines. After guarding Π_3 by p_3, the evader will be restricted to a smaller region. **(c)** Multiple shortest paths in polygonal environments are caused by obstacles. **(d)** On a polyhedron multiple shortest paths are caused by hills and valleys. From [9].

between two points a_1, b_1, there must be at least one vertex in the region between them. Let v be this vertex. In order to split the evader region, the third and the fourth pursuers are assigned to guard the shortest paths $\Pi_3 = \Pi(a_1, v)$ and $\Pi_4 = \Pi(b_1, v)$ respectively (Figure 11(a)). As a result, the pursuers maintain the invariant that the contaminated region is guarded by at most three pursuers. The pursuers continue the strategy until the evader is confined in a single triangular face. At this point they follow a similar divide and guard strategy until the length of at least one edge of the triangle is at most one, in which case one pursuer can sweep the entire triangle and capture the evader.

The three-pursuer strategy on polyhedral surfaces proposed by Noori and Isler [9] uses the observation that when the capture radius is non-zero, there is a safe region around each of the shortest paths guarded by pursuers. This region is called the **capture region** $C(\Pi)$, and is defined by:

$$C(\Pi) = \{q \in \mathcal{S} : \exists p \in \Pi, \quad d(p,q) \leq r/2\}.$$

The evader cannot enter $C(\Pi)$ without being captured by the pursuer that guards Π [9]. The intersection points of the boundary of the two capture regions $C(\Pi_1)$ and $C(\Pi_2)$ are used as the endpoints of the splitting path Π_3 (Figure 11(b)). With this adaption, the pursuit proceeds as in a polygonal environment in the plane.

In this section, we have seen that simply by guarding shortest paths, we can devise successful pursuit strategies for multiple pursuers. In the next section, we will consider a more aggressive pursuer that clears territory on its way to guarding a path.

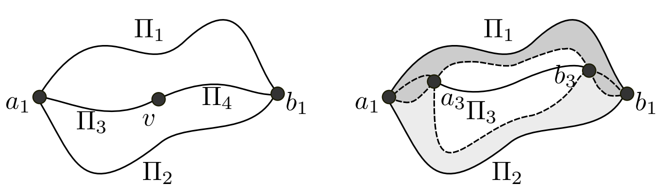


Fig. 11. **(a)** Klein and Suri [10] use a fourth pursuer to handle multiple shortest path case. **(b)** Noori and Isler [9], exploit the non-zero capture radius to define the splitting path and ensure capture with three pursuers.

V. Guarding Minimal Paths

In this section, we further generalize the projection and path guarding concepts introduced in Section IV. We start with a motivating example, shown in Figure 12(a). The path Π partitions $\mathcal{S}$ into two pieces R and Q: we have $R \cap Q = \Pi$ and $R \cup Q = \mathcal{S}$. Note that Π is not a shortest (a,b)-path in $\mathcal{S}$. However, it is a shortest path *in the sub-environment* R. Therefore, we can define a closest point projection $\pi : R \to \Pi$. Now, suppose that $e \in R$ and that p is on the closest point projection $\pi(e)$. Then p can guard Π and prevent e from entering the region Q.

In summary, a path doesn't have to be a shortest path to be guardable: it just needs to be a shortest path with respect to the current evader territory. We call such a path a **minimal path** in R, reserving the term "shortest path" for a global shortest path. Note that attaining position on a minimal path is trickier than attaining position of a (global) shortest path. Below, we explain how to attain position when Q is simply connected, and then give some examples of pursuit strategies that use minimal paths. We start by defining a type of projection that will be useful for a pursuer who hopes to guard a minimal path.

Definition 2: Let $\Pi \subset \mathcal{S}$ be a path between boundary points $u, v \in \partial\mathcal{S}$ that splits the environment into subset Q, R, with $Q \cap R = \Pi$ and $Q \cup R = \mathcal{S}$. If Π is minimal in R then the **path projection** $\pi : \mathcal{S} \to Q$ is the piecewise function that is the identity map on Q and is the path projection from R to Π on $\mathcal{S} \backslash Q$.

We now put this new projection to work. Consider an environment $\mathcal{S} = Q \cup R$ where $\Pi = Q \cap R$ is a minimal (a,b) path in R. If Q is simply connected then the pursuer can establish $p = \pi(e)$ by performing the Lion's strategy in Q (Section III). If $e \in Q$, then $e = \pi(e)$, so $p = \pi(e)$ implies that the evader is caught. If $e \notin Q$, then the pursuer is on the path projection of e onto Π. From this point forward, the pursuer can maintain its position on the projection. Therefore, it can guard Π, and prevent e from ever returning to Q. Similar to the shortest path strategy, we can use this minimal path guarding strategy to devise capture strategies.

We take a moment to compare guarding a shortest path with guarding a minimal path in R. In the former case, the pursuer is agnostic of the evader's location with respect to the shortest path Π. Once the pursuer has established $p = \pi(e)$, the evader is trapped on one side of Π, but the pursuer has no influence over which side. Furthermore, both sides of Π may contain obstacles– this has no bearing on the ability of the pursuer to guard the shortest path Π. Now consider a

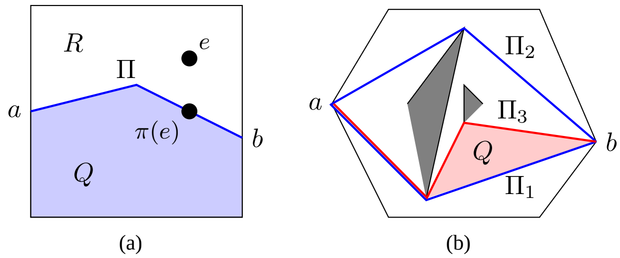


Fig. 12. **(a)** The path Π is minimal in R, but not in Q. If $e \in R$ and $p = \pi(e)$ is on its projection, then the evader is trapped in R. **(b)** The path Π_3 is the third shortest loop-free (a, b)-path in the visibility graph, but it is not a minimal path in Q.

path Π that is minimal in R, but not minimal in the simply connected Q. In the process of attaining a guarding position on Π, the pursuer also clears the region Q. In fact, this is a necessary condition for guarding Π, since it is only guardable provided that we know that the evader is in R. Next, we give two examples where guarding minimal paths leads to winning pursuit strategies.

A. Applications

For our first application, we return to pursuit in polygonal environments with obstacles. We have already summarized the Shortest Path Strategy for this setting (Section IV-A). Here we consider the second pursuit method of Bhadauria et al. [8]: the Minimal Path Strategy. Just like the Shortest Path Strategy, the pursuers take turns, guarding and splitting the evader region. The difference is how they choose the paths to guard. The pursuers construct their paths using the *visibility graph* $G(\mathcal{S})$ of the environment. The vertices of this weighted graph correspond to the vertices of the environment. Two vertices are adjacent when they are connected by a line segment that completely lies inside the polygon. Each edge is assigned a weight equal to the length of the edge. Paths in $G(\mathcal{S})$ correspond to paths in the original environment $\mathcal{S}$. We can use graph algorithms to find our minimal paths.

We start by choosing boundary points a and b. Pursuer p_1 guards the shortest (a, b)-path Π_1, which is minimal in $\mathcal{S}$. Pursuer p_2 guards path Π_2 corresponding to the next shortest loop-free (a, b)-path in the visibility graph. Then, pursuer p_3 guards the next shortest loop-free (a, b)-path and so on. Once the evader is trapped in a simply connected region, the free pursuer uses lion's strategy. There is one subtlety to guarding this sequence of paths. Suppose that path Π_3 splits the evader region into Q and R, where at least one region, say R, contains an obstacle, see Figure 12(b). By construction, the path Π_3 is minimal in R, provided that Π_2 is already guarded. If Q also contains an obstacle then Π_3 will also be minimal in Q as long as Π_1 is guarded. In this case, p_3 can simply move to guard this new path. However, when Q is simply connected, Π_3 will not be a minimal path in Q. In this case, the pursuer p_3 uses lion's strategy to force the evader to move outside Q. Once p_3 reaches Π_3, he simultaneously guards Π_3, trapping the evader in R.

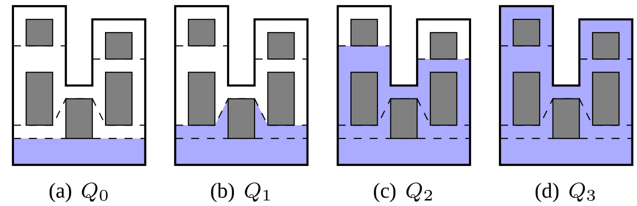


Fig. 13. A leapfrog partition $Q_0 \subset Q_1 \subset Q_2 \subset Q_3$. Region Q_1 intersects two more obstacles than Q_0. Region $Q_3 \backslash Q_2$ is disconnected. Pursuer p_1 clears Q_0, then pursuer p_2 clears $Q_1 \backslash Q_0$. This releases p_1, who leapfrogs over p_2 to clear $Q_2 \backslash Q_1$. Finally, p_2 leapfrogs over p_1 to clear $Q_3 \backslash Q_2$.

This clear-and-guard strategy is the foundation for a two-pursuer strategy that is appropriate for certain polygonal environments with obstacles [2]. Figure 13 shows an example of such an environment. Suppose that there is a family of subsets $Q_0 \subset Q_1 \subset \cdots \subset Q_k = P$ where (1) Q_0 is simply connected, (2) there is a projection $\pi_i : Q_{i+1} \to Q_i$, and (3) $Q_{i+1} \backslash Q_i$ is a collection of simply connected regions. Then two pursuers can alternately take control of each Q_i by **leapfrogging** over one another to clear and guard the next region. See [2] for some sufficient conditions for an environment to have such a leapfrog decomposition.

VI. Rook's Strategy

In Sections IV and V, the guarding pursuer plays a purely defensive role: his sole objective is to prevent the evader from entering a subregion $Q \subset \mathcal{S}$. Therefore, another (non-guarding) pursuer must take on the offensive role of reducing the evader territory. In this section, we introduce **rook's strategy** as an alternative to lion's strategy for actively chasing the evader. Simply put, rook's strategy is designed to simultaneously guard and make progress. We present two applications where we will see the advantages of choosing rook's strategy over lion's strategy.

First, we adapt the chess strategy introduced in Section I-A to the lion and man game in a square environment. Suppose that the pursuer is guarding a straight line segment Π between two boundary points. As we observed in Section I, when $r = 0$, guarding Π requires $p = \pi(e)$. As a counter-strategy, the evader can oscillate horizontally between two points, unit distance apart (Figure 1(b)). In response, the pursuer must exhaust its movement budget just to keep pace, so the guarded path never advances. However, when we have a positive capture radius $r > 0$, the pursuer guards Π as long as $0 \le d(p, \pi(e)) < r$. We claim that this slack allows the pursuer to reliably advance the guarded path.

From here forward, we consider a game with positive capture radius $r > 0$ and fix a constant $0 < \tau \le \min\{r, 1/2\}$. Given a projection π onto path Π, we say that p is in **rook position** when $0 \le d(p, \pi(e)) \le \tau$. A pursuer that is within this horizontal offset τ is in a quite powerful position, as we explain below.

Suppose that the pursuer is offset τ units to the right of the evader projection (Figure 4(c)). If the evader moves leftwards one unit, then the pursuer matches pace and maintains this offset. However if the evader moves rightward, then the pursuer switches to using a leftward offset instead (Figure 4(d), (e)). As a result, the pursuer only needs to move rightward $1-2\tau$ units, which means that it can move diagonally to achieve at least $(1-(1-2\tau)^2)^{1/2} > \tau$ units of upward progress. The key observation is that the evader must move rightward at some point, since diam($\mathcal{S}$) is finite, so the pursuer makes at least τ units of vertical progress every diam($\mathcal{S}$) steps.

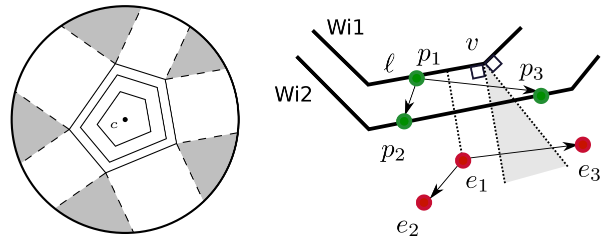
It is worth noticing the connection between the lion's strategy and the rook's strategy to see how the pursuer guards and attacks simultaneously. During lion's strategy, the pursuer consistently radiates outwards from its initial point c. Let p^t be the location of the pursuer at time t and let $B^t = B(c, d^t)$ where $d^t = d(c, p^t)$. Each pursuer move decreases the evader territory: after time t, the evader cannot step into the region B^t. Indeed, the pursuer is actually located on the closest point projection onto ∂B^t, and it is useful to view p as guarding the expanding sequence of **wavefronts** $\partial B^1, \partial B^2, \ldots, \partial B^t$. The rook's strategy also controls a sequence of advancing wavefronts. One advantage of rook's strategy is that its wavefronts are straight lines (or piecewise linear paths). This makes it a natural fit for polygonal and polyhedral environments, where it can lead to simpler pursuit algorithms than those employing lion's strategy.

A. Centered Rook's Strategy

It may seem that rook's strategy is only suited for rectilinear pursuit, since it is crucial that the evader must "turn around" when it encounters the left or right boundary. We can make rook's strategy more powerful by expanding the wavefronts from a central point similar to the lion's strategy. This **centered rook's strategy** will guard wavefronts that bound a family of convex polygons (rather than regions with curved boundaries). We use the centered rook's strategy in Section VI-B below when we consider pursuit-evasion on convex terrains (surfaces) in $\mathbb{R}^3$.

Consider a pursuit-evasion game in a convex polygon $\mathcal{S}$ with capture radius $r > 0$. Fix an offset $0 < \tau \leq \min\{r, 1/2\}$. Pick a center $c \in \mathcal{S}$ and let A be a convex polygon such that $\max_{x \in A} d(c, x) = 1$. We explain how p can guard a monotonically increasing family of regions $A_i = \{b_i x \mid x \in A \cap \mathcal{S}\}$ where $b_1 = 1$ and $b_{i+1} \geq b_i$. We call b_i the **radius** of the guarded region. We use the closest point projection (which returns consistent values, even when the evader circumnavigates A_i). Note that this projection onto A_i partitions $\mathcal{S}$ into distinct areas, according to the pre-images of the vertices and sides of A_i, see Figure 14(a).

The pursuer starts at c. On its first move, p moves to within τ of the closest point projection on A, so that it is now guarding $A_1 = A$. Now, suppose that p currently guards A_t. We claim that in finite time, the pursuer can increase the radius b_t of the guarded region by a constant amount. Suppose that the evader projection $\pi(e)$ is on side ℓ of A_t, so the evader is in the pre-image of ℓ. See Figure 14(b). If the evader never leaves this pre-image, then the pursuer makes progress as in regular rook's strategy. Otherwise,



(a) (b)

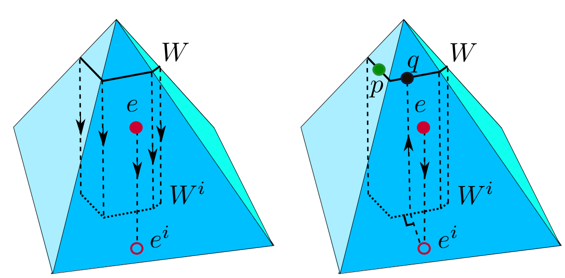
Fig. 14. The centered rook strategy. (a) The expanding wavefronts, and the pre-images of the edges and vertices of polygon A_t centered at c. (b) Two progress events are depicted. Suppose that the evader is currently at e_1 and the pursuer is guarding W_1 on the edge ℓ. If the evader's projection stays on ℓ the pursuer moves to p_2 by rook's strategy. If the evader crosses the pre-image of v, the pursuer makes progress by moving to p_3.

the evader must step into (or through) the pre-image of a vertex of A_t. As the evader crosses the pre-image of vertex v, its projection remains fixed on v. This frees up part of the pursuer's movement budget for progressing to the next wavefront (Figure 14(b)). For full details, see [12]. In Section VI-B we will present an application of the centered rook strategy in capturing the evader on convex terrains.

B. Application: Convex Terrains

In this section, we adopt the centered rook's strategy to pursuit-evasion with capture radius $r > 0$ on a piecewise linear convex **terrain** $\mathcal{S}$ in $\mathbb{R}^3$ [12]. A terrain is a height map where every point in the xy-plane has a single height value.

As always, a well-suited projection function is the key to successful pursuit. The main trick here is to relate the centered rook's strategy on the two-dimensional plane to the game on the surface. To do so, first define the wavefronts on the surface as follows. A wavefront at height h is the set of points on the intersection of the surface with the plane $z = h$. Let W be the wavefront that the pursuer is currently guarding. Next, the wavefront and the players are vertically projected onto the xy-plane i.e., a point $q = (x, y, z)$ is mapped to $q^i = (x, y, 0)$, Figure 15(a). Let e^i and W^i be the vertical image of the evader and the current wavefront on the xy-plane. Then the projection of the evader onto W is the point $\pi(e)$ on the surface such that $\pi^i(e)$ is the closet point projection of e^i onto W^i, Figure 15(b).



(a) (b)

Fig. 15. (a) Images of the point e and also the wavefront W are shown. (b) The projection of e onto W ($\pi(e) = q$) is depicted. The pursuer positions itself at distance τ from $\pi(e) = q$ (the distance is measured along W).

After defining the projection mapping, the application of the centered rook's strategy is straightforward. For guarding and making progress the pursuer moves such that its vertical image p^i, on the xy-plane obeys the centered rook's strategy in the two-dimensional plane.

C. Application: Line of Sight Visibility in Sweepable Polygons

So far we assumed that both the pursuer and the evader can observe each others' location throughout the game and the pursuer's only goal is to get within the capture radius of the evader. The game becomes more complicated when the information available to the pursuer is limited due to imperfect sensing capabilities. For example, suppose that the pursuer is equipped with a camera and therefore has line-of-sight visibility, meaning that it can observe the location of the evader only when the line segment between them is completely inside the polygon. (We continue to assume that the evader has a full map of the environment including the lion's position.) In this case, the pursuer must alternate between searching for the evader (when hidden) and actively chasing him (when visible). Throughout, the pursuer must make progress toward capture, and maintain that progress when transitioning between searching and chasing.

Noori and Isler [13] successfully combined search and progress strategies in order to capture the evader in *monotone polygons*. A simple polygon is called **monotone** with respect to a line L if for any line L' perpendicular to L the intersection of the polygon with L' is connected [14]. In the proposed strategy in [13], the pursuer makes progress by chasing the evader with lion's strategy with the additional trick of changing the center among multiple points that are lined up in syzygy. We refer the interested reader to [13] for the details of the idea which can be added to the toolbox.

Berry et al. [2] describe a line-of-sight pursuit strategy that uses rook strategy (rather than lion's strategy). They consider the game played in strictly sweepable polygons, which are a generalization of monotone polygons. A polygon is **strictly sweepable** if a straight line can be moved continuously over the entire polygon (via a sequence of translations and rotations) such that (1) the intersection of the line and polygonal area is always a connected line segment, and (2) no point is swept more than once. In particular, a monotone polygon is the special case of sweeping the polygon via a single translation. Crucially, successful pursuit requires adjustments in the pursuer's point-of-view during the game. Rather than using a fixed "horizontal" direction for the rook moves, we must change the pursuer's frame of reference to match the the sweep line as it rotates through the polygon. We describe this process for a monotone polygon and for a **scallop polygon**, where the sweep line rotates through a fixed center point. We refer the reader to [2] for the general case, where the sweep line alternates between translations and rotations about different center points.

The sweepable polygon strategy guards a monotone increasing family of subsets against recontamination. The key to success is the choice of search path. For a monotone polygon, the search path traverses the polygon as horizontally as possible, following the boundary where necessary, Figure 16(a). Every time that the pursuer touches a new boundary point, the guarded region is updated. We construct the search path for the scallop polygon in the same way, updating our notion of horizontal each time that we pass a new vertex of the polygon, Figure 16(b). At each vertex, the pursuer changes its frame of reference, which requires moving down the sweep line until the new frame of reference protects the previously guarded region, Figure 17. Crucially, the pursuer does not commit to the new frame of reference until this transition is complete. This prevents recontamination. Rook's move also requires some adaptations to deal with hiding and blocking moves by the evader. For more details, see [2].

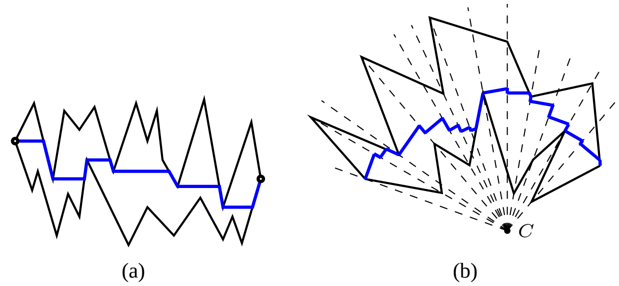


Fig. 16. (a) The search path for a monotone polygon. (b) The search path for a scallop polygon.

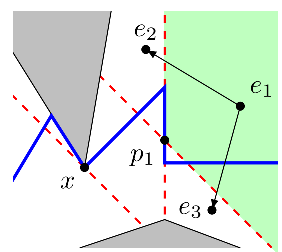


Fig. 17. Changing frame during the search phase in a scallop polygon. If e moves out of the green area while p moves down the search path, the pursuer immediately switches to the frame of reference (whose vertical is shown in red) that guards the point x.

VII. Exercises for STEM Educators

In this section, we collect some introductory exercises for students. Some exercises ask you to work through the details of claims made in the paper, or apply results from this paper to examples. Others explore novel formulations. When appropriate, the most relevant section for the exercise is specified in the question, or parenthetically. Unless otherwise stated, these exercises assume that the capture radius $r = 0$.

1) Consider the two guard strategy (Approach 2) of Section I-A in a $D \times D$ square environment.
    a) If the pursuers are positioned on the appropriate projections of the evader, show that one of them makes at least $\sqrt{2}/2$ progress in each turn.

b) Assuming that the pursuers choose their initial positions and then the evader chooses his initial position, find the best upper bound that you can for the capture time. What pair of initial configuration and evader strategy achieves this maximal capture time?

2) Prove that the lion's strategy inside a circle with radius R results in capture in R^2 steps. (III)
3) Consider a polygonal environment with obstacles, as in Section IV-A. Suppose that p_1 guards shortest path $\Pi(a_1, b_1)$ and p_2 guards shortest path $\Pi(a_2, b_2)$. If these paths have common internal vertices, prove that the evader is actually trapped in a smaller region than the region bounded by $\Pi(a_1, b_1)$ and $\Pi(a_2, b_2)$.
4) Create the visibility graph for the environment in Figure 18(a). Use this graph to find the first, second and third shortest (a, b)-paths. (V-A)
5) Show that both of the environments in Figure 18 are two-pursuer win. (V-A)

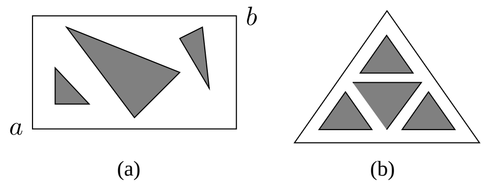


Fig. 18. Environments for the exercises. Both are two-pursuer win.

6) Consider a game in a square region with capture radius $r = 1$. Suppose that the pursuer is in rook position on horizonal line L, meaning that $d(p, \pi(e)) \leq 1/2$. Prove that if e moves to a point e' that is within $\sqrt{3}/2$ of line L, then p can respond by capturing the evader. Conclude that evader cannot cross line L. (VI)
7) Consider a game with pursuers p_1, p_2 in a square region (with capture radius $r = 0$). Suppose that the pursuers are positioned on horizontal line L at the points that are $\pm 1/2$ away from $\pi(e)$. Prove that if e moves to a point e' that is within $\sqrt{3}/2$ of line L, then one of the pursuers can respond by capturing the evader. (VI)
8) A pursuer can capture an evader in a simply connected polygon when the pursuer knows the evader's location at all times. Suppose that the pursuer has line-of-sight visibility. Design an environment and an escape strategy for the evader so that if the evader knows the pursuer's path[2], it can escape.
9) On the surface of a sphere, the shortest path between any two points is the great circle passing through them.
   a) Using the great circles passing through the north pole, devise a capture strategy for the pursuer when the players are moving on a half-sphere.
   b) Design a capture strategy for two pursuers on the surface of a sphere.
10) Add one convex obstacle to a square environment.
   a) Provide conditions on the diameter of the obstacle such that a single pursuer can capture the evader.
   b) For a large enough obstacle, design an escape strategy for the evader.
   c) Provide some conditions on the initial locations of the players such that the pursuer can capture the evader in the environment from part (b).

[2]This is the case when the pursuer's strategy is deterministic.

VIII. Open Problems and Concluding Remarks

The field of pursuit-evasion remains rich with interesting, thought-provoking questions and open-problems. We conclude with a list of open problems related to the lion and man game.

Throughout this tutorial, we focused on deterministic strategies for the full-visibility pursuit-evasion problems where the players can observe the location of their opponent at all times. In more complicated setups where the pursuer has limited information about the location of the evader, randomized strategies can be used to overcome the pursuer's lack of information [6], [15]. Application of the randomized strategies to visibility-based versions of the problem and realizing their connection to deterministic strategies is an interesting venue for further research.

While we focused on the turn-based model, the techniques presented in this paper are applicable to the continuous-time version. It can be shown that any given turn-based capture strategy to get within distance r of the evader can be adopted to a continuous-time capture strategy with capture radius $r+s$ where s is the step size (Lemma 1). Therefore, as long as the pursuer can change its step-size s, the capture guarantee for the continuous time model can get arbitrarily close to the capture guarantee in the turn-based model[3]. However, this argument is not applicable when planning must be performed in the configuration space. Moreover, the problem is mostly open in the presence of nonholonomic constraints [16] on the motion of the players.

We saw that in simply connected polygons one pursuer is enough for capture while in the presence of obstacles three pursuers are sufficient and sometimes necessary [8]. An interesting question is to determine the classes of polygons that are two-pursuer-win [1]. Similarly, while three pursuers are sufficient for capture on general three-dimensional surfaces, the set of one-pursuer-win or two-pursuer-win surfaces are unknown.

Turning to the line-of-sight version of the game, we do not yet have a full characterization of the class of polygons that are pursuer-win. An intermediate goal would be to determine whether **sweepable polygons** are pursuer win. In this family of polygons, the sweep line may visit points more than once. This relaxation cofounds the notion or progress described herein for strictly sweepable polygons.

In recent years Unmanned Autonomous Vehicles (UAVs) have been receiving increasing attention. Interesting variants

[3]In the turn-based strategy, the time unit Δt can be chosen arbitrarily small. Consequently, the step-size s can be arbitrarily small since the players' speed is fixed.

of the problem can be designed when the players have the ability to fly. For example, suppose that the evader is restricted to move on the ground which is modeled as a geodesic terrain while the pursuer is able to fly. What is the outcome of the game subject to limitations on the highest altitude accessible by the pursuer? Can we provide guarantees on capture when we have a heterogeneous team of flying and ground pursuers?

Finally, most of the pursuit-evasion research is focused on the equal speed assumption. A full characterization of the game regarding different speeds for the pursuer and the evader is still unknown [17].

IX. Appendix

Lemma 1: If the pursuer can get within distance r of the evader in the turn-based model, then it can get within distance $r + s$ of the evader in the continuous time model where s is the step size, i.e. the distance that the players can travel in one time unit Δt.

Proof: In the continuous time model, the pursuer considers the continuous movement of the evader at discrete time steps with its specific time unit Δt. Then it plays the same turn-based pursuit strategy with respect to the location of the evader at $t - \Delta t$. Notice that the capture condition in the turn-based version is whether the distance between the players becomes less than r. With the aforementioned modification to the continuous setting, the capture guarantee is that the pursuer will decrease its distance to the evader to at most $r + s$. ■

References

[1] B. P. W. Ames, A. Beveridge, R. Carlson, C. Djang, V. Isler, S. Ragain, and M. Savage, "A leapfrog strategy for pursuit-evasion in a polygonal environment," *International Journal on Computational Geometry and Applications*, vol. 25, pp. 77–100, 2015.

[2] L. Berry, A. Beveridge, J. Butterfield, V. Isler, Z. Keller, A. Shine, and J. Wang, "Line-of-sight pursuit in strictly sweepable polygons," 2014.

[3] J. E. Littlewood, *A Mathematician's Miscellany*. Methuen London, 1953.

[4] J. Sgall, "Solution of David Gale's lion and man problem," *Theoretical Computer Science*, vol. 259, no. 1–2, pp. 663 – 670, 2001.

[5] L. Alonso, A. S. Goldstein, and E. M. Reingold, "'Lion and Man' : Upper and Lower Bounds," *Informs Journal on Computing*, vol. 4, pp. 447–452, 1992.

[6] V. Isler, S. Kannan, and S. Khanna, "Randomized pursuit-evasion in a polygonal environment," *Robotics, IEEE Transactions on*, vol. 21, no. 5, pp. 875–884, Oct 2005.

[7] A. Beveridge and Y. Cai, "Two dimensional pursuit-evasion in a compact domain with piecewise analytic boundary," http://arxiv.org/abs/1505.00297.

[8] D. Bhadauria, K. Klein, V. Isler, and S. Suri, "Capturing an evader in polygonal environments with obstacles: The full visibility case," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1176–1189, 2012.

[9] N. Noori and V. Isler, "Lion and man game on polyhedral surfaces with boundary," in *IEEE Conference on Intelligent Robots and Systems (IROS)*, 2014.

[10] K. Klein and S. Suri, "Pursuit evasion on polyhedral surfaces," in *Algorithms and Computation*, ser. Lecture Notes in Computer Science, L. Cai, S.-W. Cheng, and T.-W. Lam, Eds. Springer Berlin Heidelberg, 2013, vol. 8283, pp. 284–294.

[11] M. Aigner and M. Fromme, "A game of cops and robbers," *Discrete Applied Mathematics*, vol. 8, no. 1, pp. 1 – 12, 1984.

[12] N. Noori and V. Isler, "The lion and man game on convex terrains," in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2014.

[13] ——, "Lion and man with visibility in monotone polygons," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 155–181, 2014.

[14] M. De Berg, O. Cheong, and M. van Kreveld, *Computational Geometry: Algorithms and Applications*. Springer, 2008.

[15] M. Adler, H. Räcke, N. Sivadasan, C. Sohler, and B. Vöcking, "Randomized pursuit-evasion in graphs," *Combinatorics, Probability and Computing*, vol. 12, no. 03, pp. 225–244, 2003.

[16] Z. Li and J. F. Canny, *Nonholonomic motion planning*. Springer Science & Business Media, 2012, vol. 192.

[17] B. Tovar and S. M. LaValle, "Visibility-based pursuit-evasion with bounded speed," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1350–1360, 2008.