A Leapfrog Strategy for Pursuit-Evasion in a Polygonal Environment

Brendan Ames, Andrew Beveridge, Rosalie Carlson, Claire Djang, Volkan Isler, Stephen Ragain and Maxray Savage

January 14, 2014

Abstract

We study pursuit-evasion in a polygonal environment with polygonal obstacles. In this turn based game, an evader is chased by pursuers $p_1, p_2, \ldots, p_\ell$. The players have full information about the environment and the location of the other players. The pursuers are allowed to coordinate their actions. On the pursuer turn, each $p_i$ can move to any point at distance at most 1 from his current location. On the evader turn, he moves similarly. The pursuers win if some pursuer becomes co-located with the evader in finite time. The evader wins if he can evade capture forever.

It is known that one pursuer can capture the evader in any simply-connected polygonal environment, and that three pursuers are always sufficient in any polygonal environment (possibly with polygonal obstacles). We contribute two new results to this field. First, we fully characterize when an environment with a single obstacles is one-pursuer-win or two-pursuer-win. Second, we give sufficient (but not necessary) conditions for an environment to have a winning strategy for two pursuers. Such environments can be swept by a leapfrog strategy in which the two cops alternately guard/increase the currently controlled area. The running time of this algorithm is $O(n \cdot h \cdot \text{diam}(P))$ where $n$ is the number of vertices, $h$ is the number of obstacles and $\text{diam}(P)$ is the diameter of $P$.

More concretely, for an environment with $n$ vertices, we describe an $O(n^2)$ algorithm that (1) determines whether the obstacles are well-separated, and if so, (2) constructs the required partition for a leapfrog strategy.

1 Introduction

We study a pursuit-evasion game known as the lion and man game. In this game, which takes place in a polygonal environment, pursuers $p_1, p_2, \ldots, p_\ell$ try to capture an evader $e$. The environment consists of the polygon $P$ with some polygonal obstacles (or holes) $\mathcal{H} = \{H_1, H_2, \ldots, H_h\}$. The players are

---

*Institute for Mathematics and its Applications, University of Minnesota, Minneapolis MN 55455
†Department of Mathematics, Statistics and Computer Science, Macalester College, St Paul MN 55105
‡Department of Mathematics, University of California, Santa Barbara CA 93106.
§Department of Mathematics, Oberlin College, Oberlin OH 44074
¶Department of Computer Science, University of Minnesota, Minneapolis MN 55455
∥Department of Mathematics, Pomona College, Claremont CA 91711
located in \( P \setminus \bigcup_{H_i \in H} H_i \). We use \( P \) to denote the environment (with obstacles), and \( \partial P \) to denote its outer boundary. Each obstacle \( H_i \) is an open set, so that the players may occupy a point on its boundary \( \partial H_i \). A player located at point \( x \in P \) can move to any point in \( B(x, 1) = \{ y \in P \mid d(x, y) \leq 1 \} \), where \( d(x, y) = d_P(x, y) \) is the length of a shortest \((x, y)\)-path in \( P \). Let \( e^t \) denote the position of the evader at the end of round \( t \), and similarly \( p^t_i \) is the position of pursuer \( p_i \).

The game is played as follows. First, the pursuers choose their initial positions \( p_1^0, p_2^0, \ldots, p_\ell^0 \). Next, the evader chooses his initial position \( e^0 \). Gameplay in round \( t \geq 1 \) proceeds as follows. First, each pursuer \( p_i \) moves from its current position \( p_i^{t-1} \) to a point \( p_i^t \in B(p_i^{t-1}, 1) \). (Note that a pursuer may choose to remain stationary under these rules.) If \( p_i^t = e^{t-1} \) for some \( i \in \{1, \ldots, \ell\} \), then the pursuers are victorious. Otherwise, the evader moves according to the same rule, moving from \( e^{t-1} \) to a point \( e^t \in B(r_{t-1}, 1) \). The evader wins if he evades capture forever. We consider the full-information version of this game, where each agent knows the environment, as well as the location of all other agents. Furthermore, the pursuers may coordinate their strategy.

Pursuit-evasion in geometric environments has been extensively studied; for a survey, see [7]. The game’s history extends at least to the 1930s when Rado posed the Lion-and-Man problem in a circular arena, with lion chasing man [15]. Surprisingly, when time is continuous, man has a winning strategy [15, 2]. However in the turn-based version our natural intuition prevails: the lion has a winning strategy. The turn-based version has received a good deal of attention (cf. [2], [16], [14]) and it is known that a single pursuer can always catch an evader in a simply connected polygon [11]. Recently, Bhadauria et al. [4] proved that 3 pursuers can always capture an evader in any polygonal environment (with obstacles). This 3-pursuer result is analogous to Aigner and Fromme’s classic result about the pursuit-evasion game played a planar graph [1]. When played on a graph, this game is known as cops and robbers; see the surveys [3, 10] and the monograph [5]. Variants of pursuit-evasion with limited pursuer sensing capabilities have also been studied [9, 12] but we focus on the full-visibility case.

Herein, we study polygonal environments where one or two pursuers are sufficient for capture. Our results leverage two techniques: guarding and projection. Given a sub-environment \( Q \subset P \), we say that a pursuer \textit{guards} \( Q \) if (a) the evader is not currently in \( Q \) and (b) if the evader crosses into \( Q \), the pursuer can respond by capturing him. A \textit{projection function} \( \pi : P \to Q \) is a function that (a) is the identity map on \( Q \) and (b) satisfies \( d_Q(\pi(x), \pi(y)) \leq d_P(x, y) \). Note that the existence of a projection function \( \pi : P \to Q \) guarantees that \( Q \) is \textit{geodesically convex} in \( P \), meaning that for any \( x, y \in Q \), there is at least one shortest \((x, y)\)-path \( \Pi \subset Q \) (though there may be other shortest \((x, y)\)-paths that leave \( Q \)). We use projection functions to devise guarding strategies. Our main result is the following.
**Theorem 1.1** (Leapfrog Theorem). Suppose that $P$ contains a family of nested subregions $Q_0 \subset Q_1 \subset Q_2 \subset \cdots \subset Q_k = P$ and, for $0 \leq i \leq k - 1$, the following hold:

(L1) $Q_0$ is simply connected,

(L2) there is family of projections $\pi_i : Q_{i+1} \to Q_i$

(L3) $Q_{i+1} \setminus Q_i$ is a finite collection of simply-connected regions,

(L4) $Q_i$ intersects fewer obstacles than $Q_{i+1}$.

Then $P$ is 2-pursuer win.

In particular, this theorem holds for a family of nested subsets that are geodesically convex in $P$.

**Corollary 1.2.** Suppose that $P$ contains a family of nested subregions $Q_0 \subset Q_1 \subset Q_2 \subset \cdots \subset Q_k = P$ such that (L1), (L3), (L4) hold for $0 \leq i \leq k - 1$. If $Q_0 \cap \partial P$ contains at least two points and $Q_i$ is geodesically convex in $P$ for $0 \leq i \leq k$ then $P$ is 2-pursuer win.

In proving Theorem 1.1 we describe a leapfrog strategy for two pursuers. First, $p_1$ evicts the evaders from $Q_0$ and then guards this region. Next, the $p_2$ clears and guards $Q_1 \setminus Q_0$. In the process, $p_2$ ends up guarding all of $Q_1$, which frees up $p_1$ to leapfrog over $p_2$ to tackle $Q_2 \setminus Q_1$, and so on. See Figures 5.4 and 6.5 below for two example environments with such leapfrog decompositions. The leapfrog strategy completes in time $O(n \cdot h \cdot \text{diam}(P))$ where $n$ is the number of vertices of the environment (on both the outer boundary and on the obstacles), $h$ is the number of obstacles, and $\text{diam}(P)$ is the diameter of the environment. (Note that the fourth condition ensures that $k \leq h + 1$.)

Along the way, we also resolve the following question: when does one pursuer have a winning strategy a polygonal environment with a single obstacle $H$? The determining factor is the length of the perimeter of the convex hull of that obstacle.

**Theorem 1.3.** Suppose that $P$ is a polygonal environment with one obstacle $H$ with convex hull $J = \text{Hull}(H)$. Then $P$ is pursuer-win if and only if $J$ has perimeter $\ell \leq 2$.

Finally, we complement this work with some computational results which identify sufficient (but not necessary) conditions for when an environment has a decomposition as described in Theorem 1.1. In Section 6, we give an $O(n^2)$ algorithm for finding such a nested family of sets when the obstacles of $P$ are well-separated. This result, formulated as Theorem 6.3 below, requires technical definitions.
of sweepable polygons and strictly two-sweepable environments, so we defer its statement until later. This algorithm uses dual polygons and adapts the monotonicity results of Bose and van Kreveld [6].

The paper is organized as follows. Section 2 describes lion’s strategy, a known strategy for a single pursuer to capture an evader in a simply connected environment. Section 3 develops the projection and guarding framework that underlies our main results. In Section 4, we prove Theorem 1.3 and in Section 5, we prove Theorem 1.1. Section 6 contains the dual polygon algorithm that finds a leapfrog partition for a strictly two-sweepable environment. We conclude in Section 7 with some avenues for future research.

2 Lion’s strategy

The lion-and-man game (cf. [2]), takes place in a circular arena, with lion chasing man. Sgall [16] considered the turn-based version played in the non-negative quadrant of the plane. He showed that a lion starting at \((x_0, y_0)\) captures a man starting at \((x'_0, y'_0)\) if and only if \(x'_0 < x_0\) and \(y'_0 < y_0\). Kopparty and Ravishankar [14] generalized this strategy to obtain the spheres strategy for \(\mathbb{R}^n\), where the evader \(e\) is caught by pursuers \(p_1, p_2, \ldots, p_\ell\) if and only if \(e\) starts in the interior of the convex hull of the pursuers. During pursuit, \(p_i\) guards an expanding circular region \(B_i^t\) so that after step \(t\), the pursuer has either captured the evader, or the area of the region \(B_i^t\) he guards is larger by a constant amount than the area of \(B_i^{t-1}\).

Isler et. al. [11] adapted this strategy for one pursuer in a simply connected polygon. We opt for the name “lion’s strategy” for this scenario because the ball \(B(x, r) = \{y \in P \mid d(x, y) \leq r\}\) usually does not look like a traditional sphere due to the boundary edges. Lion’s strategy proceeds as follows. Suppose that the players start at points \(p^0\) and \(e^0\). The pursuer fixes a point \(z \in P\) chosen so that \(p^0\) is on the (unique) shortest path from \(z\) to \(e^0\). The pursuer movement in round \(t \geq 1\) is described in Algorithm 1.

Algorithm 1 Lion’s Strategy

Given the positions \(e^{t-1}, e^t, p^t\), such that \(p^t\) is on the shortest path \(\Pi^{t-1}\) from \(z\) to \(e^{t-1}\). To compute \(p^{t+1}\):

- Let \(\Pi^t\) be the shortest path between \(z\) and \(e^t\).
- Choose \(p^{t+1}\) on \(\Pi^t\) such that \(d(p^{t+1}, e^t)\) is minimized, subject to \(d(p^t, p^{t+1}) \leq 1\).

We observe that a pursuer using lion’s strategy actually guards a monotonically increasing subset of the environment, namely \(B(z, d^t)\) where \(d^t = d(z, p^t)\). In other words, once a pursuer guards an
area, he also prevents recontamination of that region for the remainder of the game. The validity of Lion’s Strategy in a simply connected environment is proven in [11]. There are two points that are addressed. First, that the pursuer can move from \( \Pi^{t-1} \) to \( \Pi^t \), and second that the distance between pursuer and evader decreases with this move.

**Lemma 2.1 ([11]).** Lion’s strategy is a winning strategy for a single pursuer in a simply connected polygonal environment.

### 3 Projections and guarding

Let \( P \) be a polygonal environment with obstacles, and let \( Q \subset P \) be a sub-environment. In this section, we define a broad class of projection functions from \( P \) onto \( Q \). These projection functions play a crucial role in our pursuit strategies. Such projections for pursuit-evasion appear in recent results of Bhadauria et. al. [4], and we draw heavily from their viewpoint. After giving a general definition of a projection, we will prove two pursuit results: (1) a pursuer can evict the evader from \( Q \) by chasing (and capturing) the evader’s projection; and (2) a pursuer who is collocated with the evader’s projection onto \( Q \) can keep the evader from re-entering \( Q \).

**Definition 3.1.** A \((P, Q)\)-projection is a function \( \pi : P \to Q \) such that (1) if \( x \in Q \) then \( \pi(x) = x \), and (2) for all \( x, y \in P \), we have \( d_Q(\pi(x), \pi(y)) \leq d_P(x, y) \).

In other words, the \((P, Q)\)-projection \( \pi \) is the identity map on \( Q \), and the mapping never increases the distances between points. Taking \( x, y \in Q \), we find that \( d_Q(x, y) \leq d_P(x, y) \), which means that \( Q \) contains a shortest \((x, y)\)-path (also known as a geodesic). In other words, if there is a projection function \( \pi : P \to Q \) then \( Q \) must be geodesically convex.

We now give a few examples of projections. First, consider a simply connected polygonal environment \( P \). Let \( Q \subset P \) be a convex polygon. If \( \partial P \cap \partial Q = \emptyset \), then \( P \setminus Q \) is a polygonal environment with a single obstacle; otherwise the components of \( P \setminus Q \) are all simply connected. Define \( \rho : P \to Q \) to be the mapping that takes \( x \in P \) to the unique point \( y \in Q \) such that \( d_P(x, y) = \min_{z \in Q} d(x, z) \). Note that the convexity of \( Q \) ensures that this mapping is well-defined. Moreover, it is easy to see that \( \rho \) is a \((P, Q)\)-projection. Next, suppose that \( P \) is not simply connected. Let \( Q \) be a sub-environment with a convex boundary such that every obstacle of \( P \) is also contained in \( Q \). In this case, the same function \( \rho \) is still a \((P, Q)\)-projection. More broadly, when the sub-environment \( Q \) is such that every \( x \in P \) has a unique closest point in \( Q \), we introduce the term metric projection.
Figure 3.1: Some paths in polygonal environments. There is a metric \((P, \Pi_i)\)-projection only for \(i = 1, 2\). The paths \(\Pi_1, \Pi_2, \Pi_4\) are minimal in their environments. The path \(\Pi_3\) is only minimal for the sub-environment \(P'_3\). There is no metric projection from \(P''_4\) onto \(\Pi_4\).

**Definition 3.2.** Suppose that \(Q \subset P\) is such that for every \(x \in P\) there is a unique point \(y \in Q\) achieving \(d_P(x, y) = \min_{z \in Q} d(x, z)\). The projection \(\rho\) induced by this mapping is the **metric projection** from \(P\) onto \(Q\).

Clearly, the metric projection \(\rho\) is a \((P, Q)\)-projection. However, there are many instances in which there is no well-defined metric projection because there are multiple nearest points; see Figure 3.1 for some examples. Bhadauria et al. [4] introduce a second type of projection that is less intuitive, but applicable to a broader class of environments, including those with obstacles.

**Definition 3.3 (Minimal Path [4]).** Suppose that \(\Pi\) is a path in environment \(P\) dividing it into two sub-environments, and \(P_e\) is the sub-environment containing the evader \(e\). We say that \(\Pi\) is **minimal** with respect to \(P_e\) if, for all points \(x, z \in \Pi\) and \(y \in (P_e \setminus \Pi)\), we have \(d_{\Pi}(x, z) \leq d_{P_e}(x, y) + d_{P_e}(y, z)\).

For example, a shortest path between \(u, v \in P\) is always minimal with respect to \(P\). In Figure 3.1 the paths \(\Pi_1, \Pi_2, \Pi_4\) are minimal with respect to the whole environment. There is no metric projection onto the minimal path \(\Pi_4\): the obstacle results in the existence of two distinct points in \(\Pi_4\) attaining the minimum distance to point \(x\). We use the more robust **path projection** to deal with such an environment.

**Definition 3.4 (Path Projection [4]).** Let \(u, v \in \partial P\) and let \(\Pi_{u,v}\) be a minimal path in \(P_e\) between them. For \(x \in P_e\) with \(d(u, x) \leq d(u, v)\), define \(\phi(x)\) to be the point on \(\Pi_{u,v}\) at distance \(d(u, x)\) from \(u\). When \(d(u, x) > d(u, v)\) define \(\phi(x) = v\). The mapping \(\phi\) is called the **path projection** of \(P_e\) onto \(\Pi_{u,v}\). Setting \(Q\) to be the complement of \(P_e\), we extend this to a projection \(\phi : P \to Q\) by setting \(\phi(x) = x\) for \(x \in Q\).

Note that we restrict \(u, v\) to lie on the boundary of \(P\). The proof that this mapping is a \((P, Q)\)-projection is given in [4]. Considering Figure 3.1 we see that \(\Pi_3\) is a minimal path in \(P'_3\), but not in \(P''_3\).
Meanwhile, there are path projections from each of \( P'_1, P''_4 \) to \( \Pi_4 \). Restricting a pursuer’s movement to the evader’s projection is a key component of the pursuit strategies developed in the following sections. Once a pursuer captures the evader’s projection, the pursuer can maintain that colocation after every pursuer turn thereafter. Indeed, if the evader moves from \( e \) to \( e' \) then \( 1 \geq d(e, e') \geq d(\pi(e), \pi(e')) \).

We state this as a useful lemma.

**Lemma 3.5.** Let \( Q \subset P \) be a sub environment with a projection \( \pi : P \to Q \). Suppose that the pursuer starts at \( \pi(e) \). After the evader moves from \( e \) to \( e' \), the pursuer can move from \( \pi(e) \) to \( \pi(e') \). □

During our pursuit, we frequently divide the environment into intersecting regions. The following lemma explains how to patch together projections of overlapping subregions.

**Lemma 3.6.** Let \( P \) be a polygonal environment with sub-environments \( Q, P_1, P_2 \) such that \( P = P_1 \cup P_2 \) and \( Q \subset P_1 \cap P_2 \). Suppose that for \( i = 1, 2 \), we have a \((P_i, Q)\)-projection \( \pi_i \) with \( \pi_1(x) = \pi_2(x) \) for every \( x \in P_i \cap P_2 \). Then the function

\[
\pi(x) = \begin{cases} 
\pi_1(x) & x \in Q \\
\pi_2(x) & x \in P_2 \setminus P_1
\end{cases}
\]

is a projection from \( P \) to \( Q \).

**Proof.** For points \( x \in P_1 \) and \( y \in P = P_1 \cup P_2 \), let \( \Pi \) represent a minimal path from \( x \) to \( y \) in \( P \). Partition \( \Pi \) into a finite collection of subpaths \( \Pi = \{\Pi_1, \Pi_2, \ldots, \Pi_k\} \) where the odd indexed paths are in \( P_1 \) and the even indexed paths are in \( P_2 \). Let \( u_{i-1}, u_i \) be the endpoints of \( \Pi_i \), so that \( u_0 = x \) and \( u_k = y \). We consider the case that \( k \) is even; the proof for odd \( k \) is similar. We have

\[
d_Q(\pi(x), \pi(y)) \leq d_Q(\pi_1(u_0), \pi_1(u_1)) + d_Q(\pi_2(u_1), \pi_2(u_2)) + \cdots + d_Q(\pi_k(u_{k-1}), \pi_k(u_k))
\]

\[
\leq d_{P_1}(u_0, u_1) + d_{P_2}(u_1, u_2) + \cdots + d_{P_k}(u_{k-1}, u_k) = d_P(x, y) \quad \Box
\]

By induction, the analogous result holds for any finite collection of projections, with pairwise agreement on common intersections.

**Corollary 3.7.** Let \( P = P_1 \cup P_2 \cup \cdots P_k \) be a polygonal environment with a sub-environment \( P_0 \) such that for all \( 1 \leq i \leq k \), we have \( P_0 \subset P_i \). Suppose that there exists a family of \((P_i, P_0)\)-projections \( \pi_i \) such that for any \( 1 \leq i, j \leq k \), we have \( \pi_i(x) = \pi_j(x) \) for every \( x \in P_i \cap P_j \). Then the piecewise function

\[
\pi(x) = \begin{cases} 
x & x \in P_0 \\
\pi_i(x) & x \in P_i \setminus (P_0 \cup \cdots \cup P_{i-1}), 1 \leq i \leq k
\end{cases}
\]

is a projection from \( P \) to \( P_0 \). □
Figure 3.2: Minimal loops in a polygonal environment. For each loop, \( \Pi_1 \) is a minimal path in \( P_1 \) and \( \Pi_2 \) is a minimal path in \( P_2 \). In (b), the sub-environment \( P_1 \) is disconnected. In (c), we have \( P_1 = \Pi_1 \) since the path \( \Pi_1 \) is part of the external boundary of the environment.

We use Lemma 3.6 to extend our definition of projection to apply to a loop \( \Lambda \) (that is, closed path) that intersects the boundary of \( P \) in at least two points \( u, v \).

**Definition 3.8 (Loop, Minimal Loop).** Let \( u, v \in \partial P \). A loop \( \Lambda \) consists of two internally disjoint \((u, v)\)-paths \( \Pi_1, \Pi_2 \). These paths divide the environment \( P \) into three sub-environments: the interior \( Q \) between the two paths, and exterior environments \( P_1, P_2 \), bounded by \( \Pi_1, \Pi_2 \) respectively. The loop \( \Lambda \) is \((u, v)\)-minimal when \( \Pi_1 \) is a minimal path for \( P_1 \) and \( \Pi_1 \) is a minimal path for \( P_2 \).

**Lemma 3.9.** Let \( \Lambda \) be a \((u, v)\)-minimal loop in polygonal environment \( P \) where \( u, v \in \partial P \). Let \( Q \) be the sub-environment bounded by the loop \( \Lambda \). There is a projection \( \phi : P \to Q \).

**Proof.** For \( i = 1, 2 \), let \( P_i \) be the sub-environment of \( P \) bounded by the minimal path \( \Pi_i \), and let \( \phi_i : P_i \cup Q \to Q \) be the path projection. By Lemma 3.6, we can combine these projections to get a projection \( \phi : P \to Q \).

In the remainder of this section, we explain how to use projections to evict the evader from a region, and then guard that region thereafter (that is, prevent the evader from re-entering). First, we show that if the pursuer is co-located with the evader’s projection onto a minimal path, then the pursuer can prevent the evader from crossing it.

**Lemma 3.10 (Guarding Lemma).** Let \( Q \subset P \) with the projection \( \pi : P \to Q \). Suppose that \( p^1 = \pi(e^0) \). Then the pursuer can maintain \( p^{i+1} = \pi(e^i) \) for \( i \geq 1 \). Furthermore, if the evader moves so that a shortest path from \( e^{t-1} \) to \( e^t \) intersects \( Q \), then \( d_P(p^t, e^t) \leq d_P(e^{t-1}, e^t) \leq 1 \), so the pursuer can capture the evader at time \( t + 1 \).
Proof. The first claim follows easily by induction, since
\[ d_Q(p^i, \pi(e^i)) = d_Q(\pi(e^{i-1}), \pi(e^i)) \leq d_P(e^{i-1}, e^i) \leq 1, \]
meaning that the pursuer can remain on the projection of the evader. For the second claim, suppose that the shortest path from \( e^{t-1} \) to \( e^t \) includes the point \( x \in Q \). Then
\[ d_P(p^t, e^t) \leq d_Q(p^t, x) + d_P(x, e^t) \leq d_P(e^{t-1}, x) + d_P(x, e^t) = d_P(e^{t-1}, e^t) \leq 1. \]
Therefore the pursuer can move to \( e^t \) on his next turn.

When a pursuer \( p \) follows the strategy in Lemma 3.10, we say that \( p \) guards \( Q \) with respect to the projection \( \pi \). Note that we must have \( p^t = \pi(e^{t-1}) \) before the pursuer can start to guard \( Q \). After that, the pursuer never leaves the region he guards. If the evader travels into \( Q \) (or across \( Q \) in a single turn), then the pursuer apprehends the evader on his next turn. One example of this last kind of capture is given in the following corollary.

**Corollary 3.11.** Suppose that pursuer \( p \) is guarding subregion \( Q \subset P \) with respect to \( \pi : P \to Q \). Suppose further that \( P \setminus Q \) is the disjoint union of simply connected components \( R_1, R_2, \ldots, R_s \). If the evader moves from \( R_i \) to \( R_j \), where \( i \neq j \), then the pursuer can catch him on his next turn.

Our final pursuit lemma asserts that a single pursuer can evict the evader from any simply connected subregion with a valid projection function. We refer to this as clearing the subregion.

**Lemma 3.12 (Clearing Lemma).** Let \( Q \) be a simply connected subenvironment of \( P \) with projection \( \pi : P \to Q \). In a finite number of moves, one pursuer can either capture the evader or guard \( Q \).

**Proof.** The pursuer executes lion’s strategy in the simply connected region \( Q \), chasing after \( \pi(e) \). By Lemma 2.1, a single pursuer can capture an evader in a simply connected region in finite time, say \( p^t = \pi(e^{t-1}) \). If \( e^{t-1} = \pi(e^{t-1}) \), then the evader is caught. Otherwise, the pursuer has attained position to guard \( Q \) from this time forward, as described by Lemma 3.10.

## 4 Environments with one obstacle

In this section, we prove Theorem 1.3. Two pursuers are always enough to catch an evader when there is a single obstacle in the environment. Indeed, \( p_1 \) can move to guard a shortest path from the obstacle to the boundary of \( P \). This makes the environment simply connected, so \( p_2 \) can catch \( e \) using lion’s
strategy. However, if the obstacle is small enough, one pursuer can actually catch the evader. We use projections to determine when an environment with a single obstacle is one-pursuer-win. Let $H$ denote the obstacle. The critical factor is the length of the boundary of the convex hull $J = \text{Hull}(H)$. Theorem 1.3 states that the environment is one-pursuer-win if and only if this boundary length is smaller than 2. The theorem follows directly from Lemmas 4.1 and 4.3 below.

**Lemma 4.1.** Suppose that $P$ is a polygonal environment with one obstacle $H$ whose convex hull $J = \text{Hull}(H)$ has perimeter $\ell > 2$. The evader has a winning strategy against a single pursuer.

**Proof.** For simplicity of exposition, we assume that $\text{Hull}(H)$ does not intersect the boundary of the environment. The proof can be adapted for this case, but we must redefine the convex hull of the obstacle in the natural way to handle the interaction with the external boundary. We first consider the case where the obstacle $H$ is convex. Let $\rho : P \rightarrow H$ be the metric projection. The mapping $\rho$ projects every point in $P \setminus H$ onto $\partial H$. We prove that the evader can always guarantee that $d(p^i, e^i) > d(\rho(p^i), e^i) = \ell/2 > 1$ for $i \geq 0$, which means that the pursuer can never catch the evader. The game begins when the pursuer chooses his location $p^0$ in $P$. The evader responds by placing himself at the unique point $e^0$ on $\partial H$ that is distance $\ell/2$ from $\rho(p^0)$. Proceeding by induction, assume that $d(\rho(p^i), e^i) = \ell/2$. The pursuer moves to a new location $p^{i+1}$ with $1 \geq d(p^i, p^{i+1}) \geq d(\rho(p^i), \rho(p^{i+1}))$. The evader responds by moving from $e^i$ to the unique point on the perimeter of $H$ that is distance $\ell/2$ from $\rho(p^{i+1})$. Of course, $d(e^i, e^{i+1}) = d(\rho(p^i), \rho(p^{i+1})) \leq d(p^i, p^{i+1}) \leq 1$, so the evader can attain this position, evading capture.

Next, we consider the case where the obstacle $H$ is not convex, but has convex hull $J = \text{Hull}(H)$ with perimeter $\ell > 2$. Analogous to the above case, the evader will restrict his movement to the perimeter $\partial J$ of the convex region $J$. As long as the pursuer does not enter $J$, the argument for convex obstacles shows that the evader can remain at distance $\ell/2$ from the pursuer projection. Meanwhile, entering $J$ (or more precisely, some component of $J \setminus H$) is worse for the pursuer than staying on the boundary of $J$. We make this more precise by using projections.

Let $\rho : P \rightarrow J$ be the metric projection. Next, we define projections for the areas in $J$. Let the vertices of $J$ be $v_0, v_1, \ldots, v_{k-1}$, indexed counterclockwise. Let $\Pi_i$ be the line segment joining $v_i, v_{i+1}$ (here the index is modulo $k$). Let $Q_i$ be the simply connected component of $J \setminus H$ whose boundary includes $\Pi_i$. (Note that $Q_i = \Pi_i$ when the segment between $v_i, v_{i+1}$ is part of obstacle $H$.) Let $\phi_i : Q_i \rightarrow \Pi_i$ be the path projection from $Q_i$ to $\Pi_i$, anchored at $v_i$. Finally, we define the piecewise
function \( f : P \to \partial J \) as
\[
\pi(x) = \begin{cases} 
  x & x \in \partial J, \\
  \rho(x) & x \in P \setminus J, \\
  \phi_i(x) & x \in Q_i \setminus \Pi_i.
\end{cases}
\]
This piecewise function is a projection from \( P \) to \( \partial J \) by Corollary 3.7. Suppose that \( p^t \in P \setminus J \) and \( p^{t+1} \in Q_i \). Let \( x \in \Pi_i \) be on a minimal \((p^t, p^{t+1})\)-path. Then
\[
1 \geq d(p^t, p^{t+1}) = d(p^t, x) + d(x, p^{t+1}) \geq d(\pi(p^t), x) + d(x, \pi(p^{t+1})) = d(\pi(p^t), \pi(p^{t+1})),
\]
where the last equality holds because \( x \) is on a shortest path from \( \pi(p^t) \) to \( \pi(p^{t+1}) \) in \( \partial J \). Therefore, the evader can move to the point at distance \( \ell/2 \) from \( \pi(p^t+1) \). The analogous argument holds when the pursuer moves from \( Q_i \) to \( P \setminus J \), or from \( Q_i \) to \( Q_j \).

We now turn to the pursuer-win situation. The proof of this result rests mainly on the following lemma, which shows that we can transition from guarding a line segment \( \Pi \) to lion’s strategy in such a way that the evader cannot cross \( \Pi \) without being caught.

**Lemma 4.2.** Let \( \Pi \) be a line segment in \( P \) that connects boundary points \( u, v \in \partial P \). Let \( P_e \) be the sub-environment of \( P \setminus \Pi \) containing the evader, and let \( Q = P \setminus P_e \). Let \( \rho : P_e \to Q \) be the metric projection. If the pursuer starts at \( p^0 = \rho(e^0) \) then in a single move, the pursuer can transition to a lion’s strategy, keeping the line segment \( \Pi \) within his guarded region.

**Proof.** Let \( \Lambda \) be the line through \( p^0 \) and \( e^0 \). Let \( a = d(e^1, \Lambda) \) and let
\[
b = \max \{ 2, d(p^0, u), d(p^0, v), \sqrt{a} \}.
\]
We use the coordinate system with \( p^0 = (0, 0) \) and where \( \Lambda \) is the \( x \)-axis, so that \( e^0 = (a_0, 0) \) and \( e^1 = (a, h) \) for some \( a_0, a \in \mathbb{R}^+ \) where \( (a - a_0)^2 + h^2 \leq 1 \). We can then safely replace \( \Pi \) by a line segment of length \( 2b \) (which extends outside the boundary of \( P \)) with endpoints \( u = (0 - b), v = (0, b) \). The relevant geometry is shown in Figure 4.3. We will show that in a single move, the pursuer can transition to a lion’s strategy that still guards \( \Pi \). In particular, we will find \( s \in \mathbb{R}^+ \) such that the pursuer can move onto the circle with center \( z = (-s, 0) \) and radius \( \sqrt{s^2 + b^2} \), which contains the segment \( \Pi \). We will see that choosing \( s = b^2/a \) is sufficient.

The center of our circle \( z = (-s, 0) \) and the pursuer location \( p^1 = (x, y) \) must satisfy
\[
x^2 + y^2 \leq 1, \quad (1)
\]
\[
(s + x)^2 + y^2 = s^2 + b^2, \quad (2)
\]
\[
y = \frac{h}{s + x}, \quad (3)
\]

\[11\]
The last equation is equivalent to \((s + x) = y(s + a)/h\). Using this value in equation (2) and setting \(s = b^4/a\) yields
\[
y^2 \leq h^2 \left( \frac{b^8/a^2 + b^2}{b^8/2a^2 + 2b^4 + a^2 + h^2} \right) < h^2 \leq 1.
\]
We have
\[
x = \frac{s + a}{h} y - s \leq \sqrt{s^2 + b^2} - s \leq s \left( 1 + \frac{b^2}{2s^2} \right) - s = \frac{b^2}{2s} = \frac{a}{2b^2} < 1,
\]
because we chose \(b\) so that \(a \leq b^2\). Therefore
\[
x^2 + y^2 \leq \frac{b^4}{4s^2} + \frac{h^2(s^2 + b^2)}{s^2 + 2as + a^2 + h^2} \leq \frac{b^4}{4s^2} + \frac{s^2 + b^2}{s^2 + 2as + a^2} < 1.
\]
In other words, this pair \((x, y)\) satisfies the final constraint equation (1). This also guarantees that \(x < a\) because \(y < h\) and equation (3) holds. In conclusion, the pursuer can take one step onto the line connecting \(z\) and \(e^1\) and immediately guard a disc that contains the line segment \(\Pi\).

We note that the progress that the pursuer makes during this transition depends upon how close the evader is to the obstacle. Indeed, we end use the point \(z\) as our center, which is at distance \(s = b^4/a\), so the capture time is inversely proportional to the evader’s initial distance from the obstacle. Given our restrictive definition of capture (colocation, as opposed to proximity), there is no way around this. This problem does not manifest itself in the two-cop strategy in Theorem 1.1 since the pursuers alternate their guarding and pursuit roles.

**Lemma 4.3.** Suppose that \(P\) is a polygonal environment with one obstacle \(H\) whose convex hull \(J = \text{Hull}(H)\) has perimeter \(\ell \leq 2\). Then \(P\) is one-pursuer-win.
Proof. Initially, the pursuer chooses his position $p^0$ to be some point on the boundary of $J = \text{Hull}(H)$. The evader then chooses his initial position $e^0$. On his first turn, the pursuer moves to the metric projection $p^1 = \rho(e^0)$ on $\partial J$. Draw the maximal line segment $\Pi \subset P$ through $p^1$ that is perpendicular to the segment joining $p^1$ and $e^0$. Let the endpoints of $\Pi$ be $u, v \in \partial P$. The pursuer currently guards $\Pi$. If the evader is not in $J$, then the area guarded by the pursuer contains the entire obstacle $H$, which means that $P_e$ is simply connected. If the evader is in $J$, then he is trapped in a simply connected area between $\Pi$ and $H$. In either case, Lemma 4.2 allows the pursuer to transition from guarding $\Pi$ to lion’s strategy in a simply connected environment. Using Lion’s Strategy, the pursuer catches the evader by Lemma 2.1.

5 The leapfrog strategy

In this section, we prove our main result, Theorem 1.1. We show that if an environment has a decomposition satisfying (L1) – (L4) then two pursuers can use the leapfrog strategy to capture the evader. Intuitively, this strategy works as follows. While the first pursuer guards a subregion $Q$, the second pursuer clears new territory and guards a larger subregion $Q'$ containing $Q$. At that point, the second pursuer switches into guarding mode, and the first pursuer leapfrogs over him to clear new territory. This process continues until the evader is caught. However, there is another more subtle way for the pursuers to make progress: if the evader “makes a mistake” by passing through $Q$, then the current guarding pursuer can immediately capture the evader in his responding move.

Figure 5.4 shows an example of an environment with a leapfrog partition $Q_0 \subset Q_1 \subset Q_2 = P$. Conditions (L2) and (L2) let us use the projection framework in Section 3 for region $Q_0$. Condition (L3) ensures that $\partial Q_{i+1} \cap \partial Q_i \neq \emptyset$. Indeed, these boundaries are both polygons, and if they are disjoint, then the closed path $\delta Q_{i+1}$ is a nontrivial loop in $Q_{i+1} \setminus Q_i$. Condition (L4) is included for efficiency: if we start with a larger family of nested regions, then we can simply ignore the subregions that do not touch additional obstacles.

Proof of Theorem 1.1. Note that the condition (L2) does not require that projections agree wherever the boundaries of the subregions intersect. The first order of business is to construct a new family of projections $\sigma_i : P \rightarrow Q_i$ for which this is the case. We define these projections recursively, starting at $k - 1$. We set $\sigma_{k-1} = \pi_{k-1}$ and then define $\sigma_i = \pi_i \circ \sigma_{i+1}$ for $k - 2 \geq i \geq 0$. The function $\sigma_i$ is a
Figure 5.4: A leapfrog partition \( Q_0 \subset Q_1 \subset Q_2 \) where \( Q_1 \setminus Q_0 \) is disconnected. The two components of \( \partial Q_0 \cap \partial Q_1 \) are thickly drawn. Pursuer \( p_1 \) is on the \( \partial Q_0 \)-projection of evader \( e \) (using the path projection rooted at \( u \)). If \( e \) moves from \( R_1 \) to \( R'_1 \) then he will be caught by \( p_1 \). Next, pursuer \( p_2 \) will move to clear \( R_1 \), which also clears \( Q_1 \).

projection: for \( x, y \in P \), we have

\[
d_{Q_0}(\sigma_0(x), \sigma_0(y)) = d_{Q_0}(\pi_0(\sigma_1(x)), \pi_0(\sigma_1(y))) \\
\leq d_{Q_1}(\sigma_1(x), \sigma_1(y)) = d_{Q_1}(\pi_1(\sigma_2(x)), \pi_1(\sigma_2(y))) \\
\leq \cdots \leq d_{Q_{k-1}}(\pi_{k-1}(x), \pi_{k-1}(y)) \leq d_P(x, y).
\]

By construction, these recursively defined projections satisfy the conditions of Corollary 3.7. Most importantly for us, if \( \sigma_{i+1}(x) \in \partial Q_i \cap \partial Q_{i+1} \) then \( \sigma_i(x) = \sigma_{i+1}(x) \). In other words, suppose that \( p_1 \) is using projection \( \sigma_i \) and \( p_2 \) is using projection \( \sigma_{i+1} \). If an evader projection is on the shared boundary \( \partial Q_i \cap \partial Q_{i+1} \), then both pursuers agree on its location. This continuity is crucial to the leapfrog strategy, since it allows \( p_1 \) to react when \( p_2 \) sees the evader projection cross through \( \partial Q_i \).

The leapfrog strategy proceeds as follows. First, \( p_1 \) clears \( Q_0 \) with respect to the projection \( \sigma_0 : P \to Q_0 \), as described in Lemma 3.12. Assume inductively that one pursuer, say \( p_1 \), currently guards \( Q_i \). Next, \( p_2 \) works to clear \( Q_{i+1} \) with respect to \( \sigma_{i+1} : P \to Q_{i+1} \). When \( Q_{i+1} \setminus Q_i \) is not connected, this clearing movement requires \( p_1 \) as well, see Figure 5.4. The guarding \( p_1 \) captures \( e \) whenever the evader moves between components of \( Q_{i+1} \setminus Q_i \). Indeed, if \( Q_{i+1} \setminus Q_i \) is not connected, then we must have \( \partial Q_i \cap \partial Q_{i+1} \neq \emptyset \). If a shortest path between \( \sigma_{i+1}(e^{t-1}) \) and \( \sigma_{i+1}(e^t) \) intersects \( Q_i \) (and therefore intersects \( \partial Q_i \cap \partial Q_{i+1} \)), then \( p_1 \) can immediately respond by the capture move \( p_1^{t+1} = e^t \) by Corollary 3.11. Indeed, \( \sigma_{i+1}(e) \) moves between components of \( Q_{i+1} \setminus Q_i \) if and only if \( e \) moves between components of \( Q_{i+1} \setminus Q_i \). By the construction of the projections \( \sigma_0, \ldots, \sigma_{k-1} \), the guarding \( p_1 \) is in position to capture the evader in response to this boundary crossing. This means that the evader cannot move between the components of \( Q_{i+1} \setminus Q_i \) without being captured.
Let us return to $p_2$’s attempt to clear $Q_{i+1}$. Let $R_1, R_2, \ldots, R_k$ be the simply connected components of $Q_{i+1} \setminus Q_i$, and say that $\sigma_{i+1}(e) \in R_1$. While $p_1$ guards $Q_i$, pursuer $p_2$ moves into $R_1$ and tries to clear this region. If the projection $\sigma_{i+1}(e)$ ever leaves $R_1$, then $p_1$ can immediately respond by capturing $e$ (because the evader’s projection moves through $\partial Q_i$, as described above). Otherwise, the projection $\sigma_{i+1}(e)$ always remains in $R_1$, so $p_2$ can capture this position by Lemma 3.12. At this point, $p_2 = \sigma_{i+1}(e)$. If $e = \sigma_{i+1}(e)$, then the evader is caught. Otherwise, $p_2 = \sigma_{i+1}(e) \in \partial Q_{i+1}$, and $p_2$ switches to guarding $Q_{i+1}$. This releases $p_1$ to start clearing $Q_{i+2}$. This leapfrogging continues until the evader is caught, which must occur when the pursuers finally control $Q_k \setminus Q_{k-1}$. □

Proof of Corollary 1.2 Since $Q_0$ is geodesically convex and $\partial Q_0$ contains two points $u, v \in \partial P$, the boundary $\partial Q_0$ is a minimal loop. By Lemma 3.9, there is a path projection $\pi_0 : P \rightarrow Q_0$. Likewise, $u, v \in \partial Q_i$ for every $0 \leq i \leq k$, so there is a path projection $\pi_i : P \rightarrow Q_i$. This is a family of projection functions required by (L2) in Theorem 1.1. □

We conclude this section by giving an upper bound on the time to capture of the leapfrog strategy. The leapfrog strategy repeatedly uses lion’s strategy in simply connected environments. Isler et. al. [11] prove that in a simply connected polygon $R$, lion’s strategy completes in time $O(m \cdot \text{diam}(R))$ where $m$ is the number of vertices of $R$ and $\text{diam}(R) = \max_{u,v} d_R(u,v)$. Therefore, lion’s strategy completes in time $O(n \cdot \text{diam}(P))$ for each of $Q_0$ and $Q_{i+1} \setminus Q_i$, $0 \leq i \leq k - 1$. Since $k \leq h + 1$ (where $h$ is the number of holes), the leapfrog strategy completes in time $O(n \cdot h \cdot \text{diam}(P))$

6 Strictly two-sweepable environments

In this section, we describe a class of polygons for which the the two-pursuer leapfrog strategy is successful. Specifically, we show that a strictly two-sweepable environment $P$ has a family of nested sub-environments $Q_0 \subset Q_1 \subset \cdots Q_k = P$ that satisfy the hypothesis of Corollary 1.2. Finally, we give an algorithm that determines if a given polygon $P$ is strictly two-sweepable, based on finding a specific path in the dual of $P$. This algorithm explicitly constructs the family $Q_0, Q_1, \ldots, Q_k$ for use in the leapfrog strategy.

In [6], Bose and Kreveld present a method for determining if a polygon can be monotonically swept by a line. We extend these ideas to a polygonal environment with obstacles. Intuitively speaking, the obstacles of these polygons are well-separated, so we can find our leapfrog partition using straight-line boundaries. We begin with two definitions.
Definition 6.1 (Sweepable Polygons [6]). A polygon $P$ is sweepable if a line $\ell$ can be swept continuously over $P$ such that each intersection of the line and $P$ is a convex set. We call such a line a sweep line of $P$. Polygon $P$ is strictly sweepable if there exists a sweep line such that no portion of $P$ is swept over more than once.

Definition 6.2 (Strictly Two-Sweepable Environments). A polygonal environment $P$ is said to be two-sweepable if a line can be swept continuously over $P$ such that each cross section of $P$ with respect to this line is the disjoint union of at most two convex sets. Environment $P$ is strictly two-sweepable if $P$ is two-sweepable and its boundary polygon $B_P$ is strictly sweepable. Equivalently, $P$ is strictly two-sweepable if there exists a sweep line $\ell$ such that no portion of $P$ is swept more than once and the intersection of $\partial P$ and each cross section of $P$, with respect to $\ell$, contains at most two points.

This brings us to the main result of this section.

Theorem 6.3. If the polygonal environment $P$ is strictly two-sweepable then $P$ is two-pursuer-win.

Proof. We describe how to construct a family of nested sub-environments satisfying the hypothesis of Corollary 1.2 using the movement of the sweep line. Let $P$ be a strictly two-sweepable polygonal environment with $n$ vertices $\{v_1, \ldots, v_n\}$ and sweep line $\ell$. Let $\ell_1, \ldots, \ell_n$ denote the positions of the sweep line $\ell$ intersecting each vertex of $P$. Here the vertex labels (and cross section labels) are ordered non-increasingly with respect to the order in which they are swept by $\ell$; the fact that $P$ is strictly two-sweepable ensures that such an ordering is well-defined.

Figure 6.5: A region partitioned into convex regions by a sweeping line. Left: the sweep lines $\ell_1, \ldots, \ell_n$. Right: the partition $\mathcal{P}$.

We construct our family of nested subregions. Let $\mathcal{H} = \{H_1, H_2, \ldots, H_h\}$ denote the set of obstacles in the environment $P$; as before, the obstacles are labeled with respect to the order in which are swept by $\ell$. Let $\{\lambda_1, \ldots, \lambda_h\} \subset \{\ell_1, \ldots, \ell_n\}$ denote the positions of the sweep line $\ell$ first intersecting each obstacle $H_1, H_2, \ldots, H_h$ as $\ell$ sweeps $P$. We also define $\lambda_0 = \ell_1$, $\lambda_{h+1} = \ell_n$, corresponding to the first and last boundary vertices encountered by $\ell$ during the sweep. Let $\mathcal{P} = \{P_0, P_1, \ldots, P_h\}$
denote the sub-environments of $P$, where $P_k$ is inscribed by $\lambda_k$ and $\lambda_{k+1}$. The family $\mathcal{P}$ is a partition of $P$ where each sub-environment $P_i$ is simply connected, as shown in Figure 6.5. Finally, construct our family $Q$ of nested sub-environments by taking inductively by taking $Q_0 = P_0$ and $Q_i = P_i \cup Q_{i-1}$ for $1 \leq i \leq h$. It is easy to check that the family $Q$ satisfies the hypothesis of Corollary 1.2.

6.1 Duality Algorithms for Finding Sweepable Polygons

We conclude with a practical discussion of how to determine if a given polygonal environment is strictly two-sweepable (and therefore two-pursuer win). The technique extends the duality-based method of Bose and Kreveld in [6] from simply connected polygons to polygonal environments containing obstacles. We summarize Bose and Kreveld’s results here and direct the reader to [6, Section 4] for a more thorough treatment of the topic.

Let $P$ be a simply connected polygon. To determine if $P$ is sweepable and/or strictly sweepable, Bose and Kreveld consider the movement of a proposed sweep line in the dual of $P$. We consider the duality transform that maps a given point $p = (a, b)$ to the line $D_p = \{(x, y) : ax - b\}$ with slope $a$ and $y$-intercept $-b$, and maps each line $L = \{(x, y) : y = mx + c\}$ to the point $(m, -c)$. Each edge of $P$ is mapped to the face of the dual inscribed by the two lines corresponding to its endpoints; we call the pair of faces of the dual corresponding to an edge in the primal a double wedge. We call the lower and upper envelopes of all faces in the dual the start face and end face, respectively.

The movement of a line swept continuously across $P$ dualizes to a path from the start face to the end face in the dual arrangement. Each cross section of $P$ defined by the line $\ell$ is mapped to a point in the dual in the intersection of all double wedges corresponding to the edges intersected by $\ell$. Rotation of a line past a vertical position is represented by the path “jumping” from one unbounded face of the dual arrangement to the opposite unbounded face in the double wedge corresponding to the edges of $P$ intersected by the vertical line. Each such jump corresponds to a change in orientation of the line as it sweeps $P$. Crucially, as the line sweeps $P$ its trajectory in the dual is restricted to an even number of such jumps to ensure that the orientation of the line is maintained.

Recall that $P$ is sweepable if and only if a line can be continuously across $P$ such that each cross section of $P$ is a convex set. Equivalently, $P$ is sweepable if and only if there exists sweep line intersecting at most two edges of $P$ at a time. This sweep line corresponds to a path in the dual arrangement that does not traverse the intersection of more than two double wedges. We call a face in the dual arrangement in which at least three double wedges overlap a forbidden face. Therefore, $P$
is sweepable if and only if there is a path from the start face to the end face in the dual arrangement avoiding all forbidden faces. We call such a path in the dual arrangement a sweep path. Both the dual arrangement of \( P \) and the set of all forbidden faces can be constructed in \( O(n^2) \) time (cf. [8, Chapter 8]). Performing depth-first search on the faces of the dual arrangement also takes \( O(n^2) \) time; indeed, the graph induced by the dual arrangement is planar and contains \( O(n^2) \) nodes and arcs.

A similar process can be used to determine if \( P \) is strictly sweepable. We must take care when sweeping reflex vertices, where the interior angle is greater than \( \pi \) radians. In this case, we add additional vertices to \( \partial P \) by temporarily extending the edges at reflex vertices into lines, and then adding a new vertex at each intersection of these lines with the boundary. We denote the resulting extended polygon as \( P' \). The following lemma from [6] characterizes when a polygon \( P \) is strictly sweepable using the extension \( P' \).

\textbf{Lemma 6.4 ([6, Lemma 4]).} Let \( P \) be a simple polygon, and let \( P' \) be its extended polygon. \( P \) is strictly sweepable if and only if \( P' \) admits a sweep line that traverses each vertex exactly once.

Algorithm 2 determines whether a polygon is strictly sweepable and identifies the corresponding sweep line (if one exists). Extending \( P \) to \( P' \) can be performed in \( O(n^2) \) time using a brute-force algorithm. The extended polygon \( P' \) contains \( m = O(n) \) vertices, so constructing the dual arrangement and performing depth-first search on the faces of the dual arrangement can be performed in \( O(m^2) = O(n^2) \) time.

We conclude by describing how to extend Algorithm 2 to an algorithm for identifying strictly two-sweepable environments. Let \( P \) be a polygonal environment containing obstacles and let \( B = \partial P \) be its boundary polygon. Let \( P' \) and \( B' \) be the extensions of \( P \) and \( B \), respectively, obtained by adding
Algorithm 2 Strictly Sweepable Path Search [6]

Given polygon $P$.
Extend all reflex vertices to obtain $P'$ with $m$ vertices
Compute dual arrangement $D_{P'}$ and identify all forbidden faces of $D_{P'}$.
Apply depth first search on the faces of $D_{P'}$ to find the sweep path in $D_{P'}$ crossing the fewest lines.
If this path crosses exactly $m$ lines then $P$ is strictly sweepable.

![Image of Algorithm 2](image)

Figure 6.7: Determining if an environment is strictly two-sweepable. Extend all reflexive vertices to obtain $P'$ (left) and dualize it (center), taking forbidden faces to be those in which more than four double wedges overlap. Overlay the forbidden faces from the dual of $B$. The valid path shown establishes that $P$ is strictly two-sweepable.

vertices by extending edges at all reflex vertices of $P$ and $B$. The environment $P$ is strictly two-sweepable if (1) a line can be swept across $P$ such that the cross sections of $P$ with respect to the line consist of at most two disjoint convex sets, (2) the cross sections of $B$ are convex, and (3) no point of $P$ is swept more than once. In order to decide whether $P$ is strictly two-sweepable, we must determine (a) if the dual arrangement of $P'$ admits a sweep-path that avoids all intersections of more than four double wedges, and (b) if the dual arrangement of $B'$ admits a sweep-path avoiding all intersections of more than two double wedges. This is summarized in Algorithm 3.

Algorithm 3 Strictly Two-Sweepable Path Search

Given an environment $P$ with boundary polygon $B = \partial P$.
Extend all reflex vertices to obtain $P'$ (with $m$ vertices) and $B'$.
Compute dual arrangements $D_{P'}$ and $D_{B'}$.
Identify all forbidden faces of $D_{P'}$.
Identify all forbidden faces of $D_{B'}$ and overlay on $D_{P'}$.
Apply depth first search on the faces of $D_{P'}$ to find a path avoiding all forbidden faces of $D_{P'}$ and $D_{B'}$ crossing the fewest lines.
If this path crosses exactly $m$ lines then $P$ is strictly two-sweepable.
7 Conclusion

We have characterized when one pursuer can capture an evader is an environment with a single obstacle. An immediate question that remains to be answered is: Under what conditions can one pursuer win in an environment with multiple obstacles? For example, our proof no longer holds in the case of two obstacles $H_1, H_2$, even when $\text{Hull}(H_1 \cup H_2) \leq 2$. Indeed, we could have a long zig-zagging alleyway between the obstacles. This would allow the pursuer to sit in the alley, and force the pursuer to give up his guarding position of the convex hull. To forbid such a pathological environment, it would be reasonable to enforce a minimum feature size (cf. [13]), meaning that no two vertices are within unit distance of one another. This simplifying assumption should make the two-obstacle case tractable.

The main open question regarding the lion and man game in polygonal environments is to fully characterize environments that are two-pursuer-win. In this work, our focus has been to give a characterization for environments in which a leapfrogging strategy is effective. Theorem 1.1 gives a very general description of the required family of nested subregions. The dual polygon algorithm in Section 6 identifies one such family, namely strictly two-sweepable environments. It would be interesting to develop an algorithm that can detect when an environment has a leapfrog decomposition, or at least construct other types of leapfrog decompositions.

8 Acknowledgments

This work was supported in part by the Institute for Mathematics and its Applications and in part by NSF Grant DMS-1156701. Volkan Isler was supported in part by NSF Grant IIS-0917676.

References

[1] M. Aigner and M. Fromme. A game of cops and robbers. Discrete Appl. Math., 8:1–12, 1983.

[2] L. Alonso, A. S. Goldstein, and E. M. Reingold. Lion and man: Upper and lower bounds. INFORMS Journal on Computing, 4(4):447, 1992.

[3] B. Alspach. Sweeping and searching in graphs: a brief survey. Mathematische, 59:5–37, 2006.

[4] D. Bhadauria, K. Klein, V. Isler, and S. Suri. Capturing an evader in polygonal environments with obstacles: The full visibility case. International Journal of Robotics Research, 31(10):1176–1189, 2012.
[5] A. Bonato and R. Nowakowski. *The Game of Cops and Robbers on Graphs*. American Mathematical Society, 2011.

[6] P. Bose and M. van Kreveld. Generalizing monotonicity: On recognizing special classes of polygons and polyhedra. *International Journal of Computational Geometry and Applications*, 15(6):591–608, 2005.

[7] T. H. Chung, G. A. Hollinger, and V. Isler. Search and pursuit-evasion in mobile robotics: a survey. *Autonomous Robots*, 31:299–316, 2011.

[8] M. De Berg, M. Van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational geometry*. Springer, 1997.

[9] L. J. Guibas, J.-C. Latombe, S. M. Lavalle, D. Lin, and R. Motwani. A visibility-based pursuit-evasion problem. *International Journal of Computational Geometry and Applications*, 9:471–494, 1996.

[10] G. Hahn. Cops, robbers and graphs. *Tatra Mountain Mathematical Publications*, 36:163–176, 2007.

[11] V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion in a polygonal environment. *IEEE Transactions on Robotics*, 5(21):864–875, 2005.

[12] V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion with local visibility. *SIAM Journal on Discrete Mathematics*, 1(20):26–41, 2006.

[13] K. Klein and S. Suri. Catch me if you can: Pursuit and capture in polygonal environments with obstacles. In *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence*, pages 2010–2016, 2012.

[14] S. Kopparty and C. V. Ravishankar. A framework for pursuit evasion games in $\mathbb{R}^n$. *Inf. Process. Lett.*, 96:114–122, 2005.

[15] J. E. Littlewood. *Littlewood’s Miscellany*. Cambridge University Press, 1986.

[16] J. Sgall. A solution to David Gale’s lion and man problem. *Theoretical Comp. Sci.*, 259(1–2):663–670, 2001.