

Pursuit Evasion on Polyhedral Surfaces

Kyle Klein · Subhash Suri

Abstract We consider the following variant of a classical pursuit-evasion problem: *how many pursuers are needed to capture a single (adversarial) evader on the surface of a 3-dimensional polyhedral body?* The players move on the closed polyhedral surface, have the same maximum speed, and are always aware of each others' current positions. This generalizes the classical lion-and-the-man game, originally proposed by Rado [16], in which the players are restricted to a two-dimensional circular arena. The extension to a polyhedral surface is both theoretically interesting and practically motivated by applications in robotics where the physical environment is often approximated as a polyhedral surface. We analyze the game under the discrete-time model, where the players take alternate turns, however, by choosing an appropriately small time step $t > 0$, one can approximate the continuous time setting to an arbitrary level of accuracy. Our main result is that 4 pursuers always suffice (upper bound), and that 3 are sometimes necessary (lower bound), for catching an adversarial evader on any polyhedral surface with genus zero. Generalizing this bound to surfaces of genus g , we prove the sufficiency of $(4g + 4)$ pursuers. Finally, we show that 4 pursuers also suffice under the “weighted region” constraints where the movement costs through different regions of the (genus zero) surface have (different) multiplicative weights.

Keywords Lion and man game, polyhedral surface, shortest path, weighted surfaces

1 Introduction

Pursuit-evasion problems serve as a mathematical abstraction for a number of applications that involve one group (pursuers) attempting to track down members of another group (evaders). Many such games with colorful names including Cops-and-Robbers, Hunter-and-Rabbit, Homicidal Chauffeur, and Princess-and-Monster have been studied in the literature [1, 4, 6, 11]. We are inspired by the oldest such problem, the so-called *lion-and-the-man game*, in which a lion and a man are enclosed in a circular arena, both able to move continuously with the same maximum speed, and able to react instantaneously to each other's motion. Can the lion capture the man? For many years, it was believed that the following simple strategy guarantees a win for the lion in finite time: start at the center of the arena and continuously move toward the man along the radial line. This was proved false by Besicovitch who showed that the man can in fact evade the lion forever [16]: in Besicovitch's strategy, the lion can get arbitrarily close to the man but never quite reach it. This impossibility proof can be circumvented by either allowing the lion a fixed non-zero capture radius $r > 0$, or playing the game in discrete-time (alternating moves).

In this paper, we investigate the pursuit-evasion problem played on the (closed) surface of a 3-dimensional polyhedron. Multiple pursuers (lions) attempt to capture an adversarial evader (man), with all players constrained to remain on the polyhedral surface, and all able to move equally fast. In this setting, how many pursuers are needed to capture the evader in finite time? We study the problem in

A preliminary version of this paper was presented at the 24th International Symposium on Algorithms and Computation (ISAAC '13) [13].

the discrete time model: this avoids the intractable problem of having to compute players’ moves and reactions *instantaneously*, and also allows approximation of the continuous time setting to an arbitrary level of accuracy by choosing an appropriately small time step $t > 0$. On the practical side, the problem of pursuit on a polyhedral surface is well-motivated because many robotics applications involve searching or tracking on “terrain-like” surfaces. On the theoretical side, the problem is interesting because the surface acts as an “intrinsic” obstacle, introducing non-linearity in the behavior of shortest paths. For instance, *although the genus zero polyhedral surface is topologically equivalent to a disk, the game has a distinctly different character and outcome than its planar counterpart (circular arena)*. In particular, it is known that a single pursuer can always win the discrete-time lion-and-the-man game in the plane (an easy corollary of [22]). Therefore, one may hope that an appropriate topological extension of the “follow the shortest path towards the evader” strategy will also succeed on the polyhedral surface. However, we show that this is not possible, and provide a constructive lower bound that *at least 3 pursuers* are needed in the worst-case for successful capture on a polyhedral surface. Intuitively, the problem is caused by the discontinuity in mapping “straight line” shortest paths in the unobstructed planar arena to geodesics on the polyhedral surface; in the unobstructed plane, a small move by the evader only causes a small (local) change in the straight line connecting pursuer and the evader, but on the polyhedral surface, the geodesic can jump discontinuously.

Complementing our lower bound, we show that 4 pursuers always suffice on any polyhedral surface of genus zero. Specifically, we present a strategy for the pursuers that always leads to capture of the evader in $O(\Delta_S(n^2 \log n + \log \Delta_S))$ time steps, where n is the number of vertices of the polyhedral surface S and Δ_S is its diameter (the maximum shortest path distance between any two points). We then generalize our result to surfaces of non-zero genus and prove that $(4 + 4g)$ pursuers can always capture an evader on the surface of any genus g polyhedron. Our technique for analyzing pursuit evasion on polyhedral surfaces appears to be quite general, and likely to find application in other settings. As one example, we consider pursuit evasion under the “weighted region” model of shortest paths, where non-negative weights dictate the per-unit cost of travel through different regions of the surface.

Related Work

In the discrete-time model, a single pursuer can capture the evader in a simply-connected polygon [10], while 3 pursuers are both necessary and sufficient for polygonal environments with multiple holes (obstacles) [5]. In a visibility-sensing model, where pursuers can localize the evader only when the latter is in direct line of sight, the number of pursuers is $O(\sqrt{h} + \log n)$ for n -vertex environment with h holes [12].

There exists an extensive literature on pursuit-evasion in 3-dimensional environments and surfaces, but no result appears to be known on the number of pursuers necessary for capture. Instead, the prior research has focussed on heuristics approaches for capture [14], classification of environments where capture is achievable [3], or on game-theoretical questions [15, 18].

The most relevant work to our research is the cops-and-robbers games in graph theory, where Aigner and Fromme have shown that 3 cops always suffice against a robber in any planar graph [1], and $\lfloor 3g/2 + 3 \rfloor$ cops are necessary for graphs of genus g [21]. However, the continuous-space of polyhedral surfaces requires very different set of techniques from those used for graphs.

2 Preliminaries and the Lower Bound

The geometric environment for our pursuit-evasion problem is the (closed) surface of a 3-dimensional polyhedron S . We assume that S has n vertices, and therefore $O(n)$ faces and edges. Without loss of generality, we assume that each face is a triangle, which is easily achieved by triangulating the faces with four or more sides. We use the notation p_1, p_2, \dots to denote the group of pursuers who wish to track and capture a single (adversarial) evader e . Slightly abusing the notation, we also use e and p_i , respectively, for the current location of the evader and the i th pursuer.

We make the standard assumption about the game: all the players know the environment (the surface of the polyhedron S), each player knows the current positions of all the other players, all players have identical maximum speed, and the game is played in the discrete-time alternating turn model. By an appropriate scaling of the environment, we assume that the maximum speed of the players is 1, meaning

that on its turn a player can move to any position within *geodesic distance* one of its current location on the surface. On their turn, all the pursuers move simultaneously. The pursuers win the game if, on their turn, some p_i reaches the current position of the evader and the evader wins if it can avoid capture indefinitely.

We use the notation $P_{a,b}$ for a shortest path between two points a and b on the surface S , and $d(a,b)$ for the length of this path. (In general, the path $P_{a,b}$ is not unique, but its length is.) The path $P_{a,b}$ is piece-wise linear and its vertices lie on the edges or vertices of the surfaces. Throughout, we will use the terms *vertices* and *edges* to refer to the graph of the polyhedral surface, and *points* and *arcs* to refer to the geometric objects embedded on the surface such as a path. We explain specific properties of these shortest paths that are used in our analysis in Section 3.3. The following theorem establishes the lower bound for our pursuit game.

Theorem 1 *In the worst-case at least three pursuers are required to capture an evader on the surface of a polyhedron.*

Proof We start with a dodecahedron D , all of whose edges have length 1, as shown in Fig. 1(a). Our polyhedron S is constructed by extending each face of D *orthogonally* (to the face) into a “tower” of height $\Delta_D + 1$, where Δ_D is geodesic diameter of the dodecahedron; see Fig. 1(b). The polyhedron S has 12 such towers, one for each of the 12 pentagonal faces of D . The “walls” of these towers meet along the edges of D , forming the skeleton graph, which we denote $G(D)$, as shown in Fig. 1(c). We now argue that an evader can indefinitely avoid capture from two pursuers on the surface of this polyhedron.

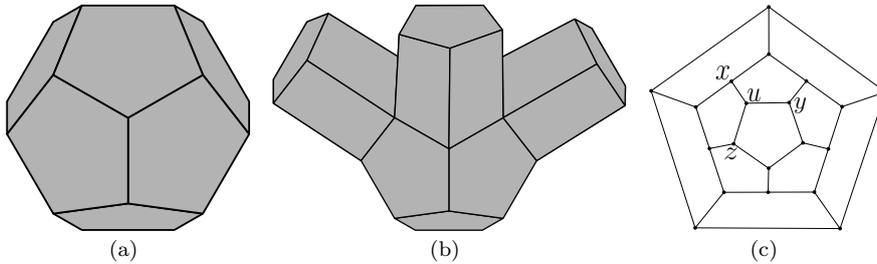


Fig. 1 A dodecahedron (a); partial construction with three faces orthogonally extended (b); and the skeleton graph (c).

Suppose there are only two pursuers, p_1 and p_2 . Initially, they choose their locations on S , and then the evader picks its initial position at a vertex of $G(D)$ to satisfy $d(p_i, e) > 1$, for $i = 1, 2$. (It is easy to see that this is possible.) Our proof shows that regardless of the pursuers’ strategies, the evader can indefinitely maintain this distance condition (after its move) by always moving among the vertices of $G(D)$, and thus evade capture forever.

The key observation is that the evader’s choice to remain on the skeleton graph $G(D)$ means that pursuers gain no advantage from positions not on $G(D)$. In particular, any pursuer located on the top face of a tower is not an immediate threat to the evader, and thus can be safely ignored by the evader. (Such a pursuer is more than 2 moves away from threatening the evader.) Similarly, for any pursuer p positioned on a wall, map its position to the nearest point p_s on the skeleton, called the *shadow* of p . (Thus, p_s is the foot of the perpendicular from p to the edge of $G(D)$ that is closest to p .) Since $d(p_s, e) \leq d(p, e)$, the evader only needs to ensure its distance to p_s remains more than 1. Thus, we only need to ensure that the evader maintains its distance condition with respect to two pursuers (or their shadows) that are constrained to move along the skeleton graph. *However, the pursuers are not constrained to the vertices of the skeleton graph; they can situate in the interior of skeleton graph edges.*

The evader’s strategy is *reactive*: it remains at a vertex until some pursuer is within distance 1. When one or both pursuers are within distance 1 of the evader, we show that the evader can move to a safe neighboring vertex and restore its distance condition. In particular, suppose evader’s current location is vertex u in $G(D)$, and let x, y, z be the three neighboring vertices of u . Then it is easy to see that no point other than u among (the line segments forming) the edges of $G(D)$ is within distance one of more than one neighbor in the set $\{x, y, z\}$. This follows because the minimum length path joining any two points of the skeleton graph lies entirely in the skeleton graph (i.e., it does not use the walls or top faces

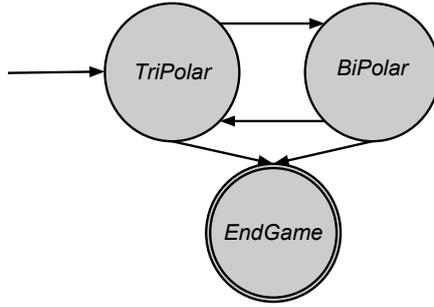


Fig. 2 A finite state machine representing the possible states of the pursuit and transitions between them.

of the surface S). Thus, at least one neighbor of u among $\{x, y, z\}$ is more than 1 away from both the pursuers, and this is the vertex to which e moves.

□

3 Catching the Evader with 4 Pursuers

We begin with a high level description of the pursuers' strategy, and then develop the necessary technical machinery to prove its correctness.

3.1 Surround-and-Contract Pursuit Strategy

The pursuers' overall strategy is conceptually quite simple: repeatedly shrink the region containing the evader while making sure that it cannot escape from this region, which can be intuitively thought of as a *surround-and-contract* strategy. More specifically, at any time, the evader is constrained within a connected portion S_i of the surface S , which is bounded by at most three paths, each guarded by a pursuer. The fourth pursuer is used to divide S_i into two non-empty regions (contraction), trapping the evader within one of them. This division is done in such a way that at least one of the 3 pursuers bounding S_i becomes free, thus allowing the process to continue until the target region reduces to a single triangle, and the capture can be completed.

The paths used by the pursuers are shortest paths on the polyhedral surface, *restricted to the current region*. The computation of shortest paths on a polyhedral surface is a well-known problem in computational geometry, and we rely on the following result of [7, 20]: given a source point x on the surface of a polyhedron S of n vertices, one can compute a shortest path map encoding the shortest paths from x to all other points on S , in $O(n^2)$ time using $O(n \log n)$ space. With this map, one can find the shortest path from x to any other point y in time $O(\log n + k)$ when the path consists of k arcs.

We use *phases* to monitor the progress of the algorithm: in phase i , the region containing the evader is denoted S_i where $S_i \subseteq S_{i-1}$, for all i . Each time the pursuers guard a new path dividing S_i , the phase transitions, with S_{i+1} as the region containing the evader. In addition, each region S_i has a rather special form: it is bounded by either two or three shortest paths. The finite automaton of Figure 2 shows the simple state diagram of the pursuit: the pursuit transitions between regions bounded by 2 and 3 paths until it reaches a special terminal state marked ENDGAME. For ease of reference, we name the first two states BIPOLAR and TRIPOLAR to emphasize that the regions corresponding to these states are bounded by shortest paths between 2 or 3 points (poles). The region in the terminal state ENDGAME is also bounded by 3 paths but contains no vertices in the interior (only the points of the boundary paths), which simplifies the search leading to capture. In particular, the three possible states throughout the pursuit are the following:

- BIPOLAR: S_i is bounded by two shortest paths $P_{a,b}$ and $P'_{a,b}$ between two points (poles) a and b .
- TRIPOLAR: S_i contains at least one interior vertex, and is bounded by three shortest paths $P_{a,b}$, $P_{b,c}$, and $P_{a,c}$.
- ENDGAME: S_i has no interior vertices and is bounded by three shortest paths $P_{a,b}$, $P_{b,c}$, and $P_{a,c}$.

We initialize the pursuit by choosing a triangular face (a, b, c) of the surface, and assigning one pursuer to each of the three (single-arc) shortest paths $P_{a,b}$, $P_{b,c}$, and $P_{a,c}$. If the evader lies inside the triangle face, we enter the terminal state ENDGAME; otherwise, we are in state TRIPOLAR. The fourth pursuer shrinks the region S_i , resulting in a smaller TRIPOLAR region, or forces a transition to a BIPOLAR region. In each state BIPOLAR, at least one interior vertex is eliminated from S_i . Further, each state consists of a finite number of phases, which guarantees that the algorithm terminates in the region ENDGAME.

In the following, we use $\nu(S_i)$ to denote the *number of interior* vertices of S_i ; that is, the number of vertices in S_i that are not on the boundary paths. Throughout the pursuit, the following invariant is maintained.

PURSUIT INVARIANT. During the i th phase of the pursuit, (1) $S_i \subseteq S_{i-1}$, (2) $\nu(S_i) \leq \nu(S_{i-1})$, and if phase $i - 1$ is in state BIPOLAR, then $\nu(S_i) < \nu(S_{i-1})$, and (3) at most 4 paths are guarded, each by a single pursuer at any time.

The first condition ensures that the region containing the evader only shrinks; the second ensures that at least one interior vertex is removed in state BIPOLAR; and the third ensures that 4 pursuers succeed in capturing the evader.

3.2 Guarding Shortest Paths

Our algorithm employs one pursuer to guard a shortest path, ensuring that any attempt by the evader to cross the shortest path leads to capture. The key idea behind this strategy is the “projection” of the evader along the shortest path, defined as follows.

PROJECTION. Given a shortest path $P_{a,b}$ between two points a and b , and the current evader location e , a point e_π on $P_{a,b}$ is called the *projection* of e if $d(e_\pi, x) \leq d(e, x)$, for all $x \in P_{a,b}$.

That is, if a pursuer p is positioned at e_π , then it is always closer than evader to every point of $P_{a,b}$, and therefore any move by the evader crossing $P_{a,b}$ leads to capture by p on its next move. While multiple projections may exist, the pursuers will guard a path by maintaining their location at the *canonical projection* of the evader, defined as follows.

CANONICAL PROJECTION. Given a shortest path $P_{a,b}$ between two points a and b , and the current evader location e , a point e_π on $P_{a,b}$ is called the *canonical projection* of e if $d(a, e_\pi) = \min(d(a, e), d(a, b))$.

The following three lemmas establish the technical preliminaries about the existence, maintainability, and reachability of the canonical projection. *Throughout, a shortest path always means the minimum length path restricted to the current subsurface S_i , and e_π refers to the unique canonical projection.*

Lemma 1 *Given any shortest path $P_{a,b}$ on the polyhedral surface, the canonical projection e_π is a projection of the evader.*

Proof Let us first consider the more interesting case where $d(a, b) \geq d(a, e)$ and let e_π be the point at distance $d(e, a)$ along $P_{a,b}$. For any point $x \in P_{a,b}$ such that e_π precedes x , the condition $d(e_\pi, x) > d(e, x)$ leads to a violation of $P_{a,b}$ being a shortest path, as follows:

$$\begin{aligned} d(a, x) &= d(a, e_\pi) + d(e_\pi, x) \\ &= d(a, e) + d(e_\pi, x) \\ &> d(a, e) + d(e, x) \end{aligned}$$

Similarly, for any point x that precedes e_π , the condition $d(x, e) < d(x, e_\pi)$ also leads to a violation:

$$\begin{aligned} d(a, e) &\leq d(a, x) + d(x, e) \\ &< d(a, x) + d(x, e_\pi) \\ &= d(a, e_\pi) \end{aligned}$$

which is a contradiction because $d(a, e) = d(a, z)$, and thus e_π is a projection of e .

On the other hand, if $d(a, b) < d(a, e)$, then b is chosen as e_π . In this case, the argument is identical to the second case above: $\forall x$ that precede e_π , $d(x, e) \geq d(x, e_\pi)$, and thus e_π is a projection. \square

Lemma 2 *Suppose the current position of the evader is e , the pursuer p is positioned at the canonical projection e_π on the shortest path $P_{a,b}$, and the evader moves to a new position e' . Then, p can reposition itself at the new canonical projection e'_π in one move, or capture the evader if the evader's move crossed the path $P_{a,b}$.*

Proof First, suppose that the new position e' is on different side of the path $P_{a,b}$ than e , namely, the evader crosses the path, say, at a point z . Because the evader can move at most distance one, we have the inequality $d(e, z) + d(z, e') \leq 1$. On the other hand, since p is located at the canonical projection of e before the move, $d(p, z) \leq d(e, z)$. Therefore, the new position of the evader e' is within distance one of p , and the pursuer can capture the evader on its move.

If the evader does not cross $P_{a,b}$, and moves to a position e' on the same side of the path, let e'_π be the canonical projection of e' . Because the evader moves distance at most one further from u or at most one closer to u , it must satisfy $d(e_\pi, e'_\pi) \leq 1$, and so p can relocate from e_π to e'_π in one move. \square

Lemma 3 *Consider a shortest path $P_{a,b}$ on the polyhedral surface S , and suppose a pursuer p is located at the endpoint a of this path. Then, after at most $L + 1$ moves, p can locate itself at the canonical projection of the evader, where L is the (Euclidean) length of the $P_{a,b}$.*

Proof By definition the canonical projection can move at most distance one per turn. Thus p may move along $P_{a,b}$ until either, p reaches e_π , or when e moves to e' on its turn the new projection e'_π precedes p on $P_{a,b}$. However, because $d(e_\pi, e'_\pi) \leq 1$, it follows that $d(p, e'_\pi) \leq 1$, and thus p may move to e'_π on its turn. In the worst case p traverses the entire path and reaches e_π in at most $L + 1$ moves. \square

These lemmas together show that a single pursuer is able to guard a shortest path on the surface. We now describe the pursuers' strategy for each of the three states: BIPOLAR, TRIPOLAR, ENDGAME.

3.3 Pursuit Strategy for the TRIPOLAR State

In TRIPOLAR state, the current region S_i is bounded by three shortest paths, $P_{a,b}$, $P_{a,c}$, and $P_{b,c}$, between the three poles a, b, c . The pursuers' strategy is to force the game either into BIPOLAR or ENDGAME state while preserving the Pursuit Invariant. Towards that goal, we need to introduce some properties of shortest paths on polyhedral surfaces.

It is well-known that a shortest path is a sequence of line segments (arcs), whose endpoints lie on the edges of the surface, and that the path crosses any edge of the surface at most once. Thus, the sequence of edges crossed by a path, called the *edge sequence*, consists of at most n edges. Given a source point a and an edge (b, c) , it is also known that (b, c) is partitioned into $O(n)$ closed *intervals of optimality* [20], where the shortest path from a to any point d in an interval follows the same edge sequence. Let us suppose that an edge (b, c) is partitioned into k intervals of optimality, $[d_0, d_1], [d_1, d_2], \dots, [d_{k-1}, d_k]$, where the edge sequence for the interval $[d_{i-1}, d_i]$ is denoted as σ_i . Since two adjacent intervals, say $[d_{j-1}, d_j]$ and $[d_j, d_{j+1}]$, share a common endpoint d_j , there are two equal length shortest paths from a to d_j , following edge sequences σ_j and σ_{j+1} . Because our algorithm may guard one or both of these shortest paths, we use a superscript to identify the associated edge sequence. In particular, the shortest path from x to y under the edge sequence σ_j is denoted $P_{x,y}^j$.

The following lemma shows that if the shortest paths $P_{a,b}$ and $P_{a,c}$ have the same edge sequence, and $P_{b,c}$ is a single arc, then the interior of the region bounded by these 3 paths has no vertex of the surface, which implies that the pursuit region has entered the terminal state ENDGAME.

Lemma 4 *Suppose the current region S_i is bounded by pairwise shortest paths between the three points a, b, c , and that $P_{b,c}$ consists of a single arc. Then, the paths $P_{a,b}$ and $P_{a,c}$ follow the same edge sequence if and only if S_i contains no interior vertices.*

Proof Clearly, if $P_{a,b}$ and $P_{a,c}$ have the same edge sequence, then there cannot be an interior vertex in S_i because $P_{b,c}$ is a single arc. For the converse, if S_i has no interior vertices and $P_{b,c}$ is a single arc, then S_i can only contain edges that intersect both $P_{a,b}$ and $P_{a,c}$. These edges do not cross each other, and therefore they must be crossed by $P_{a,b}$ and $P_{a,c}$ in the same order. \square

By the preceding lemma, if $P_{a,b}$ and $P_{a,c}$ follow the same edge sequence and $P_{b,c}$ consists of a single arc, then we are in the terminal state **ENDGAME**. Therefore, assume that either the edge sequences of $P_{a,b}$ and $P_{a,c}$ are unequal or $P_{b,c}$ consists of multiple arcs. In both cases, the following lemma shows how to either reduce $P_{b,c}$ to a single point, which changes the state to **BiPOLAR**, or replace $P_{a,b}$ and $P_{a,c}$ with shortest paths with the same edge sequence, and $P_{b,c}$ with a single arc, which changes the state to **ENDGAME**.

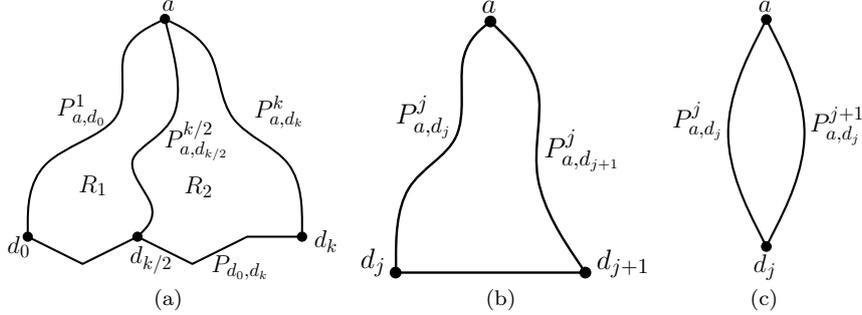


Fig. 3 Illustration for the proof of Lemma 5.

Lemma 5 *Suppose S_i is in state **TRIPOLAR**, then we can force a transition either to state **BiPOLAR** or state **ENDGAME**.*

Proof Consider the shortest path map with source a , and suppose it partitions $P_{b,c}$ into k intervals of optimality (across all of $P_{b,c}$'s arcs), $[d_0, d_1]$, $[d_1, d_2] \cdots$, $[d_{k-1}, d_k]$ with corresponding edge sequences $\sigma_1, \sigma_2, \cdots, \sigma_k$, where $d_0 = b$ and $d_k = c$. Relabel $P_{a,b}$ as P_{a,d_0}^1 , and $P_{a,c}$ as P_{a,d_k}^k , and order the paths by their endpoints on $P_{b,c}$ as follows:

$$P_{a,d_0}^1, P_{a,d_1}^1, P_{a,d_1}^2, P_{a,d_2}^2, \cdots, P_{a,d_{k-1}}^k, P_{a,d_k}^k$$

We leave two pursuers to guard (maintain canonical projections on) the paths $P_{a,b}$ and $P_{a,c}$, and deploy a guard on the center path $P_{a,d_{k/2}}^{k/2}$ (constrained to lie within the current region); see Figure 3(a). This path splits the original region S_i into two non-empty regions, each containing half the intervals of optimality, and we recurse the process on the side with the evader, namely, the region S_{i+1} . The first two conditions of the invariant are trivially satisfied, since the evader region can only shrink, and the third condition holds because the pursuer associated with either the path $P_{a,b}$ or $P_{a,c}$ is freed up, keeping the total pursuer count at four.

The recursion terminates when the evader is confined between two successive paths in the original ordering. In particular, if the evader is trapped between paths P_{a,d_j}^j and $P_{a,d_{j+1}}^j$, then we have state **ENDGAME** as shown in Fig. 3(b). On the other hand, if the evader is trapped between two paths P_{a,d_j}^{j-1} and P_{a,d_j}^j , we have successfully transitioned to state **BiPOLAR**, as shown in Fig. 3(c). It is clear that throughout this search, the evader remains confined to a subsurface of S_i and cannot escape without being captured, and that the pursuit invariant is maintained. Because the path $P_{b,c}$ has at most n arcs, with n intervals of optimality each, we have $k \leq n^2$. Thus, in $O(\log n)$ phases, we can force a change of state to either **BiPOLAR** or **ENDGAME**. \square

3.4 Pursuit Strategy for the **BiPOLAR** State

We now describe how to make progress when the search region is **BiPOLAR**. Without loss of generality, assume that the current region S_i is bounded by two shortest paths between points a and b , each guarded by a pursuer. The algorithm shrinks the region by removing at least one vertex from the interior of S_i . In particular, let c be a vertex of the surface that lies in the interior, and consider the two shortest paths (constrained to remain inside S_i) from c to a and b . The concatenation of these two paths splits

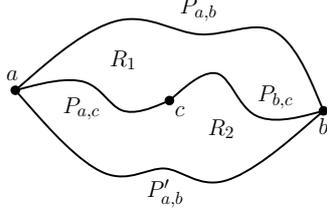


Fig. 4 An abstract illustration of the two paths that may be guarded during state BiPOLAR.

S_i into two subregions, say R_1 and R_2 , both bounded by three paths. (These paths can share a common prefix, starting at c , but they do not cross each other.) Only one of these regions contains the evader, and so by guarding $P_{a,b}$ and $P_{a,c}$ the state of the search transitions to either TRIPOLAR or ENDGAME depending on whether or not this region, which becomes S_{i+1} , contains an interior vertex. See Figure 4 for illustration. During this transition the pursuit invariant holds because (1) $R_1, R_2 \subseteq S_i$, (2) both R_1 and R_2 contain at least one fewer interior vertex, namely, c , and (3) at most 4 pursuers are used. Thus, we have established the following lemma, completing the discussion of the state BiPOLAR.

Lemma 6 *If the evader lies in a BiPOLAR region S_i , then we can force a transition to a TRIPOLAR or ENDGAME region with at least one fewer interior vertex, and no more than 4 pursuers are used during the pursuit.*

3.5 Pursuit Strategy for the ENDGAME State

We now describe how the pursuers capture the evader when the search region is ENDGAME. First, by Lemma 5, the path $P_{b,c}$ can be reduced to a single arc. Next, by Lemma 4, since S_i has no interior vertices, $P_{a,b}$ and $P_{a,c}$ follow the same edge sequence. Thus, S_i consists of a chain of faces, each a triangle or a quadrilateral. For ease of presentation, we assume that all faces are triangles, which is easily achieved by adding a diagonal to each quadrilateral. The pursuers perform a sweep of S_i , by repeatedly replacing $P_{b,c}$ with the previous edge in the edge sequence of $P_{a,b}$ and $P_{a,c}$, until the evader is trapped in a triangle each of whose sides are guarded by a pursuer. For example, in Figure 5(a), the fourth pursuer guards the edge (b, x_1) , which either confines the evader to the triangle b, c, x_1 or frees the evader guarding $P_{b,c}$.

Lemma 7 *Once the evader enters the ENDGAME state, the 4 pursuers can shrink the confinement region to a single triangle of S_i in $O(n)$ phases.*

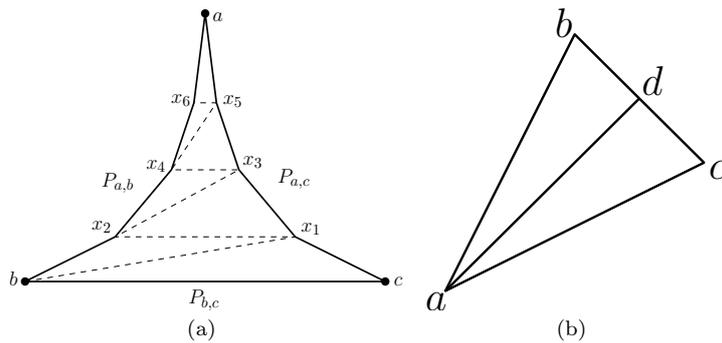


Fig. 5 Illustrating the algorithm used for capture in state ENDGAME.

Finally, the following lemma completes the capture inside the triangle.

Lemma 8 *If S_i consists of a single triangle, then in $O(\Delta_S \log \Delta_S)$ moves the evader can be captured.*

Proof The pursuers progressively “shrink” the triangle containing the evader, leading to eventual capture, as follows. Pick the midpoint of the arc (b, c) , say d , and deploy a guard on the arc (a, d) ; see Figure 5(b). This path splits the original triangle into two non-empty triangles, and we recurse the process on the triangle containing the evader. Notice that the pursuer associated with either the path $P_{a,c}$ or $P_{a,b}$ is freed up, keeping the total pursuer count at four. After $\log \Delta_S$ applications (b, c) will be replaced with an arc of length at most one, at which point a pursuer can capture the evader by sweeping the triangle once. At most $O(\log \Delta_S)$ paths of length $O(\Delta_S)$ are guarded, and so this process takes at most $O(\Delta_S \log \Delta_S)$ moves. \square

We analyze the total number of moves before the evader is captured. The total number of pursuer moves over all the BIPOLAR and TRIPOLAR moves is bounded by the number of paths guarded times the number of steps to guard those paths. By Lemma 3, the time to guard a path (reach the canonical projection) is proportional to its length, and the following result shows an upper bound on this length.

Lemma 9 *At a given phase i , Δ_{S_i} is $O(n \cdot \Delta_S)$.*

Proof Consider the longest shortest path P in S_i . Because it crosses any edge of the surface at most once, it has $O(n)$ arcs, each of length at most Δ_S , which yields the desired bound. \square

We can now state our main result.

Theorem 2 *On a n -vertex genus 0 polyhedral surface S , 4 pursuers can always capture the evader in $O(\Delta_S(n^2 \log n + \log \Delta_S))$ moves.*

Proof There are at most n phases in state BIPOLAR because each occurrence removes at least one interior vertex from S_i . Only two paths are guarded per phase, so BIPOLAR takes $O(n^2 \cdot \Delta_S)$ moves. There are at most $O(\log n)$ phases during TRIPOLAR to force the state transition, each requiring a single path to be guarded. Further, there are at most n transitions from state BIPOLAR to TRIPOLAR, and thus there are at most $O(n \log(n))$ phases in state TRIPOLAR, requiring $O(n^2 \log(n) \cdot \Delta_S)$ moves. In state ENDGAME, there are $O(\log n)$ phases where a path is guarded to reduce $P_{b,c}$ to a single arc, an additional $O(n)$ phases where an arc is guarded to confine the evader to a face, and finally $O(\log(\Delta_S) \cdot \Delta_S)$ moves are needed to capture the evader in a triangle. Adding them up, in the worst case, the total number of moves is $O(n^2 \log(n) \cdot \Delta_S + \log(\Delta_S) \cdot \Delta_S)$. \square

4 Catching the Evader on Genus g Surface

In this section, we show that $(4 + 4g)$ pursuers are sufficient to catch the evader on a polyhedral surface of genus $g > 0$. The main idea is to cut the surface along $2g$ cycles, reducing it to a genus 0 surface. By assigning 2 pursuers to each cycle, we can ensure that each cycle is guarded, and that the evader cannot cross a cycle without being captured. We then use the 4 remaining pursuers to search the genus zero surface and capture the evader.

The existence of these $2g$ cycles that split a genus g surface into genus 0 subsurfaces follows from a result of Erickson and Whittlesey [8]. The intuition behind their algorithm is simple: compute the cut locus of S , which is the closure of the set of points with at least two shortest paths from a base-point x . Then greedily choose the shortest cycle that does not disconnect the surface of S ; remove this cycle; and choose the next shortest cycle that does not disconnect S with the first cycle removed, and so on. After choosing $2g$ such cycles, all remaining cycles disconnect the surface, and thus the resulting surface has genus zero. Erickson and Whittlesey show that this greedy strategy results in a set of $2g$ loops with the minimum possible sum of lengths, for the given base-point. We need only a weaker property that each cycle is geodesic (composed of two shortest paths), and thus cannot be shortcut. In particular, we need the following result from [8].

Theorem 3 [8] *Given any piecewise-linear manifold M in \mathbb{R}^3 and any base-point $x \in M$, we can compute the shortest system of loops for M based at x in $O(n^2)$ time.*

Using a total of $4g$ pursuers, the $2g$ cycles found by Erickson and Whittlesey’s algorithm can be guarded, confining the evader to a gn vertex subsurface of genus zero, whereupon the evader can be captured by 4 additional pursuers.

Theorem 4 *On a n -vertex genus g polyhedral surface S , $4g + 4$ pursuers can always capture the evader in $O(((gn)^2 \log(gn) + \log \Delta_S) \cdot \Delta_S)$ moves.*

5 Pursuit Evasion Under Weighted Region Path Metric

Our surround-and-contract technique appears to be quite general, and may be applicable to many other settings where shortest paths are well-behaved and where the frequency of state transitions between BIPOLAR and TRIPOLAR can be combinatorially bounded. As one illustrative example, we consider the pursuit evasion problem on a polyhedral surface under a *region weighted* definition of shortest paths, and deduce the same result that 4 pursuers suffice in this setting as well.

In the weighted region model, each face f (triangle) of the polyhedral surface S is associated with a non-negative real weight w_f , and traveling along a line segment of length ℓ on this face incurs a *cost* of $w_f \cdot \ell$. (These weights can be used to model the non-homogeneity of movement speed on a terrain, such as paved roads, dirt roads, marshlands, sand, water etc.) Each edge e of the polyhedron also has a weight $w_e \in (0, \infty]$ subject to the condition $w_e \leq \min\{w_f, w_{f'}\}$, where f, f' are the two faces adjacent to e . This condition is imposed to disallow the unnatural phenomena of paths that travel arbitrarily close to an edge, but not on it.¹

A *minimum cost path* between a source x and a destination y under the weighted metric is a non-self-intersecting piecewise linear curve, which only makes turns at edges and vertices of the surface. These shortest paths, however, have some characteristics that are distinctly different from Euclidean length paths: e.g., the straight line segment xy is not necessarily the minimum cost path for two points x and y in the interior of a convex face f . Nevertheless, we show below that the basic structure and proof of surround-and-contract holds in this generalized path setting.

We first dispense with a computational issue. Computing a weighted region path is computationally non-trivial, and all the polynomial-time algorithm only compute a $1 + \epsilon$ approximation [2, 17, 19]. (Algorithms that compute an exact minimum cost path take time doubly exponential in n [19].) Our primary goal, however, is to show the correctness of the strategy and sufficiency of 4 pursuers, and we do not concern ourselves with the computational aspects.

A more serious issue is the *combinatorial complexity* of the paths. While Euclidean shortest paths cross a single edge of the surface at most once, a weighted minimum cost path can cross an edge $\Theta(n)$ times! Without a more careful analysis, this can lead to an exponential blowup in the complexity of the surface regions: a region bounded by 3 shortest paths can have boundary complexity $\Omega(n^2)$, causing a shortest path in it to have complexity $\Omega(n^4)$, and so on. In the following, we offer a more refined analysis of the weighted shortest paths and prove that such an explosion does not occur.

5.1 Path Complexity under the Weighted Metric

The number of moves required to transition between states BIPOLAR and TRIPOLAR is controlled by the number of arcs in a shortest path and the intervals of optimality into which an arcs is subdivided. Deriving a non-trivial upper bound on this complexity requires delving into the proof of the weighted region shortest path algorithm of Mitchell et al. [19], which is quite complicated. Instead, we offer below a substantially simpler and direct proof for bounding the number of intersections between an edge and a shortest path, which we are able to generalize to our more involved setting.

Lemma 10 *Let S be an n -vertex polyhedron with weighted regions, then any shortest path in S has at most $O(n^2)$ arcs.*

Proof Fix a shortest path P in S_i , order the edges of S_i in the increasing order of weight, and let e_1, e_2, \dots, e_n be this order. We claim that the edge e_i intersects P only $O(i)$ times.

Consider the edge e_i , and suppose it is intersected by P at $x_1, x_2, x_3, \dots, x_k$ in that order (meaning x_3 could precede x_2 on e_i , but P visits x_2 before x_3). Then we claim that P can always be chosen such that only one subpath $P_{x_j, x_{j+1}}$ intersects e_1 . For the sake of contradiction, suppose that $P_{x_l, x_{l+1}}$ also intersects e_1 , and without loss of generality suppose that $j+1 \leq l$. Further, let the first and last points $P_{x_j, x_{j+1}}$ intersects e_1 be y_1 and y_2 respectively (see Figure 6(a)). Then, we can construct a path P' by replacing the subpath P_{y_1, y_2} of P with the arc (y_1, y_2) . As P_{y_1, y_2} has Euclidean length at least as long as (y_1, y_2) , and is weighted at least as much as e_1 everywhere, P' can be no longer than P . This process can be repeatedly applied until there is only one subpath $P_{x_{j'}, x_{j'+1}}$ which intersects e_1 .

¹ edges of infinite weight and, for such an edge, it is permissible to travel along it at the cost of its neighboring face, but it is not permissible to cross it. This allows for the modeling of an impenetrable obstacle.

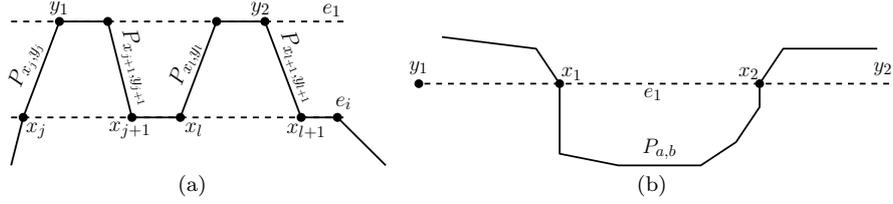


Fig. 6 Illustration of proof of Lemma 10.

Now suppose there are two subpaths $P_{x_j, x_{j+1}}$, and $P_{x_l, x_{l+1}}$ which intersect e_2 (excluding the single subpath which intersects e_1) first at y_1 and last at y_2 . Since we know neither subpath can intersect e_1 , neither subpath can traverse an edge of weight smaller than e_2 . Further, neither subpath can traverse a face f of smaller weight, as to do so would require crossing one of f 's bounding edges, which we know have weight less than or equal to the faces they bound. Thus, P_{y_1, y_2} has no arc with weight less than that of e_2 , and thus the arc (y_1, y_2) has weighted length no more than P_{y_1, y_2} and can replace it. Note that (y_1, y_2) cannot intersect P , as to do so would contradict the choice of P as the shortest path. Therefore, P can be chosen such that only one subpath $P_{x_{j'}, x_{j'+1}}$ intersects e_2 (in addition to the single subpath which intersects e_1). This process can be continually applied for each edge e_j where $j < i$ until we have accounted for all $i - 1$ edges with weight less than or equal to e_i .

The resulting scheme accounts for at most $i - 1$ subpaths to intersect e_i . Suppose then that were an i -th subpath, $P_{x_j, x_{j+1}}$. This subpath could necessarily only intersect e_l such that $l > i$. Thus, we can construct a path P' no longer than P by replacing the subpath $P_{x_j, x_{j+1}}$ of P with the arc (x_j, x_{j+1}) . Thus in the worst case $k = 2(i - 1) + 2 = 2i$, where there are $i - 1$ subpaths intersecting e_i , and an arc preceding and following x_2 and x_{k-1} respectively. Thus in the worst case P consists of $O(n^2)$ arcs. \square

The following lemmas generalize the previous result to bound the growth in path complexity during our algorithm.

Lemma 11 *Suppose S_i is a subregion of S whose boundary paths have m edges in total. Then a shortest path P inside S_i has at most $m + O(n^2)$ arcs.*

Proof Using a nearly identical proof to that of Lemma 10, it can be shown that the i -th shortest edge of S will still have $O(i)$ subpaths intersecting it, excluding those of the boundary. In Figure 6(a) we replaced P_{y_1, y_2} with the arc (y_1, y_2) . However, the situation in Figure 6(b) may arise, that is, e_1 may be intersected by the boundary of S_i . Because the boundary consists of only shortest paths, replacing (y_1, y_2) with the concatenation of (y_1, x_1) , P_{x_1, x_2} , (x_2, y_2) will guarantee that there is only one *new* subpath intersecting e_i that intersects e_1 , but, the boundary portion of that subpath may intersect e_i several times. Thus, a shortest path in S_i has at most $O(n^2)$ arcs from the $O(i)$ intersections per edge, and at most m additional arcs from the boundary of S_i . \square

Lemma 12 *Suppose after j state transitions the evader is confined to S_i . Then, a shortest path in S_i consists of at most $O(j \cdot n^2)$ arcs.*

Proof We now show that the boundary paths can only grow in complexity during state transitions (and not between two phases without a state transition). Thus a boundary path would gain at most $O(n^2)$ arcs per state transition, for a total of $O(j \cdot n^2)$ after j state transitions.

First observe that if S_i is in state BIPOLAR, then the boundary of S_{i+1} will have at most $O(n^2)$ more arcs than S_i . This is because the paths found splitting S_i can only have $O(n^2)$ arcs not on the boundary, and each edge of S_i 's boundary can only be present in one of the two paths (if we clip common prefixes following the poles). Next, note that while in state TRIPOLAR, there can be many stages, in each of which a single path is guarded, however, all the paths are from the *same shortest path map*. Thus, when we transition from state TRIPOLAR to BIPOLAR we have one path consisting of a single arc, and two paths which may traverse the boundary and have gained at most $O(n^2)$ new arcs. Again, no edge of the boundary will be present twice, and thus boundary can only have gained $O(n^2)$ new arcs. Finally, in state ENDGAME we need not worry about growth in path complexity, as once $P_{b,c}$ is reduced to a single edge Lemma 13 guarantees a path crosses each edge at most once. Thus after j state transitions, the

boundary of S_i has at most $O(j \cdot n^2)$ arcs, and thus any shortest path in S_i has at most $O(j \cdot n^2)$ arcs. \square

Next, we note that in the weighted case an arc on an n vertex surface may have up to $O(n^3)$ intervals of optimality [19]. Note that during the n stage transitions it takes to reach state ENDGAME the boundary will have at most $O(n^3)$ arcs. Thus a surface S_i may effectively have $O(n^3)$ vertices, and thus a single arc may have $O(n^9)$ intervals of optimality. However, Lemma 5 will still force a transition out of TRIPOLAR in $O(\log n)$ phases. Thus in $O(n \cdot \log n)$ phases the pursuit will reach state ENDGAME.

5.2 Modifications to State ENDGAME

Unlike in the unweighted case, the straight arc between two points no longer is necessarily a shortest path. Thus, we cannot simply “walk” the arc (b, c) up S_i as before. However, the minimum weight internal edge (an edge not in the boundary) in S_i is a shortest path and can be guarded, thus we will exploit this to reduce S_i to a single face. ²

Recall that state ENDGAME consists of a single chain of triangles (after reducing (b, c) to a single arc). Let (d_1, d_2) be the internal edge with minimum weight in S_i . Without loss of generality, suppose that d_1 is on $P_{a,b}$, then we deploy the fourth pursuer to guard $P_{d_1,c}$. The following Lemma shows that $P_{d_1,c}$ crosses each edge at most once.

Lemma 13 $P_{d_1,c}$ crosses each edge at most once.

Proof Suppose that the edge (x, y) is intersected twice at points x_1 and x_2 , and let (x, y) be the first such edge in the edge sequence followed by $P_{a,b}$ and $P_{a,c}$ starting from a . Then notice, there must be a second edge after (x, y) in the edge sequence that is also intersected twice. This is true as $P_{d_1,c}$ only turns at edges and vertices, and thus must intersect at least one edge between the first and second intersection of (x, y) , and then this edge would be intersected again by $P_{d_1,c}$ before reaching c . Let the edge with minimum weight that is intersected twice (besides (x, y)) be (u, v) , and suppose it is intersected at u_1 and u_2 . See Figure 7(a).

Suppose that (x, y) has weight ω_1 and (u, v) has weight ω_2 . If $\omega_1 \leq \omega_2$, construct a path $P'_{d_1,c}$ by replacing P_{x_1,x_2} with the arc (x_1, x_2) . Similarly, if $\omega_2 \leq \omega_1$, construct $P'_{d_1,c}$ by replacing P_{u_1,u_2} with the arc (u_1, u_2) . In both cases, $P'_{d_1,c}$ is no longer than $P_{d_1,c}$, and each edge has one less intersection. This can be repeatedly applied until no edge is intersected more than once. \square

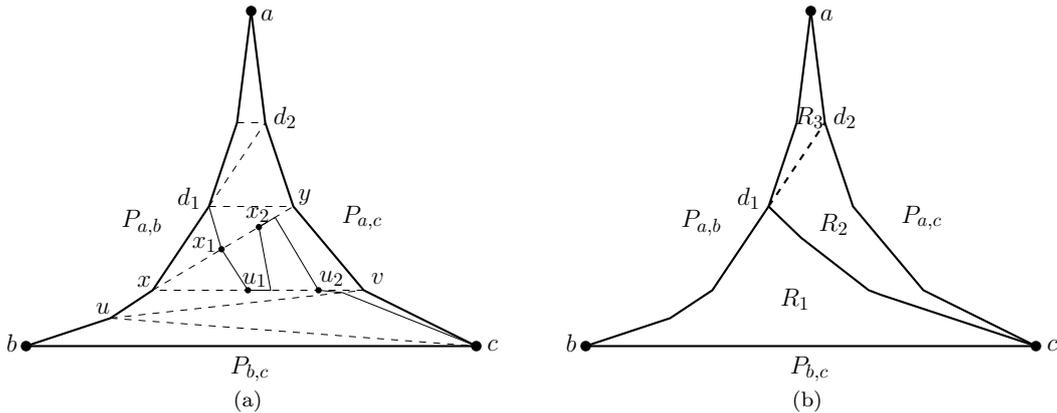


Fig. 7 In (a) an illustration for the proof of Lemma 13, and in (b) an illustration for the proof of Lemma 14.

Thus $P_{d_1,c}$ can be guarded without introducing any additional internal edges. The following Lemma shows how to remove all internal edges from S_i by guarding such a path.

² It is possible that the minimum weight internal edge is not a shortest path. However, any shorter path necessarily includes part of the guarded boundary, and the evader cannot move along such a path without being captured.

Lemma 14 *Suppose S_i is a weighted subsurface in state ENDGAME, and $P_{b,c}$ consist of a single arc, then S_i can be reduced to a single face.*

Proof We describe a procedure to remove an internal edge in at most two phases and maintain the property that one of the three bounding paths is an arc. This procedure can then be applied recursively to remove the remaining internal edges. Without loss of generality, suppose (d_1, d_2) is the minimum weight internal edge, and d_1 is on $P_{a,b}$, see Figure 7(b). Deploy the fourth pursuer to guard $P_{d_1,c}$. If the evader is in R_1 , then we are done, as (d_1, d_2) is no longer an interior edge, (b, c) is a single arc, and the pursuer guarding $P_{a,c}$ is no longer necessary.

Otherwise, (b, c) no longer needs to be guarded, and the pursuer is reused to guard (d_1, d_2) , which is a shortest path. Then, regardless of whether the evader is in R_2 or R_3 , (d_1, d_2) is no longer an internal edge and has become part of the boundary, and one of the bounding paths no longer needs to be guarded. Thus this process can be applied recursively to R_2 or R_3 to remove further internal edges. Further, in both stages the Pursuit Invariant is trivially maintained.

By Lemma 12 and the fact there are $O(n)$ state transitions, each boundary path can cross at most $O(n^3)$ edges. Thus, after $O(n^3)$ phases all internal edges are removed and the evader is confined to a single face. \square

5.3 Weighted Time to Capture

With the preceding sections we have covered all three states, and thus conclude that our algorithm will result in the capture of the evader. However, we must still address the time to capture. As in the unweighted case, the diameter of the environment can grow as we confine the evader to smaller a smaller subsurfaces of S , but unlike before, it can grow larger. This is because the original diameter may be small due to passing through regions with small weights, which are subsequently removed via path guarding. Let the minimum (non-zero) weight assigned to any face or edge be ω_{min} and similarly let ω_{max} be the maximum weight. The following Lemma bounds the path growth in terms of the input parameters.

Lemma 15 *In a given phase i , Δ_{S_i} is at most $O(\frac{\omega_{max}}{\omega_{min}} \cdot n^3 \cdot \Delta_S)$*

Proof Consider an arbitrary shortest path P in S_i . By Lemma 12 we know that any shortest path in S_i consists of at most $O(n^3)$ arcs. Let x, y be an arc of P with Euclidean length ℓ . Then, the shortest x, y path in S has weighted length at most $\omega_{min} \cdot \ell \leq \Delta_S$. In S_i , the arc x, y has length at most $\omega_{max} \cdot \ell$, and thus the arc x, y has weighted length at most $\frac{\omega_{max}}{\omega_{min}} \cdot \Delta_S$. Thus, as there are at most $O(n^3)$ arcs, the maximum length of any shortest path in S_i is at most $O(\frac{\omega_{max}}{\omega_{min}} \cdot n^3 \cdot \Delta_S)$. \square

Theorem 5 *Given a polyhedron S with n vertices, and weighted regions with min weight ω_{min} and max weight ω_{max} , 4 pursuers can capture the evader in $O(\frac{\omega_{max}}{\omega_{min}} \cdot n^6 \cdot \Delta_S + \log((\frac{\omega_{max}}{\omega_{min}}) \cdot \Delta_S) \cdot \frac{\omega_{max}}{\omega_{min}} \cdot \Delta_S)$ moves.*

Proof First notice, that with the exception of ENDGAME, the only increase in the time bound from the non-weighted problem is the increased worst case path length. Thus, the worst case number of moves in states BIPOLAR and TRIPOLAR is $O(\frac{\omega_{max}}{\omega_{min}} \cdot n^4 \log(n) \cdot \Delta_S)$. Then, in state ENDGAME there may be up to $O(n^3)$ phases in which a path is guarded taking a worst case $O(\frac{\omega_{max}}{\omega_{min}} \cdot n^6 \cdot \Delta_S)$ moves. Finally, when the evader is captured on the final face, it can have diameter at most $O(\frac{\omega_{max}}{\omega_{min}} \cdot \Delta_S)$, and thus by Lemma 8 the evader can be captured in at most $O(\log((\frac{\omega_{max}}{\omega_{min}}) \cdot \Delta_S) \cdot \frac{\omega_{max}}{\omega_{min}} \cdot \Delta_S)$ moves. Therefore, the worst case number of moves to capture the evader is $O(\frac{\omega_{max}}{\omega_{min}} \cdot n^6 \cdot \Delta_S + \log((\frac{\omega_{max}}{\omega_{min}}) \cdot \Delta_S) \cdot \frac{\omega_{max}}{\omega_{min}} \cdot \Delta_S)$. \square

6 Closing Remarks

In this paper, we considered a natural extension of the well-known pursuit-evasion game lion-and-the-man. Our extension analyzes the game on the surface of a polyhedral surface, and derives upper and lower bounds on the number of pursuers. If the surface has genus g , we show that $(4 + 4g)$ pursuers can always catch an equally fast and adversarial evader. We also showed that at least 3 pursuers are needed even on a genus 0 polyhedral surface. Closing the gap of one between the upper and lower bounds for the genus 0 surface is an open problem. Another outstanding open problem is to show a (genus dependent) lower bound on the number of pursuers for higher genus surfaces.

References

1. M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1 – 12, 1984.
2. L. Aleksandrov, M. Lanthier, A. Maheshwari, and J.-R. Sack. An ϵ -approximation algorithm for weighted shortest paths on polyhedral surfaces, 1998.
3. S. Alexander, R. Bishop, and R. Ghrist. Capture pursuit games on unbounded domains. *Enseign. Math. (2)*, 55(1-2):103–125, 2009.
4. S. Alpern, R. Fokkink, R. Lindelauf, and G.-J. Olsder. The ”princess and monster” game on an interval. *SIAM J. on Control and Optimization*, 47(3):1178–1190, 2008.
5. 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, September 2012.
6. S. D. Bopardikar, F. Bullo, and J. Hespanha. A cooperative homicidal chauffeur game. In *46th IEEE Conference on Decision and Control*, pages 4857 –4862, 2007.
7. J. Chen and Y. Han. Shortest paths on a polyhedron. In *Proc. of 6th Symposium on Computational Geometry*, pages 360–369, New York, NY, USA, 1990. ACM.
8. J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. In *Proc. SODA*, pages 1038–1046, 2005.
9. L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani. Visibility-based pursuit-evasion in a polygonal environment. *International Journal of Computational Geometry and Applications*, 9(5):471–494, 1999.
10. V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion in a polygonal environment. *IEEE Transactions on Robotics*, 21(5):875 – 884, 2005.
11. V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion with local visibility. *SIAM Journal on Discrete Mathematics*, 1:26–41, 2006.
12. K. Klein and S. Suri. Catch me if you can: Pursuit and capture in polygonal environments with obstacles. In *Proc. of 26th Conference on Artificial Intelligence*, pages 2010–2016, 2012.
13. K. Klein and S. Suri. Pursuit evasion on polyhedral surfaces. In *Proc. of 24th International Conference on Algorithms and Computation (ISAAC)*, 2013.
14. A. Kolling, A. Kleiner, M. Lewis, and K. Sycara. Pursuit-evasion in 2.5d based on team-visibility. In *Proc. Int. Conf. on Intelligent Robots and Systems*, pages 4610 –4616, 2010.
15. A. Kovshov. The simple pursuit by a few objects on the multidimensional sphere. *Game Theory & Applications II*, pages 27–36, 1996.
16. J. E. Littlewood. *Littlewood’s Miscellany*. Cambridge University Press, 1986.
17. C. S. Mata and J.S.B. Mitchell. A new algorithm for computing shortest paths in weighted planar subdivisions. In *Proc. 13th Symposium on Computational Geometry*, pages 264–273, 1997.
18. A. Melikyan. Geometry of pursuit-evasion games on two-dimensional manifolds. *Annals of the International Society of Dynamic Games*, 9:173–194, 2007.
19. J. S. B. Mitchell and C. H. Papadimitriou. The weighted region problem: finding shortest paths through a weighted planar subdivision. *Journal of the ACM*, 38(1):18–73, January 1991.
20. J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, August 1987.
21. B. S. W. Schroder. The copnumber of a graph is bounded by $\lfloor 3/2 \text{ genus}(g) \rfloor + 3$. In *Categorical Perspectives Proc. of the Conference in Honor of George Strecker’s 60th Birthday*, pages 243–263, 2001.
22. J. Sgall. Solution of david gale’s lion and man problem. *Theor. Comput. Sci.*, 259(1-2):663–670, 2001.
23. I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM J. Comput.*, 21:863–888, October 1992.