A Framework for Pursuit Evasion Games in $\mathbb{R}^n$

Swastik Kopparty and Chinya V. Ravishankar
Computer Science & Engineering Department
University of California—Riverside
swastik@cs.ucr.edu, ravi@cs.ucr.edu

April 19, 2005

Abstract

We present a framework for solving pursuit evasion games in $\mathbb{R}^n$ for the case of N pursuers and a single evader. We give two algorithms that capture the evader in a number of steps linear in the original pursuer-evader distances. We also show how to generalize our results to a convex playing field with finitely many hyperplane boundaries that serve as obstacles.

1 Introduction

The pursuit evasion problem arises naturally in various contexts. In its basic form, it consists of a single pursuer, whose goal is to capture a quarry, whose goal it is to evade capture. Capture is typically defined as occuring when a pursuer occupies the same position as the quarry, or comes to within a predefined distance of it. Variants of this game have also been considered in the literature [4, 1, 5, 14], and involve multiple pursuers, quarries, and various constraints on the space in which pursuit takes place.

In a simple version of the game, there are N pursuers $P_1, P_2, \cdots, P_N$ and a single quarry Q in some initial configuration in $\mathbb{R}^n$. The quarry is aware of the positions of all the pursuers, but each pursuer is aware only of its own position and that of the quarry. Each move of the game proceeds as follows. The quarry first moves by a distance $d \leq 1$ to a new position in $\mathbb{R}^n$. Then each of the N pursuers simultaneously and independently moves by a distance of no more than 1 to a new position. The pursuers win the game if one of them lands on the quarry after a finite number of moves. The quarry wins the game if it can avoid the pursuers for infinitely many moves.

We develop a framework for the study of such pursuit evasion games in $\mathbb{R}^n$. In its most general form, our model deals with games with multiple pursuers and a single quarry, in which all players move with bounded velocity in a region with zero or more hyperplane boundaries. We describe the initial conditions under which pursuers can win, and those under which the quarry can win. We describe an algorithm for the pursuers, assuming that each pursuer knows only its own location and that of the quarry. All pursuers follow the same algorithm. We show that in all cases, our algorithm is a winning strategy if the initial conditions allow the pursuers to have some winning strategy. Furthermore, our algorithm for the pursuers is memoryless, communication-free, and requires constant time computations, while achieving optimal catching times.

We first solve the pursuit problem for the simple case of the unrestricted space $\mathbb{R}^n$ by giving pursuit algorithms that succeed whenever the quarry does not have an obvious escape strategy. We then build upon the analysis here to solve the general version of the problem when the playing field is restricted by hyperplane boundaries.

2 Related Work

A variety of pursuit-evasion games have been studied in different contexts. The folklore "lion and man" game considered a single pursuer and a quarry inside a circular arena. Although the naive "solution" predicts that the pursuer would catch the quarry, Littlewood [10], based on ideas of Besicovitch, showed that the quarry could avoid capture (against any pursuit strategy) for an infinite amount of time. Related quantitave estimates on number of steps involved were derived in [2]. Earlier, Janković [7] derived necessary and sufficient conditions for the multiple lion man game when played on all of $\mathbb{R}^2$. It was shown that the man is capturable iff he lies within the convex hull of the lions. We show that the same condition holds for arbitrary dimensions, and that a natural variant of it holds when the game is played in a playing field restricted by hyperplane boundaries.

A version of the lion and man problem posed by David Gale for a playing field defined by the first quadrant was only recently solved by Sgall [13]. The game is played by one man and one lion on the first quadrant of the plane, with moves made as in the framework above. Sgall's algorithm for the lion is extremely simple and uses only elementary Euclidean geometry. The analysis also generalizes (as noted in [13]) to pursuit by a single lion in arbitrary dimensions in a wedge (with all bounding hyperplanes intersecting in a common point). Sgall's algorithm requires a catching time that is is quadratic in the initial lion-man distance, which is also tight for that problem.

The pursuit algorithms that we consider here for the general case are similar to those of Sgall, although our analysis has quite a different flavor. In particular, the asymptotic bound on the capture time that we prove (for the game on $\mathbb{R}^n$) is linear in the initial pursuer-evader distances, while Sgall's analysis of Sgall's algorithm (adapted to this setting) would provide only a quadratic upper bound. This speedup is only possible in the multiple pursuers case, and indeed, our analysis makes crucial use of this.

Other pursuit problems [8, 9] consider scenarios in which the pursuers, quarry, or both follow a random walk. Estimates are made of various parameters such as the time to capture and the distance travelled by the quarry before capture. For the sake of completeness, we also estimate the expected time to capture for our pursuit algorithms when the quarry follows a random walk.

Several papers [4, 1] consider discrete variants of the pursuit problem, and prove the hardness of certain pursuit algorithms on graphs. [4] also gives a tight analysis of several pursuit algorithms for the classical problem inside a circle.

A related question that has been considered in the literature is visibility-based pursuit within a polygon. Here the pursuer only knows the location of the quarry if the straight line connecting the pursuer and the evader does not interesect any of the walls of the bounding polygon. Several papers [5, 6, 14] give algorithms for this scenario.

3 Notation and Problem Definition

We use vectors to represent points. The directed line segment joining two points A and B is denoted by AB, and its length by $|AB|$. We use $B(p, r)$ to represent the closed ball of radius r centered at p.

We first formally define the game played by the quarry and the pursuers. Initially, the quarry is located at point Q^0 and the pursuers are located at $P_1^0, P_2^0, \ldots P_N^0$. The locations of the quarry and the pursuers are updated at every move according to rules that we will describe soon. At move i of the game, we denote the position of the quarry by Q^i, and that of pursuer P_k by P_k^i. The *configuration* of the game at step i is defined to be the tuple $\langle Q^i, P_1^i, P_2^i, \ldots, P_N^i\rangle$.

The i^{th} move of the game is played as follows:

- First the quarry selects Q^i such that $|Q^iQ^{i-1}| \leq 1$.
- Then, simultaneously for all k, pursuer P_k selects P_k^i such that $|P_k^iP_k^{i-1}| \leq 1$.

If for some finite i, k we have $P_k^i = Q^i$, then the pursuers win the game. Otherwise, the quarry wins the game. We wish to determine necessary and sufficient conditions on $Q^0, P_1^0, P_2^0 \ldots P_N^0$ so that there is a strategy for the pursuers to guarantee that they win game.

We say an algorithm or strategy $\mathcal{P}$ for the pursuers is *local* if the pursuers are not allowed to exchange information after the first move. $\mathcal{P}$ is *oblivious* if the pursuers never exchange any information. $\mathcal{P}$ is *memoryless* if a pursuer's new position depends solely on its current position and the quarry's current and new positions.

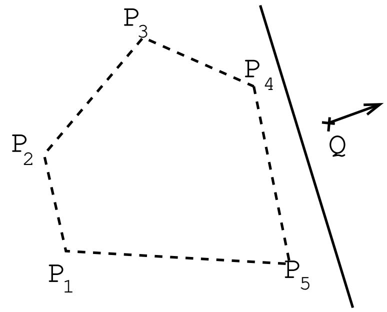

Figure 1: Escape strategy for a quarry outside the convex hull of the pursuers

4 Pursuit Without Boundaries in $\mathbb{R}^n$

We begin with the case where the region of pursuit is all of $\mathbb{R}^n$. We discuss two initial configurations: one in which the quarry can always win, and the other in which the pursuers can always win. Note that if Q^0 is within distance 1 of some P_k^0, then pursuers trivally win. From here onwards, we will assume that this is not the case.

4.1 Winning Initial Configurations for the Quarry

We denote by $\mathcal{C}^i \subset \mathbb{R}^n$ the convex hull of the positions $P_1^i, P_2^i, \ldots, P_N^i$ of the pursuers at step i. First, we observe that if Q^0 is not in the interior of the convex hull $\mathcal{C}^0$ of $P_1^0, P_2^0, \ldots P_N^0$ as in Figure 1, then there is a winning strategy for the quarry. The quarry simply constructs a hyperplane H that separates it from the pursuers (H exists by Farkas' lemma [3]), and moves away from the convex hull in the direction of the normal to H in steps of 1 unit, as shown in Figure 1. Since no P_i moves faster than Q, it is possible for Q to evade capture forever. This works even if Q lies on the boundary of the convex hull.

4.2 Winning Initial Configurations for the Pursuers

When Q^0 is strictly contained within the interior of the convex hull $\mathcal{C}^0$ of $P_1^0, \ldots, P_N^0$, we give an oblivious and memoryless algorithm for computing P_k^i given P_k^{i-1}, Q^{i-1}, and Q^i, which guarantees that the quarry will be caught within a finite number of steps. We will also show that this number of steps is optimal in terms of the initial distances to within a constant factor.

Algorithm 1 Algorithm PLANES

Given Q^i, Q^{i+1} and P_k^i, to compute P_k^{i+1}
Draw line L_k^{i+1} through Q^{i+1} parallel to the line segment $Q^iP_k^i$.
Pick P_k^{i+1} to be the point on L_k^{i+1} such that $|Q^{i+1}\, P_k^{i+1}|$ is minimized, subject to $|P_k^i\, P_k^{i+1}| \le 1$.

Intuitively, capture occurs for the following reason. For each k, draw a hyperplane through P_k^i orthogonal to $P_k^iQ^i$. Observe that if Q is in the convex hull of the P_k, then these hyperplanes restrict Q to a bounded region. No matter which direction Q moves, it approaches one of the hyperplanes. The pursuer on that wall, in its next move, will significantly reduce its distance to the quarry, while the others will at least maintain their distance.

Theorem 1 *Let a quarry lie in the interior of the convex hull of the the pursuers. If every pursuer follows algorithm PLANES, the quarry will be captured in a finite number of steps.*

Proof: Let d_k^i be the distance between P_k^i and Q^i, and let $d^i = \sum_{k=1}^n d_k^i$. We will show that

$$d_k^{i+1} \le d_k^i, \ \forall k, i, \text{ and} \tag{1}$$
$$d^{i+1} + C \le d^i, \ \forall i, \tag{2}$$

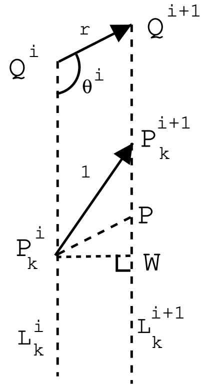

Figure 2: Algorithm PLANES: The move of pursuer P_k

where $C > 0$ depends only on the initial configuration.

Assume that move $(i+1)$ has just taken place, and that the quarry is still free. Observe that in algorithm PLANES, the points $P_k^i, P_k^{i+1}, Q^i, Q^{i+1}$ all lie on the same two-dimensional plane Π_k, so that we can restrict our discussion to positions and angles on Π_k. In Figure 2, let $\angle P_k^i Q^i Q^{i+1} = \theta_k^i$, and let $|Q^i Q^{i+1}| = r$. Introduce a point P to complete the parallelogram $P_k^i Q^i Q^{i+1} P$. From elementary trigonometry, we have $d_k^i - d_k^{i+1} = |PP_k^{i+1}| = |\overrightarrow{WP_k^{i+1}} + \overrightarrow{PW}| = \sqrt{(1 - r^2\sin^2(\theta_k^i)} + r\cos(\theta_k^i)$. (Here, the signs are taken care of automatically).

To prove (1), note that if $\theta_k^i \leq \pi/2$, $r\cos(\theta_k^i) \geq 0$, and hence $d_k^i - d_k^{i+1} \geq 0$. Otherwise, if $\theta_k^i > \pi/2$, $r\cos(\theta_k^i) = -\sqrt{r^2 - r^2\sin^2(\theta_k^i)}$, and so $d_k^i - d_k^{i+1} \geq \sqrt{1 - r^2\sin^2(\theta_k^i)} - \sqrt{r^2 - r^2\sin^2(\theta_k^i)} \geq 0$.

To prove (2), let $\vec{v}_k$ denote the unit vector in the direction $Q^0 P_k^0$. We observe that our construction in algorithm PLANES gurantees that for all i, $Q^{i+1}P_k^{i+1}$ is parallel to $Q^i P_k^i$, and in particular, parallel to $Q^0 P_k^0$. That is, the unit vector in direction $Q^i P_k^i$ stays at its initial value $\vec{v}_k$ for all time steps i before capture.

We first show that there is a $D_{min} > 0$ such that if $\vec{v}$ is any unit vector, then for some k, $\vec{v} \cdot \vec{v}_k \geq D_{min}$. By assumption, Q^0 lies strictly in the interior of the convex hull of the P_k^0. Thus there is a ball of radius $\epsilon > 0$ centered at the origin that lies entirely within the convex hull of the points on the unit sphere represented by the $\vec{v}_k$. Choose the largest such ϵ.

Since we are working in $\mathbb{R}^n$, any point in the convex hull of a set of points is in the convex hull of some $n+1$ of the points [11]. It follows that for any unit vector $\vec{v}$, $\epsilon\vec{v}$ is a convex combination of $n+1$ of the $\vec{v}_k$ (We are restricting our analysis to some $n+1$ lions which contain the man in their convex hull). Without loss of generality, let them be $\vec{v}_1, \ldots \vec{v}_{n+1}$. So there exist $\{\alpha_k \in [0,1],\ k = 1, \ldots, n+1\}$, such that $\epsilon\vec{v} = \sum_k \alpha_k \vec{v}_k$. Therefore, $\epsilon(\vec{v} \cdot \vec{v}) = \epsilon = \sum_k \alpha_k(\vec{v} \cdot \vec{v}_k)$. Thus for some k, $\vec{v} \cdot \vec{v}_k \geq \epsilon/(n+1)$. Take $D_{min} = \epsilon/(n+1)$. It is clear from the proof that D_{min} depends solely on the directions $\vec{v}_k$ and not on the magnitudes d_k^0.

Therefore for unit vector $\vec{v}$ in the direction of the quarry's motion, there is at least one k with $\vec{v} \cdot \vec{v}_k \geq D_{min}$. For that k, $d_k^i - d_k^{i+1} = \sqrt{1 - r^2\sin^2(\theta_k^i)} + r\cos(\theta_k^i) \geq \sqrt{1 - r^2} + r\cos(\theta_k^i)$. Now, $\sqrt{1 - r^2} + r\cos(\theta_k^i) - \cos(\theta_k^i) = \sqrt{1 - r^2} - \cos(\theta_k^i)(1 - r) > \sqrt{1 - r}(\sqrt{1 + r} - \sqrt{1 - r}\cos(\theta_k^i)) \geq 0$. So $d_k^i - d_k^{i+1} \geq \cos(\theta_k^i) = \vec{v} \cdot \vec{v}_k \geq D_{min}$.

Since the other d_j^i are nonincreasing in i, $d^i = \sum_{j=1}^n d_j^i$ decreases by at least $C = D_{min} > 0$ at each step. The quarry is captured when $d^i = 0$.

Since D_{min} is independent of the d_k^0, the time to capture is linear in d_k^0 for any specific k. No algorithm can do better in the general case. ∎

5 SPHERES: a more general algorithm

We now present a generalization of the method of [13]. This algorithm, called SPHERES, runs slower than the PLANES algorithm of Section 4.2, but has the advantage that all the d_k are strictly decreasing with

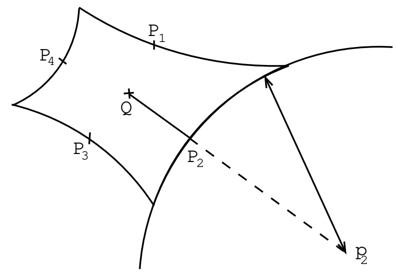

Figure 3: Algorithm SPHERES: Choosing points p_k in the initialization stage.

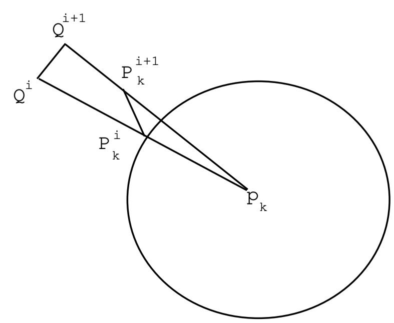

Figure 4: Algorithm SPHERES: The move of pursuer P_k.

time. We will use this property in Section 6 to design an algorithm for pursuit in the presence of obstacles. While the algorithm SPHERES is local, it is not oblivious. Our analysis for the unbounded plane shows that SPHERES performs far better and in a far wider range of contexts than [13] suggests.

Let Q^0 lie within the convex hull $\mathcal{C}^0 \subset \mathbb{R}^n$ of $P_1^0, P_2^0, \ldots, P_N^0$. Each pursuer P_k initially selects (by communication) a point p_k such that:

1. P_k^0 lies on the line segment $p_k Q^0$.
2. The connected component of $\mathbb{R}^n \setminus \bigcup_{k=1}^N B(p_k, |p_k P_k^0|)$ that contains Q^0 is bounded.

An example is shown in Figure 3.

To see that such points p_k can be chosen, note that if p_k was a point on the ray $Q^0 P_k^0$ "at infinity", the boundary of the corresponding ball $B(p_k, |p_k P_k^0|)$ would be a hyperplane passing through P_k^0 normal to $Q^0 P_k^0$. Since Q^0 lies within the convex hull of the P_k^0, the required properties for p_k are clearly satisfied. Thus choosing points that are sufficiently far on the line will suffice.

Algorithm 2 Algorithm SPHERES

Given Q^i, Q^{i+1} and P_k^i, to compute P_k^{i+1}
Draw line L_k^{i+1} passing through Q^{i+1} and p_k.
Choose P_k^{i+1} on L_k^{i+1} such that $|P_k^{i+1} Q^{i+1}|$ is minimized, subject to $|P_k^{i+1} P_k^i| \leq 1$.

Since pursuer P_k selects P_k^{i+1} solely based on P_k^i, Q^i and Q^{i+1}, this algorithm is memoryless and local. However, since condition (2) above requires the pursuers to communicate before the first step, the algorithm is not oblivious.

Let $\mathcal{A}$ denote the convex hull of the p_k, and let $\partial\mathcal{A}$ be its boundary. All pursuit will occur within $\mathcal{A}$. Now, let us define the *range* $\mathcal{R}^i$ of the quarry at step i to be $\mathcal{R}^i = \mathcal{A} \setminus \bigcup_{k=1}^N B(p_k, |p_k P_k^i|)$. After each step i, algorithm SPHERES enlarges the region defined by the spheres, thus shrinking the quarry's range $\mathcal{R}^i$. Further, each pursuer is always located at the point on its sphere closest to the quarry. Consequently, $\mathcal{R}^i$

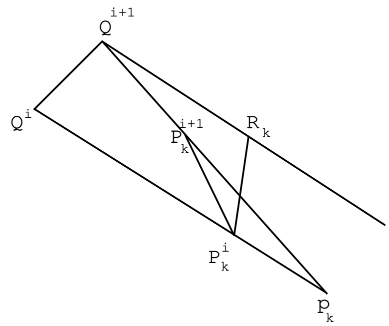

Figure 5: Algorithm SPHERES: Proof of Capture.

decreases by a significant amount at every step, so that it shrinks to 0 in a finite number of steps, resulting in a win for the pursuers. We now formalize this argument.

Lemma 1 *For all i before capture, $Q^i \in \mathcal{R}^i$.*

Proof: By definition, $Q^0 \in \mathcal{R}^0$. If Q isn't caught on move i, then, by construction of SPHERES, $p_k P_k^i Q^i$ are collinear in that order, for all k. Thus for all k, $Q^i \notin B(p_k, |p_k P_k^i|)$. Also, $Q^i \in \mathcal{A}$. So $Q^i \in R^i$. ∎

Theorem 2 *Let a quarry lie within the convex hull of the the pursuers. If every pursuer follows algorithm SPHERES, the quarry will be captured in a finite number of steps.*

Proof:

If the quarry is not caught on move i, we have that $Q^i \in \mathcal{R}^i$. Draw l through Q^{i+1} parallel to $P_k^i Q^i$. Choose point R_k on l so that $|R_k Q^{i+1}|$ is minimized with $|R_k P_k^i| = 1$. R_k is the point to which the PLANES algorithm would have taken the pursuer.

We will first show that $|Q^{i+1} P_k^{i+1}| < |Q^{i+1} R_k|$. Compare $\Delta P_k^i Q^{i+1} P_k^{i+1}$ and $\Delta P_k^i Q^{i+1} R_k$. $P_k^i Q^{i+1}$ is common, $|P_k^i P_k^{i+1}| = |P_k^i R_k|$, but $\angle Q^{i+1} P_k^i P_k^{i+i} < \angle Q^{i+1} P_k^i R_k$. Thus by the cosine rule, $|Q^{i+1} P_k^{i+1}| < |Q^{i+1} R_k|$.

For a point $x \in \mathcal{R}^i$, let $\vec{v}_k(x)$ be the unit vector in the direction xp_k. We claim that there exists a $D_{min} > 0$ such that for any unit vector v and any $x \in \mathcal{R}^i$, there is some k such that $\vec{v}_k(x) \cdot \vec{v} \geq D_{min}$.

Consider the closure of region $\mathcal{R}^0$. This is a closed set disjoint from $\partial\mathcal{A}$. Thus, there is an ϵ such that for any point p in $\mathcal{R}^0$, $B(p, \epsilon) \bigcap \partial\mathcal{A} = \emptyset$. So, any point in a ball of radius ϵ around x can be written as a convex combination of $n+1$ of the vectors $xp_k = |xp_k|\vec{v}_k$, $k = 1, 2, \ldots, n+1$.

$$\epsilon\vec{v} = \sum_k \alpha_k |xp_k| \vec{v}_k.$$

Thus, $\epsilon = \epsilon\vec{v} \cdot \vec{v} = \sum_k \alpha_k |xp_k| \vec{v} \cdot \vec{v}_k$. We choose

$$D_{min} = \frac{\epsilon}{(n+1)\max_{x \in \mathcal{R}^i, k} |xp_k|}.$$

Now, take x to be $Q^i \in \mathcal{R}^i \subset \mathcal{R}^0$, and $\vec{v}$ to be the unit vector in direction $Q^i Q^{i+1}$. Let $\vec{v}_k = \vec{v}_k(Q^i)$, the unit vector in direction $p_k Q^i$. By the above argument, we know that there is a k for which $\vec{v} \cdot \vec{v}_k \geq D_{min}$. For that k, imitating the analysis of the PLANES algorithm, $(|P_k^i Q^i| - |R_k Q^{i+1}|) \geq \vec{v} \cdot \vec{v}_k \geq D_{min}$. Thus, for that k, $|P_k^i Q^i| - |P_k^{i+1} Q^{i+1}| \geq D_{min}$. As in the PLANES case, we conclude that the quarry will be caught in finite time. ∎

A weaker result can also be obtained from the following lemma of Sgall [13].

Lemma 2 *As long as Q is not caught,* $|p_k P_k^{i+1}|^2 \geq |p_k P_k^i|^2 + 1$

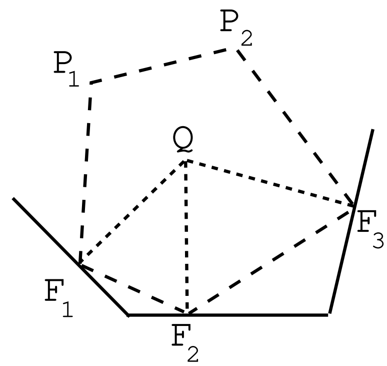

Figure 6: Pursures simulating hyperplane obstacles. Here F_k represents a virtual pursuer at the foot of the perpendicular from the current position of Q to the k^{th} hyperplane.

Proof: See [13] . ■

The argument goes as follows. Observe that since p_k, P_k^i, Q^i are collinear in that order, $p_k P_k^i \leq p_k Q^i$. Further, Q is confined to a bounded region, and thus $\sum_k |p_k Q^i|$ is bounded as long as Q is not caught. $\sum_k |p_k Q^i|^2 \geq \sum_k |p_k P_k^i|^2 \geq \sum_k |p_k P_k^0|^2 + Ni$, which grows unbounded with i, and hence the quarry will get caught in some finite amount of time.

The dependency on the d_k as obtained by this argument, however, is not as strong as possible. However, this argument generalizes easily to the case of obstacles, as we shall see in the next section.

6 Pursuit With Obstacles

We observe that algorithm SPHERES has the useful property that for each k, $|p_k P_k^i|^2$ increases by at least 1 for every increment in i. If we can guarantee that $\sum_k |p_k Q^i|^2$ is bounded from above by some constant C, then we will get $\sum_k |p_k P_k^i|^2 \leq \sum_k |p_k Q^i|^2 \leq C$, and hence the quarry will get caught in some finite time.

To obtain bounds on $\sum_k |P_k^i Q^i|^2$ we proceed as follows. As in [13], let a set of hyperplane boundaries exist in the pursuit region, and the quarry be forbidden to cross any of them. We observe that each of these boundaries can be simulated by a pursuer on the hyperplane who moves at each step to the position of the quarry's projection on the hyperplane (see Figure 6). These pursuers serve to corral the quarry, although they do not actively chase the quarry. For this case, we can derive a capture condition similar to that in Section 4.2.

Theorem 3 *Consider the quarry and pursuers game played on the solution space $\mathcal{F} = \{x : \vec{a}_i \cdot \vec{x} \geq b_i\}$ by 1 quarry and $N \geq 1$ pursuers. Drop perpendiculars from Q^0 to each of the hyperplanes $a_i \cdot x = b_i$, with the feet being $F_1, F_2, \ldots F_R$. Then, the quarry can be caught by the pursuers iff Q^0 is in the interior of the convex hull of $F_1, F_2, \ldots F_R, P_1^0, P_2^0, \ldots P_N^0$.*

Proof: Suppose Q^0 does not lie in the convex hull of the points $F_1, \ldots, F_R, P_1^0, P_2^0, \ldots P_N^0$. Then, draw a separating hyperplane for Q^0 and $\{F_1, F_2, \ldots, F_R, P_1^0, P_2^0, \ldots P_N^0\}$. Our strategy for the quarry is to move in a direction orthogonal to the hyperplane at jumps of 1 unit every time. Clearly, no pursuer can catch him. It remains to show that this move is "legal", that is, the quarry will never hit any of the bounding hyperplanes. But this is immediate, since the quarry does not lie in the convex hull of the F_i.

Now we describe the pursuit algorithm. First, we discard all the pursuers P_k for which the ray $Q^0 P_k^0$ intersects one of the bounding hyperplanes. It is easy to verify that Q^0 still lies in the interior of the convex hull of the remaining P_k^0 and F_j. So we assume that P_k^0 are all such that $Q^0 P_k^0$ does not intersect any of the bounding hyperplanes.

Our initialization phase selects, for each k, a point p_k such that:

- p_k, P_k^0 and Q^0 are collinear (in that order).

- p_k is in the playing field.
- The connected component of $\mathcal{F} \bigcap \left(\mathbb{R}^n \setminus \bigcup_k B(p_k, |p_k P_k^0|)\right)$ that contains Q^0 is bounded.

The fact that such points p_k exist can be deduced from an argument similar to the one used in the previous section (recall that we discarded some pursuers: it was precisely to justify the above step).

The pursuers then follow algorithm SPHERES. Observe that all the moves are legal (no moved dictated by SPHERES requires a pursuer to leave the playing field). To see this, recall that $\mathcal{F}$ is convex, $p_k \in \mathcal{F}$, and P_k^i as dictated by SPHERES lies between p_k and Q^i.

Suppose the quarry is not caught by move i. By definition, algorithm SPHERES will keep p_k, P_k^i, Q^i collinear in that order and hence $|p_k Q^i| \geq |p_k P_k^i|$. Furthermore, Q^i will always lie within the bounded region $\mathbb{R}^n - \bigcup B(p_k, |p_k P_k^0|)$. Also, by Lemma 2, $|p_k P_k^i|^2 \geq |p_k P_k^0|^2 + i$. Thus,

$$\sum_k |p_k Q^i|^2 \geq \sum_k |p_k P_k^i|^2 \geq Ni + \sum_k |p_k P_k^0|^2 \tag{3}$$

.

Q^i lies in a bounded region, and so $\sum_k |P_k Q^i|^2$ is bounded. But the right hand side of inequality 3 is unbounded as i increases. Thus there must be a move i on which the quarry is caught. ■

We note that this result subsumes the analysis in [13].

7 Optimality

For the case of pursuit without boundaries, the number of steps required by algorithms PLANES and SPHERES to catch the quarry is at most linear in any d_k^0. Clearly, no pursuit algorithm can hope to do better than this in general. When we use pursuers to simulate obstacles in the pursuit region, we get capture times *quadratic* in the d_k^0. There are instances when this bound is tight (see Sgall [13]).

8 A simultaneous continuous-time game

Consider a variant of the original game where on move i, all players move simultaneously. Since the pursuers have no advance knowledge of the quarry's movements, they can no longer be expected to catch the quarry. We therefore redefine capture as having occurred when at least one of the pursuers comes to within unit distance of the quarry.

This game has a straightforward solution based on the solution of the previous game. For the first move, the pursuers do nothing. On the second move, they execute the PLANES algorithm in response to the quarry's first move. In general, the game will be played exactly as though P_k^i is chasing Q^{i-1}, the quarry's previous position. Now, since the original algorithm could exactly catch the quarry, this algorithm will exactly catch the quarry's previous position. Since the quarry's moves by no more than unit distance at each step, the quarry is captured under our new definition.

This game can be extended to the case of continuous time in the obvious fashion.

9 Average Case Analysis

Now we show that if Q follows a random walk on the plane, our algorithm enables capture of the quarry in extremely low expected time. Further, by bounding the tail using deviation inequalities, we show that the probability of evading capture for more than $(1+\epsilon)$ times the expected number of steps is no more than $e^{-\Omega(\epsilon^2)}$. The basic tool we use for this bounding is the Chernoff bound [12].

Suppose for every i, Q^i is chosen independently at random from a distribution $\mathcal{D}^i$. Let us analyze the performance of the PLANES algorithm under this scenario. We consider spherically symmetric random walks: that is, the probability density function $\mathcal{D}^i$ is independent of θ, and so $p(r,\theta) = f(r)$. In such a scenario, we show that expected time to capture is really small.

The following analysis is for the PLANES algorithm, but a similar result holds for the SPHERES algorithm too. Recall that $d_k^i - d_k^{i+1} = |PP_k^{i+1}| = \sqrt{1 - r^2 \sin^2(\theta_k^i)} + r\cos(\theta_k^i)$. So,

$$E_{r,\theta_k^i}\left[d_k^i - d_k^{i+1}\right] = E_{r,\theta_k^i}\left[\sqrt{(1 - r^2\sin^2(\theta_k^i)}\right] + E_{r,\theta_k^i}\left[r\cos(\theta_k^i)\right] \geq E_{\theta_k^i}\left[|\cos\theta_k^i|\right] + E_{r,\theta_k^i}\left[r\cos(\theta_k^i)\right] = 2/\pi$$

Thus, $E_{r,\theta_k^i}\left[d^i - d^{i+1}\right] \geq 2N/\pi$. By linearity of expectation, $E\left[d^i\right] \leq d^0 - 2iN/\pi$.

We know that if X_i are independent, identically distributed random variables contained within $[0,1]$ with mean μ, the probability that $\sum_{i=0}^n X_i < (1-\epsilon)\mu n$ is no more than $e^{-2\epsilon^2 n}$.

Now, the $d^i - d^{i+1}$ are independent, identically distributed random variables with mean greater than $2N/\pi$, taking values in $[0, N]$. Let $D = \pi d^0/2N$, $E = D(1+\epsilon)$. By Chernoff bound, the probability that $d^0 - d^E = \sum_{i=0}^E (d^i - d^{i+1})$ is less than d^0 is no more than $e^{-2\left(1-\frac{1}{1+\epsilon}\right)^2 E} = e^{-\pi\frac{d^0\epsilon^2}{N(1+\epsilon)}}$. Thus, with very high probability, capture will occur within $2d^0(1+\epsilon)/N$ moves.

10 Conclusions

We first introduced a pursuit problem with multiple pursuers. We then identified the trivial cases in which the quarry could escape. For all non-trivial cases, we gave algorithms (PLANES and SPHERES) that caught the quarry in a number of steps linear in the initial distances. The analysis involved carefully estimating the progress made by any given pursuer and showing that the geometry of the situation forced that at every move, at least one pursuer makes significant progress. Then using a slightly different analysis, we showed that algorithm SPHERES generalizes to the case of obstacles too. In the most general case, SPHERES works in any convex playing field $\mathbb{R}^n$ with finitely many hyperplane boundaries, and $N > 0$ pursuers. Furthermore, the expected time to capture under any spherically symmetric random walk model is shown to be very small.

11 Acknowledgements

This work was supported in part by a grant from Tata Consultancy Services, Inc., by a matching grant from the Discovery Program of the University of California, and by contract F30602-01-2-0536 from the Defense Advanced Research Projects Agency.

References

[1] M. Adler, H. Räcke, N. Sivadasan, C. Sohler, and B. Vöcking. Randomized pursuit-evasion in graphs. *Combinatorics, Probability & Computing*, 12(3):225–244, 2003.

[2] L. Alonso, A. S. Goldstein, and E. M. Reingold. "Lion and man": upper and lower bounds. *ORSA J. Comput.*, 4(4):447–452, 1992.

[3] J. Farkas. Uber die theorie der einfachen ungleichungen. *J. Reine Angew. Math.*, 124:1–27, 1901.

[4] A. S. Goldstein. Unbounded unimodal search and pursuit problems. In *Ph.D. Thesis*, 1993.

[5] L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani. Visibility-based pursuit-evasion in a polygonal environment. In *Workshop on Algorithms and Data Structures*, pages 17–30, 1997.

[6] S.-M. P. Jae-Ha Lee and K.-Y. Chwa. Simple algorithms for searching a polygon with flashlights. *Information Processing Letters*, 81(5):265–270, 2002.

[7] V. Janković. About a man and lions. *Mat. Vesnik*, 2(15)(30)(4):359–361, 1978.

[8] P. Krapivsky and S. Redner. Kinetics of a diffusive capture process: Lamb besieged by a pride of lions. In *http://arxiv.org*, 1996.

[9] P. Krapivsky and S. Redner. Capture of the lamb: Diffusing predators seeking a diffusing prey. In *http://arxiv.org*, 1999.

[10] J. Littlewood. *Littlewood's Miscellany.* Cambridge University Press, Cambridge, 1986.

[11] J. Matousek. *Lectures on Discrete Geometry.* Springer-Verlag New York, Inc., Secaucus, NJ, 2002.

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

[13] J. Sgall. A solution to David Gale's lion and man problem. In *Theoretical Computer Science*, 2001.

[14] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry. Probabilistic pursuit-evasion games: Theory, implementation and experimental evaluation. In *IEEE Transactions on Robotics and Automation*, 2002.