Pursuit Evasion on Polyhedral Surfaces

Kyle Klein and Subhash Suri

University of California
Santa Barbara, CA, USA 93106
{kyleklein,suri}@cs.ucsb.edu

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 remain 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 [12], 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.

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 [1,3,5,8]. We are inspired by the oldest such problem, the so-called *man-and-the-lion 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 [12]: in Besicovitch's strategy, the lion can get arbitrarily close to the man but never quite reach it. This impossibility proof can be circumvented

L. Cai, S.-W. Cheng, and T.-W. Lam (Eds.): ISAAC2013, LNCS 8283, pp. 284–294, 2013.

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 the discrete time model: this avoids the intractable problem of computing 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 man-and-the-lion game in the plane (an easy corollary of [16]). 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 [7], while 3 pursuers are both necessary and sufficient for polygonal environments with multiple holes (obstacles) [4]. 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 [9].

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 [10], classification of environments where capture is achievable [2], or on game-theoretical questions [11,13].

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 [15]. 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 $p_1, p_2, \ldots$ 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 ith 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 (see 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). 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 argue that an evader can indefinitely avoid capture from two pursuers on the surface of this polyhedron. In particular, the two pursuers, p_1 and p_2 initially choose their positions, and then the evader picks its initial position at a vertex of $G(D)$ to satisfy $d(p_i, e) > 1$, for $i = 1, 2$. We show that regardless of the pursuers' strategies, the evader can indefinitely maintain this distance condition (after its move) by always moving among the vertices of $G(D)$. 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. Due to space limitation, we omit the further details and refer the reader to the full version of the paper.

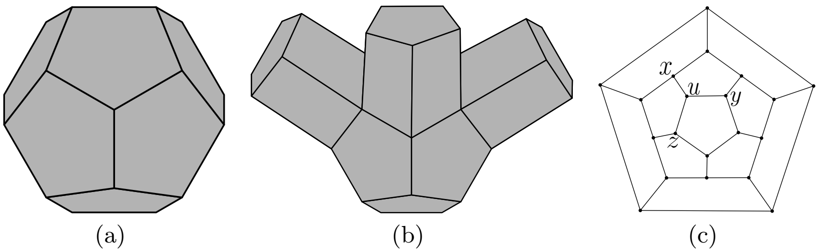

Fig. 1. A dodecahedron (a); partial construction with three faces orthogonally extended (b); and the skeleton graph (c)

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 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 [6,14]: 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

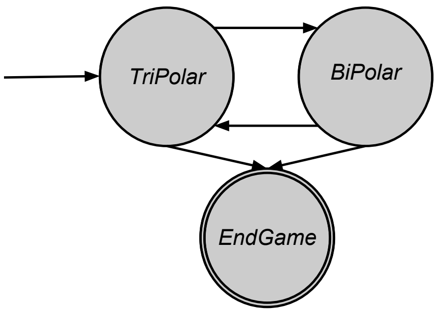

Fig. 2. A finite state machine representing the possible states of the pursuit and transitions between them

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 ith 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. Due to space limitation, we omit the further details and refer the reader to the full version of the paper. *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.*

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* $P_{a,b}$*.*

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* $P_{a,b}$*.*

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* [14], 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, $[d_0, d_1], [d_1, d_2], \cdots, [d_{k-1}, d_k]$, 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 $P^j_{x,y}$.

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. □

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.

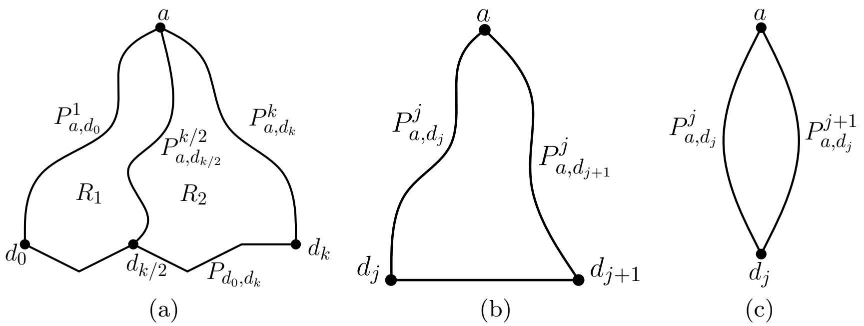

Fig. 3. Illustration for the proof of Lemma 5

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 $d_o = b$ and $d_k = c$. Relabel $P_{a,b}$ as P^1_{a,d_0}, and $P_{a,c}$ as P^k_{a,d_k}, and order the paths by their endpoints on $P_{b,c}$ as follows:

$$P^1_{a,d_0}, P^1_{a,d_1}, P^2_{a,d_1}, P^2_{a,d_2}, \ldots, P^k_{a,d_{k-1}}, P^k_{a,d_k}$$

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 $P^{k/2}_{a,d_{k/2}}$ (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^j_{a,d_j} and $P^j_{a,d_{j+1}}$, then we have state EndGame as shown shown in Fig. 3(b). On the other hand, if the evader is trapped between two paths P^{j-1}_{a,d_j} and P^{j}_{a,d_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. □

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 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 $P_{a,b}$ an $P_{a,c}$ 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 pursers are used. Thus, we have established the following lemma, completing the discussion of the state BiPolar.

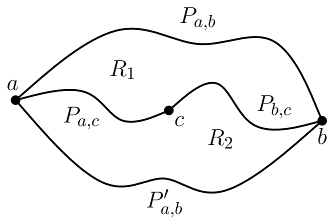

Fig. 4. An abstract illustration of the two paths that may be guarded during state BiPolar

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.*

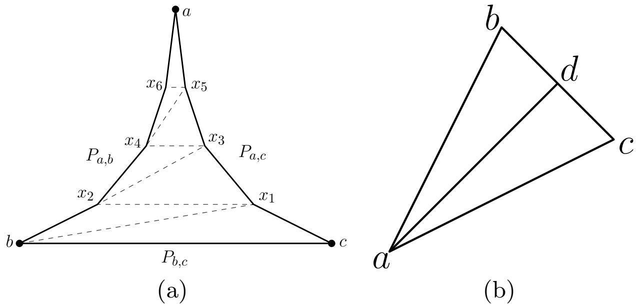

Fig. 5. Illustrating the algorithm used for capture in state EndGame

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. □

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.*

4 Extensions and Generalizations

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. In particular, we have the following two results, whose details can be found in the full version of the paper.

Theorem 3. *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.*

Theorem 4. *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.*

References

1. Aigner, M., Fromme, M.: A game of cops and robbers. Discrete Applied Mathematics 8(1), 1–12 (1984)
2. Alexander, S., Bishop, R., Ghrist, R.: Capture pursuit games on unbounded domains. Enseign. Math. (2) 55(1-2), 103–125 (2009)
3. Alpern, S., Fokkink, R., Lindelauf, R., Olsder, G.-J.: The "princess and monster" game on an interval. SIAM J. on Control and Optimization 47(3), 1178–1190 (2008)
4. Bhadauria, D., Klein, K., Isler, V., Suri, S.: Capturing an evader in polygonal environments with obstacles: The full visibility case. International Journal of Robotics Research 31(10), 1176–1189 (2012)
5. Bopardikar, S.D., Bullo, F., Hespanha, J.: A cooperative homicidal chauffeur game. In: 46th IEEE Conference on Decision and Control, pp. 4857–4862 (2007)
6. Chen, J., Han, Y.: Shortest paths on a polyhedron. In: Proc. of 6th Symposium on Computational Geometry, pp. 360–369. ACM, New York (1990)
7. Isler, V., Kannan, S., Khanna, S.: Randomized pursuit-evasion in a polygonal environment. IEEE Transactions on Robotics 21(5), 875–884 (2005)
8. Isler, V., Kannan, S., Khanna, S.: Randomized pursuit-evasion with local visibility. SIAM Journal on Discrete Mathematics 1, 26–41 (2006)
9. Klein, K., Suri, S.: Catch me if you can: Pursuit and capture in polygonal environments with obstacles. In: Proc. of 26th Conference on Artificial Intelligence, pp. 2010–2016 (2012)
10. Kolling, A., Kleiner, A., Lewis, M., Sycara, K.: Pursuit-evasion in 2.5d based on team-visibility. In: Proc. IROS, pp. 4610–4616 (2010)
11. Kovshov, A.: The simple pursuit by a few objects on the multidimensional sphere. In: Game Theory & Applications II, pp. 27–36 (1996)
12. Littlewood, J.E.: Littlewood's Miscellany. Cambridge University Press (1986)
13. Melikyan, A.: Geometry of pursuit-evasion games on two-dimensional manifolds. Annals of the International Society of Dynamic Games 9, 173–194 (2007)
14. Mitchell, J.S.B., Mount, D.M., Papadimitriou, C.H.: The discrete geodesic problem. SIAM Journal on Computing 16(4), 647–668 (1987)
15. Schroder, B.S.W.: The copnumber of a graph is bounded by $\lfloor 3/2$ genus(g)$\rfloor$ +3. In: "Categorical Perspectives" — Proc. of the Conference in Honor of George Strecker's 60th Birthday, pp. 243–263 (2001)
16. Sgall, J.: Solution of david gale's lion and man problem. Theor. Comput. Sci. 259(1-2), 663–670 (2001)