Pursuit-evasion of an Evader by Multiple Pursuers

Alexander Von Moll*, David W. Casbeer†, and Eloy Garcia‡
Controls Science Center of Excellence
Air Force Research Laboratory
WPAFB, OH 45433
Email: *alexander.von_moll@us.af.mil,
†david.casbeer@us.af.mil,
‡eloy.garcia.2@us.af.mil

Dejan Milutinović
Department of Computer Engineering
UC Santa Cruz
Santa Cruz, CA 95064
Email: dmilutin@ucsc.edu

Abstract—In this paper, we extend the well-studied results of the two-pursuer, single-evader differential game to any number of pursuers. The main objective of this investigation is to exploit the benefits of cooperation amongst the pursuers in order to reduce the capture time of the evader. Computational complexity is a chief concern as this problem would need to be solved in an online fashion, e.g. in the case of autonomous unmanned aerial vehicles. A new geometric approach to solving the game is introduced and analyzed, which changes the problem of optimizing over continuous domains to a discrete combinatoric optimization. While past efforts at solving multiple pursuer problems have suffered from the curse of dimensionality, the geometric algorithms put forth here are shown to be scalable. Categorization and removal of redundant pursuers is the primary means by which scalability is achieved. The solution of this problem serves as a stepping stone to more complex problems such as the M-pursuer N-evader differential game.

I. Introduction

Pursuit-evasion scenarios have been extensively studied and they represent an important research area with applications to autonomous vehicles, robots, and systems. Several interesting pursuit-evasion problems were formulated in the seminal work by Isaacs [1]. Breakwell and Hagedorn [2] studied the dynamic game of a fast pursuer trying to capture in minimal time two slower evaders in succession. Motivated by the work in [2], the paper [3] analyzed the case where the fast pursuer tries to capture multiple evaders.

Games with multiple pursuers that try to capture an evader have also been addressed in [4], [5]. Thus, a dynamic Voronoi diagram has been used in scenarios with several pursuers in order to capture an evader within a bounded domain [5], [6]. The work in [7] considers a group of pursuers are assigned to intercept a set of evaders where the dynamics and goals of the evaders are assumed to be known to the pursuers. The so-called Prey, Protector, and Predator game was formulated in [8] in order to model rescue missions in the presence of obstacles. In this paper we address the problem of multiple pursuers trying to capture in minimum time a single evader.

The main result of this paper is to provide a solution to the M-pursuer one-evader differential game. This represents an important extension of the two-pursuer one-evader differential game (or the two cutters and a fugitive ship as labeled by Isaacs [1]) to the case of M-pursuers. Through explicit cooperation, the pursuers seek to capture the evader in minimum time, while the evader strives to maximize the capture time. The advantage of having multiple cooperative pursuers is that, depending on the initial positions of the players, the capture time in the M-pursuer one-evader differential game is less than the capture time without their cooperation. It will be shown that in this generalized problem a new type of solution appears with respect to the one-pursuer or the two-pursuer case. Depending on the initial positions of the agents, the evader's best strategy is to remain inside its dominance region and await capture. Otherwise, fleeing at max speed in any direction will be detrimental to the evader since the capture time will decrease.

It will also be shown that, in general, the pursuers can be assigned to different categories: interceptor, escort, and redundant. It is important to note that even though redundant pursuers are not essential to the differential game and they can be removed without affecting the outcome and value of the game, the solution provided in this paper is the solution to the fully cooperative M-pursuer one-evader differential game. The approach in this paper considers full cooperation among pursuers and is not based on decomposition to avoid the high-dimensionality of the state space of the differential game with multiple agents as in other approaches [9], [10], [11].

Finally, the solution of the M-pursuer one-evader differential game along with the categorization of pursuers, brings us one step closer to the seemingly intractable M-pursuer N-evader differential game. In such scenario it is necessary to consider the possible combination of assignments, which pursuers try to capture which evaders, in addition to the undesired property of very high dimensional state space. It is expected that the solution of the differential game studied in this paper will be fundamental in addressing the assignments required in the case of multiple evaders.

The remainder of the paper is organized as follows. Section II contains the problem formulation and Section IV refines the formulation into a solvable problem. Section V introduces a new geometric approach to solving the

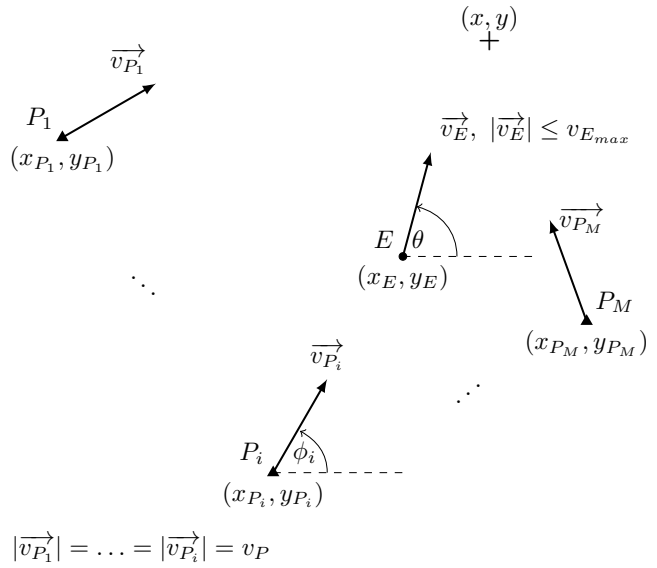


Fig. 1. Nomenclature and definitions for the M pursuer 1 evader problem. Pursuers are marked by triangles and the capture point is indicated by a +.

problem, Section VI specifies several useful algorithms, Section VII contains simulation results, and Section VIII contains the conclusion.

II. Problem Formulation

The problem of multiple pursuers capturing a single evader is considered in this paper in the setting of a two dimensional plane. Figure 1 contains the nomenclature and definitions for this problem. The objective of the pursuers is to cooperate in order to minimize the time to capture, while the objective of the evader is to maximize the time to capture (i.e. maximize the time of survival). We assume that the velocity of each pursuer, v_{P_i}, is greater than the velocity of the evader, v_E, so that capture can be guaranteed for any evader strategy. For reasons that will become clear later, the problem can be stated

Problem 1 (M Pursuer 1 Evader). Given the locations and speeds of each agent, find the point in the plane for the evader to travel such that it maximizes its capture time given that the pursuers' objective is to minimize this time.

We assume that both pursuers and evaders perform simple motions and that the pursuers all share the same velocity, v_P. Additionally, we assume that the pursuers' speed is fixed and is not a control input. From this point forward, without loss of generality, we consider the case when $(x_E, y_E) = (0, 0)$ and $v_{E_{\max}} = 1$. Thus, we redefine the pursuer velocities as a ratio: $\alpha_P = \frac{v_{E_{\max}}}{v_P}$, $\alpha_P \in (0, 1)$. The simple motions of E and the ith pursuer P_i are described by the following kinematic equations

$$\begin{aligned} \dot{x}_E &= \cos\theta,\ \dot{y}_E = \sin\theta,\ \theta \in [0, 2\pi) \\ \dot{x}_{P_i} &= \frac{\cos\phi_i}{\alpha_P},\ \dot{y}_{P_i} = \frac{\sin\phi_i}{\alpha_P},\ \phi_i \in [0, 2\pi), i = 1, \ldots, M \end{aligned} \tag{1}$$

where both θ and ϕ_i can change instantaneously and are generally considered to be functions of time. For convenience of notation, let $d(\cdot,\ \cdot)$ represent the Euclidean distance between two points. Additionally, let $E = (x_E, y_E)$ and $P_i = (x_{P_i}, y_{P_i})$.

III. Technical Preliminaries

In order to define the problem more formally, it is necessary to define capture. Let T represent the time of capture as a function of the initial conditions, the evader's target point, (x, y), and the heading angle policies of the pursuers:

$$T:\ \mathbb{R}^{2\times M+1} \times \mathbb{R}^2 \times \Phi^M \to \mathbb{R}^+$$

where the heading angle policies of the pursuers are represented as $\Phi_i = \phi_i(t),\ i = 1, \ldots, M,\ 0 < t$. Next, we say that capture occurs at the first time such that a pursuer's position coincides with the evader's position:

$$T = \min t\ s.t.\ \exists\ i\ s.t.\ (x_E(t), y_E(t)) = (x_{P_i}(t), y_{P_i}(t)) \tag{2}$$

If, for some pursuer policies $\Phi_i,\ i = 1, \ldots, M$ this condition is never satisfied, then $T = \infty$. The following constraints are also necessary for T to represent a valid capture time,

$$x^2 + y^2 \leq T^2 \tag{3}$$

$$\exists\ i\ s.t.\ (x - x_{P_i})^2 + (y - y_{P_i})^2 = T^2/\alpha_P^2 \tag{4}$$

Note that (3) ensures that the evader can reach the capture point (x, y) in T time. Similarly, (4) ensures that at least one pursuer reaches the capture point in exactly T time. The reason that the evader's constraint, (3), contains an inequality rather than equality is because the evader may need to travel to the point (x, y) and stop to maximize survival time, thereby reaching (x, y) in time less than T. Consider, for example, an evader surrounded by a ring ($M > 2$) of equidistant pursuers, as in Figure 2a. Pursuers are marked by a triangle and the evader is marked by a blue circle. The Apollonius circles are shown for each pursuer as a green circle and the solution point is marked by a +. In this case, the evader should remain at $(0, 0)$. A detailed analysis on the optimality of the evader's path to this point is left for future research; however, we note that the path is not unique as in [12]. Two particular equivalent interpretations to this idea of the evader stopping (i.e. $v_E \in \{0, 1\}$) are: (1) the evader chooses a speed to reach (x, y) in exactly t time (i.e. $v_E \in [0, 1]$) and (2) the evader, upon reaching (x, y), changes heading infinitely often such that its position remains at (x, y) (i.e. $v_E = 1$). In Figure 2b the evader's initial position is moved such

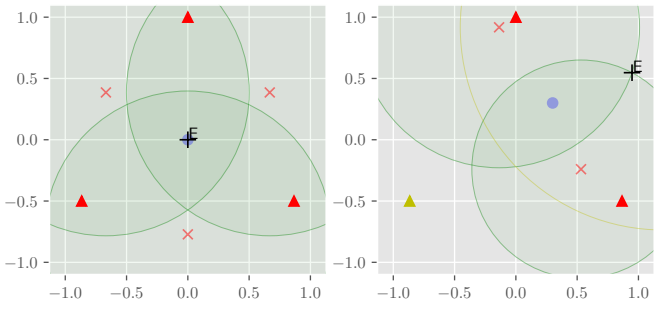


Fig. 2. Evader surrounded by a ring of pursuers (a) solution is to stand still and (b) offset of (0.3, 0.3) is applied to evader's initial position and solution degenerates to capture by two pursuers

that the solution degenerates to capture by two pursuers as described in [13].

IV. Methods of Solution

We now refine the statement of Problem 1 as

$$\max_{x,y} \min_{\Phi_1,\ldots,\Phi_M} T \quad \text{subject to (3), (4)} \tag{5}$$

In defining the problem this way, the evader is given freedom to select the capture point (x, y), while the pursuers are each free to select their heading angle policy, Φ. There are two issues with this definition. The first is that the pursuers may capture the evader before it could reach (x, y). A refinement of the constraints on (x, y) are necessary to prevent this possibility. The second is that (5) contains a minimization over multiple continuous functions which makes it difficult to reason about.

It is helpful to consider this problem in the context of regions of dominance in the $\mathbb{R}^2$ plane [14]. For two agents with identical speed, their respective regions of dominance are half-spaces partitioned by the bisector between their positions. The region of dominance represents the points in space that can be reached by a particular agent before any other agent, assuming a constant heading angle and maximum velocity. Boundaries of regions of dominance represent the points in space that can be reached simultaneously by the agents under the same assumptions. When the velocities of the agents are not identical, the region of dominance becomes an Apollonius circle $\mathcal{A}_i$. In this case, since $v_P > v_E$, the evader position x_E lies inside $\mathcal{A}_i$ and x_{P_i} lies outside $\mathcal{A}_i$. The set of the ith circle $\mathcal{A}_i$ is given by

$$\mathcal{A}_i = \left\{ \left(x - \frac{x_{P_i}\alpha_P^2}{\alpha_P^2 - 1}\right)^2 + \left(y - \frac{y_{P_i}\alpha_P^2}{\alpha_P^2 - 1}\right)^2 \leq \frac{(x_{P_i}^2 + y_{P_i}^2)\alpha_P^2}{(1-\alpha_P^2)^2} \middle| (x,y) \in \mathbb{R}^2 \right\} \tag{6}$$

which can be rewritten as

$$\mathcal{A}_i = \{(x-a_i)^2 + (y-b_i)^2 \leq R_i^2 \mid (x,y) \in \mathbb{R}^2\} \tag{7}$$

$$a_i = \frac{x_{P_i}\alpha_{P_i}^2}{\alpha_{P_i}^2 - 1}, \quad b_i = \frac{y_{P_i}\alpha_{P_i}^2}{\alpha_{P_i}^2 - 1}, \quad R_i = \sqrt{\frac{a_i^2}{\alpha_{P_i}^2} + \frac{b_i^2}{\alpha_{P_i}^2}} \tag{8}$$

Since $x_E = (0, 0) \in \mathcal{A}_i$ for all $i = 1, \ldots, M$, we can conclude that $x_E \in \cap_{i=1}^M \mathcal{A}_i$. Let $\overline{\mathcal{A}_i}$, the complement of $\mathcal{A}_i$ represent the region of dominance for pursuer i. Then the space $\cup_{i=1}^M \overline{\mathcal{A}_i}$ is dominated by one or more pursuers. Thus there must exist a strategy for the pursuers such that the evader's position (and thus the capture point) is confined to the non-pursuer-dominated region, $\cup_{i=1}^M \overline{\mathcal{A}_i} = \cap_{i=1}^M \mathcal{A}_i$. We now introduce the following additional constraint to (5)

$$(x, y) \in \mathcal{R} \tag{9}$$

using $\mathcal{R}$ to represent the evader-dominated region. Eq. (9) ensures that the evader can reach the target (x, y) before any pursuer can.

A. Optimal Pursuer Policy Assumption

As is, (5) is not very useful due to the fact that T is minimized over M continuous-time pursuer policies. We now state, without proof, the following

Proposition 1 (Optimal pursuer heading policies). The pursuers' optimal heading policy is to take a straight-line, constant-heading path to (x^*, y^*) where

$$(x^*, y^*) = \arg\max_{x,y} \min_{\Phi_1,\ldots,\Phi_M} T$$

Thus the pursuers' optimal policy can be written as

$$\Phi_i^* = \phi_i^*(t) = \tan^{-1}\frac{y^* - y_{P_i}}{x^* - x_{P_i}}, \quad i = 1, \ldots, M, \ t \geq 0$$

Note that because the capture point is constrained to be inside the evader-dominated region, $\mathcal{R}$, the evader feasibility constraint (3) will always be satisfied. Combining these results with (2), (5), and (9) yields the following

$$\max_{(x,y)\in\mathcal{R}} \min_{i\in\{1,\ldots,M\}} d((x,y),\ P_i)\alpha_P \tag{10}$$

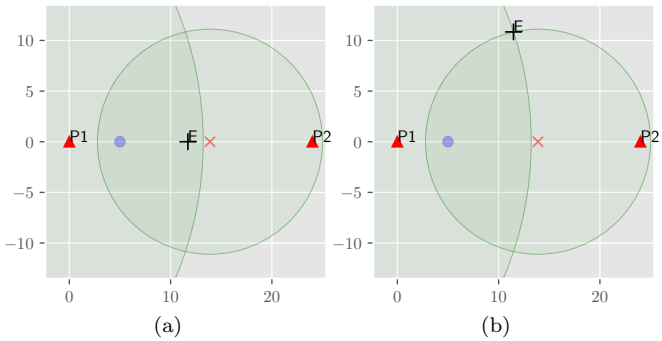
B. Numerical Optimization

One final reformulation is necessary to pose the problem expressed in (10) as an optimization problem. Let m be a lower bound on the traversal times of all the pursuers to the point (x, y):

$$m \leq d((x,y),\ P_i)\alpha_P, \quad i = 1, \ldots, M$$
$$\implies m \leq \min_{i\in\{1,\ldots,M\}} d((x,y),\ P_i)\alpha_P$$

Then (10) can be rewritten as the following linear program

$$\begin{aligned} &\max_{x,y} m && \\ s.t. \quad &d((x,y),\ (a_i,b_i)) \leq R_i && i = 1, \ldots, M \\ &m \leq d((x,y),\ P_i)\alpha_P && i = 1, \ldots, M \\ &m \geq 0 && \end{aligned} \tag{11}$$



(a) (b)

Fig. 3. Numerical solution of collinear pursuers (a) initial guess set at $(x, y) = (0, 0)$ and $m = 1$ results in a suboptimal capture on the line passing through the agents (a local optimum), (b) initial guess set at $(x, y) = (5, 5)$ and $m = 5$ results in an optimal capture at the intersection of the pursuers' Apollonius circles

Note that the constraints are quadratic and non-convex which prohibits the use of techniques used to solve linear, quadratic, or second order cone problems exactly [15]. Thus general-purpose solvers must be used. Issues arise, however, due to the shape of the objective function in (11). Figure 3 contains an example with 2 pursuers who are collinear with the evader. Figure 3b shows the true, optimal solution at the upper intersection of the pursuers' Apollonius solution which is equivalent to the lower intersection. The optimization suffers from sensitivity to the initial guess as shown in Figure 3a, where the solver found a local optimum. Systematically computing all of the solutions in the cases where multiple solutions exist would be costly and potentially error prone. Also, determining whether there are multiple solutions would be difficult.

V. Geometric Approach

We now return to the notion of regions of dominance and discuss the utility of the Voronoi diagram in the context of this problem as has been noted in [14], [16], [5], [17], [18]. The Voronoi diagram, in 2D space, defines a tessellation in which each agent resides in their own cell defining points in space they can reach before any other agent. Voronoi diagrams are typically parameterized by a set of vertices, edges, and generator points (or agent positions in the present case): $\mathcal{V} = (V, \mathcal{E}, X)$. Let us define two useful Voronoi diagrams for the M pursuer 1 evader problem,

$$\mathcal{V}_E = (V_E,\ \mathcal{E}_E,\ \{E, P_1, \ldots, P_M\})$$
$$\mathcal{V}_P = (V_P,\ \mathcal{E}_P,\ \{P_1, \ldots, P_M\})$$

one which includes the evader as a generator point along with all of pursuers, $\mathcal{V}_E$, and one which does not include the evader, $\mathcal{V}_P$. The edges $\mathcal{E}_P$ are all straight because the pursuers share the same velocity, however, the edges in $\mathcal{E}_E$ surrounding the evader are curved (concave w.r.t. evader) due to the evader's inferior speed. Thus $\mathcal{V}_E$ is in fact a multiplicatively weighted Voronoi diagram where the agents' velocities represent the weights [19]. Also note that the evader's cell of $\mathcal{V}_E$ is exactly $\mathcal{R}$; and this cell's edges correspond to segments of the pursuers' Apollonius circles while the vertices correspond to intersections of Apollonius circles. The evader's dominance region $\mathcal{R}$ can be parameterized as an ordered set of vertices and an ordered set of arcs between the vertices: $\mathcal{R} = (V_\mathcal{R},\ \mathcal{E}_\mathcal{R})$, where $\mathcal{E}_\mathcal{R} \subset \mathcal{E}_E$ and $V_\mathcal{R} \subset V_E$.

A. Types of Solutions

In the case of a single pursuer ($i = 1$), the resulting solution is well-known [1]: the evader should flee directly away from the pursuer. Thus capture occurs at

$$(x^*, y^*) = R_i(1 + \alpha_P)\frac{(E - P_i)}{d(E,\ P_i)} \tag{12}$$
$$t^* = d((x^*, y^*), P_i)\alpha_P \tag{13}$$

Single pursuer capture occurs on an arc of the evader's dominance region, $(x^*, y^*) \in \mathcal{E}_\mathcal{R}$.

The two pursuer case was also presented by Isaacs as the two cutters and fugitive ship problem [1]. Recent work on this same problem has shown that the solution(s) to this game for the case of simultaneous capture by two pursuers can be obtained from the intersection of their Apollonius circles [13]. Intersections of pursuers' Apollonius circles correspond to vertices of $\mathcal{V}_E$ in this case, thus $(x^*, y^*) \in V_\mathcal{R}$. Also note that these intersections also lie on edges of $\mathcal{V}_P$.

In the case of $M \geq 3$, the solution can of course degenerate to one of the previous cases. However, a new type of solution arises which occurs in the interior of the region $\mathcal{R}$. The solution to (10) may occur on the interior of the evader's region of dominance $\mathcal{R}$. If this is the case, we believe the solution is necessarily a vertex of the Voronoi diagram excluding the evader:

$$(x^*, y^*) \notin \partial\mathcal{R} \implies (x^*, y^*) \in V_P \tag{14}$$

A solution of this nature requires at least three pursuers to participate in the capture. Although we do not prove this assumption here, some intuition can be built up around the idea. As we have restricted the velocity of the pursuers to be constant, a vertex of $\mathcal{V}_P$ is located at a point that is equidistant from the three (or more) nearest pursuers [19]. Thus simultaneous capture by three or more pursuers occurs at a vertex of $\mathcal{V}_P$ and this point is inside $\mathcal{R}$ due to (9). The main result is that one can find solutions to (10) by searching over a finite number of candidate solutions,

Theorem 1.

$$(x^*, y^*) \in \mathcal{S}_\mathcal{R} \cup V_\mathcal{R} \cup V_{P_\mathcal{R}} \tag{15}$$

where

$$
\begin{aligned}
(x^*, y^*) &= \arg \max_{(x,y)\in\mathcal{R}} \min_{i\in\{1,\dots,M\}} d((x,y),\ P_i)\alpha_P, \\
\mathcal{S}_{\mathcal{R}} &= \left\{ x_i, y_i \middle| x_i, y_i = R_i(1+\alpha_P)\frac{E-P_i}{d(E,\ P_i)}, \right. \\
&\qquad \left. x_i, y_i \in \mathcal{R},\ i = 1,\dots,M \right\}, \\
V_{P_{\mathcal{R}}} &= V_P \cap \mathcal{R}
\end{aligned}
\tag{16}
$$

Proof. The solution to the M pursuer 1 evader problem is one of the following cases:
Case 1. $(x^*, y^*) \in \mathcal{S}_{\mathcal{R}}$. Capture is by a single pursuer ($\mathcal{S}_{\mathcal{R}}$ representing single-pursuer solutions inside the evader's dominance region). The solution is then given by (12) according to [1].
Case 2. $(x^*, y^*) \in V_{\mathcal{R}}$. Capture is by two pursuers simultaneously, which occurs in the set $V_{\mathcal{R}}$ according to [13].
Case 3. $(x^*, y^*) \in V_{P_{\mathcal{R}}}$. Capture is by three or more pursuers simultaneously, which occurs in the set $V_P \cap \mathcal{R}$ by the preceding assumption. □

B. Categories of Pursuers

The set of pursuers can be broken up into four distinct categories (or sets):
Category 1. $\mathcal{I}_1 = \{i | P_i(t^*) = E(t^*) = (x^*, y^*)\}$, "interceptors". Pursuers who participate in capturing the evader according to the solution.
Category 2. $\mathcal{I}_2$, "escorts". Pursuers who cannot reach the solution in t^* time but constrain the region $\mathcal{R}$. In other words, pursuers who are neighbors of the evader in $\mathcal{V}_E$. Additionally, removal of any of these pursuers from the game would change the solution and increase capture time.
Category 3. $\mathcal{I}_3$, "redundant (a)". Pursuers who fit the criteria of $\mathcal{I}_2$, but whose removal from the game would not change the solution or increase capture time.
Category 4. $\mathcal{I}_4$, "redundant (b)". Pursuers who do not share an edge with (i.e. are not neighbors of) the evader in $\mathcal{V}_E$.

Note that both $\mathcal{I}_3$ and $\mathcal{I}_4$ are redundant in the sense that if the pursuers in the first two categories behave optimally, then pursuers in $\mathcal{I}_3 \cup \mathcal{I}_4$ cannot affect the solution. The meaning of an optimal policy for escorts is difficult to define because their role in pursuit is to shape the evader's region of dominance. For the purposes of the present discussion we assume the escorts head towards the solution point at maximum speed. A rigorous treatment of an escort's possible range of optimal policies is left for future research. Figure 4 provides an example illustrating each of these categories. In this example, capture is achieved simultaneously by P1 and P8. P3's removal from the game would cause the evader to flee towards the lower left, so P3 must participate by shaping $\mathcal{R}$. P2 is a neighbor of E in $\mathcal{V}_E$ which is analogous to saying its Apollonius circle partially defines the boundary of $\mathcal{R}$. Removal of P2 from the game makes

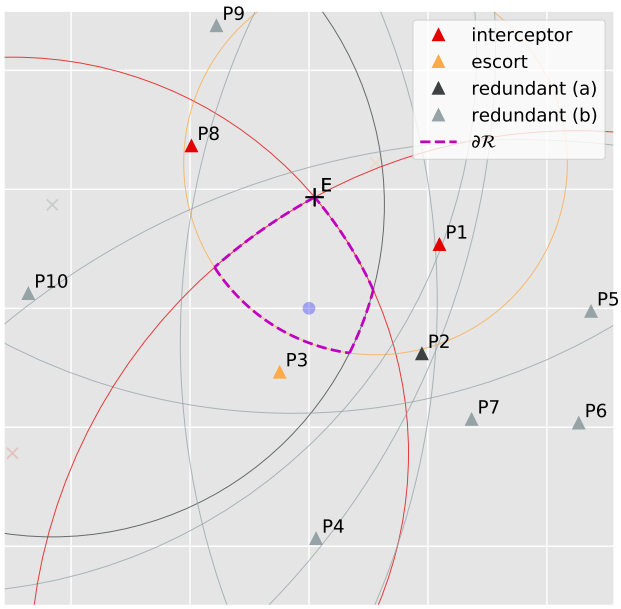


Fig. 4. Categories of pursuers with their Apollonius circles drawn, and the boundary of the evader's region of dominance is shown in dashed magenta.

available additional intersections of Apollonius circles as candidate solutions, however these points are suboptimal. Thus P2 does not affect the solution as long as P1, P3, and P8 behave optimally.

An important distinction between the $\mathcal{I}_3$ and $\mathcal{I}_4$ categories is that a pursuer can only be identified as belonging to $\mathcal{I}_3$ by finding two solutions (one including the pursuer and one without). However, $\mathcal{I}_4$ pursuers can be identified, and thus discarded, prior to solving the problem. Figure 4 shows that the Apollonius circles of redundant (b) pursuers completely contain $\mathcal{R}$. Out of the ten pursuers in this example, only four need to be considered. This provides the motivation for the following section which includes an algorithm for computing this reduced set of pursuers, $\mathcal{I}_{\mathcal{R}} = \mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_3$.

VI. Geometric Algorithms and Improving Computational Complexity

For very large numbers of randomly placed pursuers, it has been observed that the size of the reduced pursuer set, $\mathcal{I}_{\mathcal{R}}$, remains close to four unless the pursuers' positions are highly correlated. Therefore, if $\mathcal{R}$ and $\mathcal{I}_{\mathcal{R}}$ can be computed quickly (that is, in a scalable manner), then solving Problem 1 is also scalable under these assumptions. In other words, the size of the set of candidate solutions in (15) only depends on the size of $\mathcal{I}_{\mathcal{R}}$.

A. Computing $\mathcal{R}$ and $\mathcal{I}_{\mathcal{R}}$

The evader's dominance region is, by definition, a single cell of the multiplicatively weighted Voronoi diagram (MWVD) of all the agents. Computing the entire

Require: $\alpha_P < 1$

1: function EvaderCell(E, P_1, …, P_M, α_P)
2: Let q be a priority queue of tuples, prioritized by the first element
3: for all P_i do
4: $C \leftarrow \text{Apollonius}(P_i, \alpha_P)$
5: $\underline{d} \leftarrow \min_\theta \text{Dist}(E, \partial\mathcal{A})$
6: q.Enqueue($\underline{d}, \mathcal{A}, P_i, i$)
7: $\underline{d}, \mathcal{A}, P \leftarrow q$.Dequeue()
8: $V \leftarrow \{\}$ ▷ cell vertices
9: $\mathcal{E} \leftarrow \{C\}$ ▷ cell edges
10: Let $\mathcal{R} = (V, \mathcal{E})$ represent the state of the cell
11: $\mathcal{I}_\mathcal{R} \leftarrow \{P\}$
12: $d^\dagger \leftarrow \max_\theta \text{Dist}(E, \partial\mathcal{A})$
13: $\underline{d}, \mathcal{A}, P \leftarrow q$.Dequeue()
14: while $\underline{d} < d^\dagger$ and $q \neq \emptyset$ do
15: $U \leftarrow \{\partial\mathcal{A} \cap \partial\mathcal{A}_j,\ \forall \mathcal{A}_j \in \mathcal{E}\}$
16: if $\exists u \in U\ s.t.\ u \in \mathcal{R}$ then
17: $V \leftarrow V \cup (U \cap \mathcal{R})$ ▷ add new vertices
18: $V \leftarrow V \setminus (\overline{V \cap C})$ ▷ prune invalid vertices
19: Recompute $\mathcal{E}$ and $\mathcal{I}_\mathcal{R}$
20: $d^\dagger \leftarrow \max_\theta \text{Dist}(E, \partial\mathcal{R})$
21: $\underline{d}, \mathcal{A}, P \leftarrow q$.Dequeue()
22: return $\mathcal{R},\ \mathcal{I}_\mathcal{R}$

MWVD is computationally prohibitive for large numbers of agents since an optimal algorithm is necessarily $O(n^2)$ [20]. Even computing the entire MWVD for the reduced pursuer set is unnecessary. We now introduce an algorithm for computing $\mathcal{R}$ and $\mathcal{I}_\mathcal{R}$ (on pg. 6), followed by a discussion of its design and correctness.

The EvaderCell algorithm constructs the region $\mathcal{R}$ incrementally starting with a single Apollonius circle and then subsequently computing the intersection of the region with one new Apollonius circle at a time. Line 15 computes the intersections of the new circle with each circle that comprises the set of arcs defining $\mathcal{R}$. Then, in line 16, if we find that some of these intersections lie inside $\mathcal{R}$, then we know the new circle must be added and $\mathcal{R}$ recomputed. It is possible that by adding this new arc that previous arcs are now outside the current region and must be pruned. A naive approach to this algorithm would be to perform this procedure for every pursuer in the set. We can do better, however, by prioritizing the order in which we process the pursuers. If we order the pursuers by increasing distance from the evader to the closest point on its Apollonius circle we have a metric that is monotonically nondecreasing as we iterate. Meanwhile, the furthest distance from the evader to a point on $\partial\mathcal{R}$ is monotonically nonincreasing as $\mathcal{R}$ is pared down each iteration. Once the closest point on a circle is further than the furthest point on $\partial\mathcal{R}$ we need not consider any additional pursuers.

Lemma 1. The nearest pursuer to E is also a neighbor of E in $\mathcal{V}_E$. This pursuer's Apollonius circle contributes an arc to $\mathcal{R}$ and also corresponds to the pursuer whose Apollonius circle has a point closest to the evader (i.e. the pursuer returned in Line 7).

Proof. The first statement of the lemma follows from the fact that the pursuers share the same velocity.

$$
\begin{aligned}
&\arg\min_i \min_\theta d(E,\ \partial\mathcal{A}_i) \\
&\quad = \arg\min_i R_i - d(E,\ (a_i, b_i)) \\
&\quad = \arg\min_i R_i^2 - d^2(E,\ (a_i, b_i)) \\
&\quad = \arg\min_i \frac{(x_{P_i}^2 + y_{P_i}^2)\alpha_P^2}{(1-\alpha_P^2)^2} \\
&\qquad\qquad - \left(\left(\frac{x_{P_i}\alpha_P^2}{\alpha_P^2 - 1} - 0\right)^2 + \left(\frac{y_{P_i}\alpha_P^2}{\alpha_P^2 - 1} - 0\right)^2\right) \\
&\quad = \arg\min_i (x_{P_i}^2 + y_{P_i}^2)\frac{\alpha_P^2}{1-\alpha_p^2} \\
&\quad = \arg\min_i d(E,\ P_i)
\end{aligned}
$$

□

Lemma 1 supports the initialization of the iterations in EvaderCell.

Lemma 2. The value of $\underline{d}$ is monotonically nondecreasing with the iterations.

Proof. Once all pursuers have been enqueued to q, the queue property of q ensures the value $\underline{d}$ of successively dequeued elements can only increase or stay the same. □

In order to prove the monotonicity of $d^\dagger$ we must make some additional statements about the properties of $\partial\mathcal{R}$ and how it is recomputed at each iteration.

Lemma 3. The point corresponding to the maximum distance from E to $\partial\mathcal{R}$ is in the current set of vertices or valid single-pursuer solutions.

$$\max_\theta \text{Dist}(E,\ \partial\mathcal{R}) = \max_v \text{Dist}(E,\ v),\ v \in V \cup \mathcal{S}_\mathcal{R}$$

Proof. First note that each single-pursuer solution is the furthest point on that pursuer's Apollonius circle. Let k represent the number of arcs that comprise $\partial\mathcal{R}$.
Case 1. $k = 1$. The cell is defined by a single circle whose furthest point corresponds to a single-pursuer solution.
Case 2. $k = 2$. The second Apollonius circle intersects the first at two points. After recomputing $\mathcal{R}$ there are three possibilities:
Case 2.a. The single-pursuer solution from Case 1 is still in $\mathcal{R}$, thus it is the furthest point.
Case 2.b. The single-pursuer solution of the second Apollonius circle is in $\mathcal{R}$, thus it is the furthest point.
Case 2.c Neither single-pursuer solution is in $\mathcal{R}$. Note the distance to each Apollonius circle decreases monotonically away from its respective single-pursuer solution. Thus the maximum distance must occur at a point along the boundary that is closest to the single-pursuer

solution, which is necessarily an intersection (or vertex in V).
Case 3. $k > 2$. These cases follow similar logic to Case 2. □

Lemma 4. The value of $d^\dagger$ is monotonically nonincreasing with the iterations.

Proof. During each iteration there are two cases:
Case 1. None of the candidate intersections, U, lie inside the current region $\mathcal{R}$. Then, $d^\dagger$ is not updated and its value remains the same.
Case 2. At least one candidate intersection, $u \in U$, lies inside the current region $\mathcal{R}$. The vertices that get added to V are necessarily inside $\mathcal{R}$ and thus cannot be further away than the value of $d^\dagger$. Similarly, if a single-pursuer solution of this pursuer or any previous pursuer lies inside $\mathcal{R}$ it also cannot be further away than the value of $d^\dagger$. Then by Lemma 3, no other point needs consideration. □

Lemma 5. When $\underline{d} \geq d^\dagger$ the EvaderCell algorithm is finished because pursuers in q do not intersect the region $\mathcal{R}$.

Proof.

$$\underline{d} = \min_{\theta} \text{Dist}(E, \partial\mathcal{A}) \implies \underline{d} \leq \text{Dist}(E, \partial\mathcal{A}) \ \forall \ \theta \quad (17)$$

However, we know the point corresponding to $\underline{d}$ is unique unless $(a, b) = E$, so the inequality is strict. Similarly, we have

$$d^\dagger = \max_{\theta} \text{Dist}(E, \partial\mathcal{R}) \implies d^\dagger \geq \text{Dist}(E, \partial\mathcal{R}) \ \forall \ \theta$$

When $\underline{d} = d^\dagger$, the new Apollonius circle $\mathcal{A}$ is tangent to $\mathcal{R}$ at one or more locations (depending on whether or not the previous inequality is strict). From (17), every point on $\partial\mathcal{A}$ is as far or further than the point corresponding to $\underline{d}$ meaning no point on $\partial\mathcal{A}$ can be inside $\mathcal{R}$:

$$\begin{aligned} \underline{d} \geq d^\dagger &\implies \text{Dist}(E, \partial\mathcal{A}) \geq d^\dagger \ \forall \ \theta \\ &\implies \partial\mathcal{A} \cap \partial\mathcal{R} = \emptyset \end{aligned}$$

Furthermore, from Lemmas 2 and 4, the values of $\underline{d}$ and $d^\dagger$ cannot get any closer, thus we have,

$$\underline{d} \geq d^\dagger \implies \partial\mathcal{A}_i \cap \partial\mathcal{R} = \emptyset \ \forall \ \mathcal{A}_i \in q$$

□

Figure 5 highlights the benefit of the ordering of the pursuers in EvaderCell. In this example, there are 10 pursuers, however, only the first seven need to be considered based on the criteria established in this section.

Lemmas 1 – 5 show the correctness of EvaderCell. There are several other interesting properties of $\mathcal{S}_\mathcal{R}$ that are worth mentioning here.

Lemma 6. The number of single-pursuer solutions inside the evader dominance region, $\mathcal{S}_\mathcal{R}$ is either 0 or 1,

$$|\mathcal{S}_\mathcal{R}| \in \{0, 1\}$$

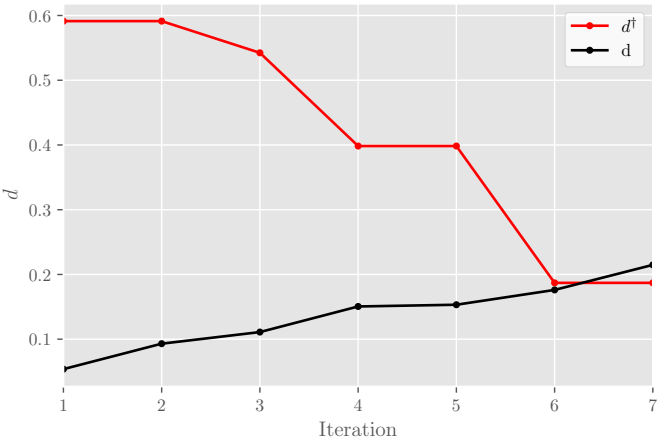


Fig. 5. Comparison of the values of $\underline{d}$ and $d^\dagger$ by iteration for the example shown in Figure 4

Proof. Suppose $\exists s_i \in \mathcal{S}_\mathcal{R}$, that is, there is at least one single-pursuer solution in $\mathcal{R}$. The point s_i is the furthest point on the corresponding pursuer's Apollonius circle $\mathcal{A}_i$. Also, $\mathcal{A}_i \supset \mathcal{R}$ because $\mathcal{R} = \cap_{k=1}^{M} \mathcal{A}_k$. Thus since $s_i \in \mathcal{R}$ it must also be the furthest point in $\mathcal{R}$. Then suppose $\exists s_j \in \mathcal{S}_\mathcal{R}, \ s_j \neq s_i$. There are two cases:
Case 1. $d(E, \ s_i) < d(E, \ s_j)$. This is a contradiction since s_i is the furthest point in $\mathcal{R}$.
Case 2. $d(E, \ s_i) > d(E, \ s_j)$. Using the same arguments as above, the fact that $s_j \in \mathcal{S}_\mathcal{R}$ means it should be the furthest point in $\mathcal{R}$, which contradicts the premise of this case. □

Lemma 7. If $\exists \ s^* \in \mathcal{S}_\mathcal{R}$, then s^* is closest single-pursuer solution to E,

$$s^* \in \mathcal{S}_\mathcal{R} \implies s^* = \arg\min_{s_i} d(E, \ s_i)$$

where s_i is described by (12).

The proof is similar to that in Lemma 6 and is omitted.

Lemma 8. If $\exists \ s^* \in \mathcal{S}_\mathcal{R}$ then s^* is the solution to the M pursuer 1 evader problem (Problem 1 and (10)).

Proof. First note that $s^* \in \mathcal{S}_\mathcal{R} \implies s^* \in \mathcal{R}$ (from (16)). Let P^* represent the pursuer corresponding to s^* according to (12). From [1], we know that every point that is not s^* is suboptimal for the evader, that is,

$$T(x, y) < t^* \ \forall \ x, y \in \mathbb{R}^2, \ x, y \neq s^* \quad (18)$$

Thus if the evader chooses any other point in $\mathcal{R}$ the capture time will decrease. Also, the evader is able to reach this point without being intercepted en route because it lies on the boundary of its region of dominance. □

1) Complexity: For the purposes of the analysis in this section, let k represent the number of iterations completed in EvaderCell.

Lemma 9. The EvaderCell algorithm has time complexity $O(M + k \log M + k^2)$. When $k \ll M$ the complexity becomes $O(M)$.

Proof. First, q is constructed in $O(M)$ time. Dequeuing an element from a priority queue generally takes (depending on the specific implementation) $O(\log M)$ time where M is the size of the queue. There are $k+2$ total dequeue operations in EvaderCell with total time complexity $O(k \log M)$. The complexity of intersecting the new Apollonius circle with the circles defining $\mathcal{R}$ in Line 15 is $O(k)$, and is computed at each iteration giving total complexity $O(k^2)$. Checking whether an intersection point u is inside the region $\mathcal{R}$ can be done in $O(k)$ time due to the parameterization of $\mathcal{R}$ as an ordered set of vertices and arcs. Note that the Dist function can be computed in constant time when the second argument is a circle. In the case of Line 20 where the second argument is the boundary $\partial\mathcal{R}$, some additional reasoning is needed. From Lemma 3 the maximum distance to $\partial\mathcal{R}$ corresponds to one of the vertices defining $\mathcal{R}$ or a single-pursuer solution in $\mathcal{R}$. The size of the set of vertices, $|V|$, is on the order of k; and the number of single-pursuer solutions to consider is 0 or 1 from Lemma 6. Thus the total time complexity of Line 20 is $O(k^2)$. When $k \ll M$, the M term dominates the $k \log M$ and k^2 terms. □

In the worst case we must iterate through all of the pursuers, thus $k = M$ and EvaderCell has time complexity $O(M^2)$. A more general algorithm for representing the intersection of M disks as a convex polytope has been shown to have complexity of $\Theta(M \log M)$ [21]. This can be used to construct an alternative algorithm (shown on pg. 8) with time complexity $\Theta(M \log M)$ which performs better in the worst case.

1: function EvaderCell2(E, P_1, …, P_M, α_P)
2: Compute Apollonius circles
3: Represent intersection of disks as the intersection of halfspaces via inversion transform [21]
4: Apply duality transform and compute convex hull [22]
5: Apply duality transform on convex hull to get polyhedron $\mathcal{P}$
6: Intersect $\mathcal{P}$ with sphere corresponding to XY plane [21]
7: Invert intersection to get $\mathcal{R}$
8: return $\mathcal{R}$

To justify its use despite poor worst-case performance, we ran EvaderCell over a range of M with different random distributions over the pursuer positions and $\alpha_P = 0.5$. Figure 6 shows a weak correlation (Pearson correlation coefficient of 0.08) between k and M. In general, k remains relatively small even as M reaches numbers in the thousands. However, when the pursuer positions are highly correlated then k and M are also highly correlated (Pearson correlation coefficient of 1) as seen in Figure 7.

B. M Pursuer 1 Evader Algorithm

Once $\mathcal{R}$ and $\mathcal{I}_\mathcal{R}$ have been computed, solving Problem 1 and (10) is straightforward. Lemma 8 already provides one type of solution which is capture by a single pursuer. This type of solution can be easily tracked by modifying EvaderCell to keep a record of the pursuer with the closest single-pursuer solution distance (i.e. (12)) to E. From Theorem 1, the two other sets of potential solutions are $V_\mathcal{R}$, which is computed in EvaderCell, and $V_{P_\mathcal{R}}$. Fortune's Algorithm can be used to compute the Voronoi diagram on the reduced pursuer set, $\mathcal{I}_\mathcal{R}$ [23]. Then, computing $V_{P_\mathcal{R}}$ is a matter of checking each vertex of the Voronoi diagram to see if it is inside $\mathcal{R}$. The following algorithm computes solutions to the M pursuer 1 evader problem.

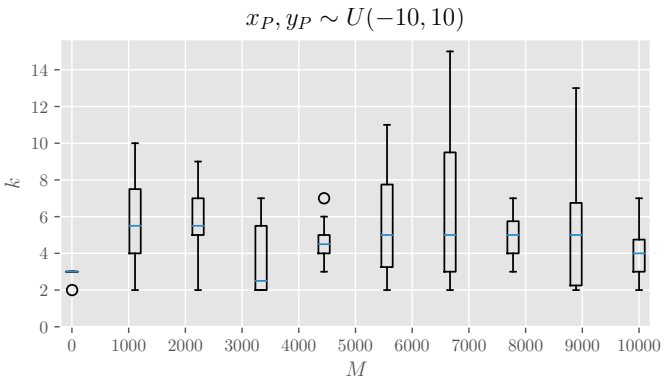


Fig. 6. Simulation results of EvaderCell, k versus M for $\alpha_P = 0.5$ and uncorrelated pursuer positions: $x_P, y_P \sim Uniform(-10,\ 10)$

1) Complexity:

Theorem 2. The Mpursuer1evader algorithm has time complexity of $O(M + k \log M + k^2)$.

Proof. The call to EvaderCell in Line 2 has time complexity of $O(M + k \log M + k^2)$ from Lemma 9. Tracking $\mathcal{S}_\mathcal{R}$ in EvaderCell can be done in constant time for each iteration. The call to Fortune's algorithm in Line 8 has time complexity of $O(|\mathcal{I}_\mathcal{R}| \log |\mathcal{I}_\mathcal{R}|)$ [23]. Note that $|\mathcal{I}_\mathcal{R}| \leq k$ because in each iteration of EvaderCell only one pursuer may be added, however one or more may be removed. Thus the time complexity of Line 8 is no worse than the time complexity of Line 2. The size of

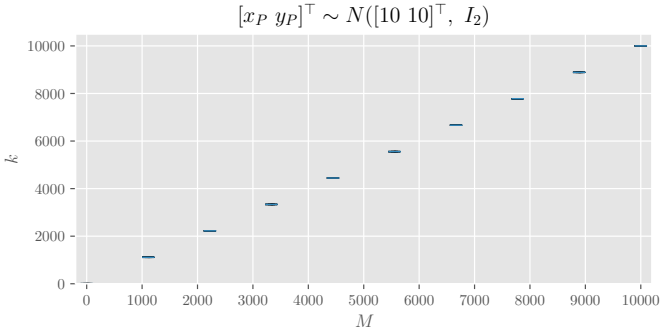


Fig. 7. Simulation results of EvaderCell, k versus M for $\alpha_P = 0.5$ and highly correlated pursuer positions: $[x\ y]^\top \sim Normal([10\ 10]^\top,\ I_2)$

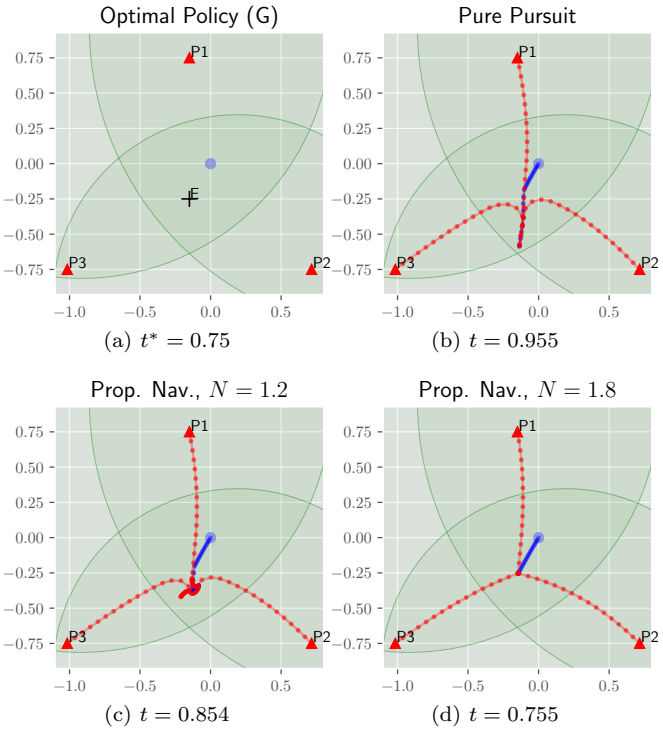
1: function Mpursuer1evader(E, P_1, …, P_M, α_P)
2: $\mathcal{R}, \mathcal{I}_\mathcal{R}, \mathcal{S}_\mathcal{R} \leftarrow$ EvaderCell($E, P_1, \ldots, P_M, \alpha_P$)
3: if $\exists s^* \in \mathcal{S}_\mathcal{R}$ then
4: $(x^*, y^*) \leftarrow s^*$
5: Let P^* be the pursuer corresponding to s^*
6: $t^* \leftarrow d((x^*, y^*),\ P^*)\alpha_P$
7: else
8: $V_P \leftarrow$ Fortune($\mathcal{I}_\mathcal{R}$) ▷ [23]
9: $V_{P_\mathcal{R}} \leftarrow V_P \cap \mathcal{R}$
10: $V \leftarrow V_{P_\mathcal{R}} \cup V_\mathcal{R}$
11: $(x^*, y^*) \leftarrow \arg\max_{(x,y)\in V}$ $\min_{P_i \in \mathcal{I}_\mathcal{R}} d((x, y),\ P_i)\alpha_P$
12: $t^* \leftarrow \max_{(x,y)\in V} \min_{P_i \in \mathcal{I}_\mathcal{R}} d((x, y),\ P_i)\alpha_P$
13: return $(x^*, y^*),\ t^*$

the set of vertices returned from Fortune's algorithm is at worst $O(|\mathcal{I}_\mathcal{R}|)$ [23]. Checking to see if a point is inside $\mathcal{R}$ is at worst $O(k)$ (cf. proof of Lemma 9). Thus the total time complexity of Line 9 is at worst $O(k^2)$. It is simple to see that $|V| = O(k)$ since $|V_{P_\mathcal{R}}| = O(|\mathcal{I}_\mathcal{R}|)$ and $|V_\mathcal{R}| = O(k)$. Finally, the number of values searched over in Lines 11 and 12 is $|V| \cdot |\mathcal{I}_\mathcal{R}|$, yielding worst-case time complexity of $O(k^2)$. □

VII. Results

The overall objective of this investigation is to utilize cooperation amongst agents on the same team. In this case the pursuers should cooperate in order to reduce the capture time of the evader compared to approaches where each pursuer is selecting its heading independent of the other pursuers. Pure pursuit (PP) and proportional navigation (PN) are two common policies for an agent pursuing a target. These policies are simply functions of the state of the pursuer implementing the policy and the target's current state (position and velocity). The approach outlined in this paper will be referred to as the geometric (G) policy. In order to compare these three policies, it is necessary to construct a discrete-time simulation wherein each policy is implemented in a feedback fashion (i.e. the states are observed by every agent and used to compute its control input at each time instant). Agents who implement the G policy will move towards the solution point (x^*, y^*) computed in that time instant. For each simulation, the evader implements the G policy by computing a velocity $0 \leq v_E \leq v_{E_{max}}$ such that the evader reaches the solution point at the capture time t^*. Note that in the case where the pursuers implement the G policy, all the agents move towards the solution point and thus the point does not change over time; thus simulation of this scenario is trivial as the capture time must be t^*. However, in cases where the pursuers do not implement the G policy, the solution point (x^*, y^*) may jump discontinously thereby causing the evader to change heading abruptly.

Figure 8 show the trajectories for the simulations of each policy. The time step in all the simulations is



(a) $t^* = 0.75$ (b) $t = 0.955$
(c) $t = 0.854$ (d) $t = 0.755$

Fig. 8. Simulation results for different pursuer policies against an evader implementing the optimal (G) policy.

TABLE I
Capture times between the 3 different policies

Policy	Capture Time	t/t^*
PP	0.955	1.273
PN, N=1.2	0.854	1.139
PN, N=1.8	0.755	1.007
G	0.750	1.000

0.05 seconds and the initial conditions are the same. Note that two simulations were done for the PN policy because of the sensitivity of the pursuer trajectories to the navigation constant N (c.f. [24]). It is evident that the PN policy struggles in the presence of abrupt heading changes by the evader. When it is not advantageous for the evader to switch headings, as in Figure 8d, the result is similar to the G policy. As seen in Table I, the G policy outperforms PP and PN. Thus there is a clear benefit to employing a cooperative pursuer strategy over one where the pursuers behave independently.

VIII. Conclusions

In this paper, the two cutters and fugitive ship problem posed by Isaacs [1] and solved by Garcia et al. [13] was extended to consider any number of pursuers. The main simplification used was that the optimal pursuer heading policies result in constant-heading (straight-line) paths, which was proven to be true in both the single and two pursuer cases. We identified three possible solution types for the M pursuer 1 evader problem: capture by a single pursuer, capture by two pursuers

simultaneously at the intersection of their Apollonius circles, and capture by three or more pursuers at a vertex of the pursuer-pursuer Voronoi diagram. Four unique categories of pursuers were identified: interceptors (pursuers who capture the evader), escorts (pursuers who constrain the evader's dominance region), and two types of redundant pursuers who do not affect the outcome of the game. Two scalable geometric algorithms were presented: EvaderCell, which computes the evader's region of dominance, and Mpursuer1evader which computes the solution. Both algorithms have worst-case time complexity of $O(M + k \log M + k^2)$, where M is the number of pursuers and k is the number of pursuers iterated through in EvaderCell.

Ongoing and future research efforts related to this topic include proving the optimality of the pursuer heading policies, characterizing optimal evader paths to interior solutions, characterizing optimal escort policies, and proving that these solutions are in fact solutions to the differential game. Possible extensions to this work include analyzing the use of Mpursuer1evader in a feedback control policy with a discrete time step, inclusion of sensor noise, inclusion of capture radius, and inclusion of higher order dynamics (e.g. turn constraints). Overall, this work is leading towards the goal of scalable analysis of M vs. N types of scenarios.

Acknowledgments

This paper is based on work performed at the Air Force Research Laboratory (AFRL) Control Science Center of Excellence. Distribution A: Approved for public release. 19 January 2018. Case #88ABW-2018-0230.

References

[1] R. Isaacs, Differential Games: A Mathematical Theory with Applications to Optimization, Control and Warfare. Wiley, New York, 1965.

[2] J. V. Breakwell and P. Hagedorn, "Point capture of two evaders in succession," Journal of Optimization Theory and Applications, vol. 27, no. 1, pp. 89–97, Jan. 1, 1979, issn: 0022-3239, 1573-2878. doi: 10.1007/BF00933327.

[3] S. Y. Liu, Z. Zhou, C. Tomlin, and K. Hedrick, "Evasion as a team against a faster pursuer," in 2013 American Control Conference, Jun. 2013, pp. 5368–5373. doi: 10.1109/ACC.2013.6580676.

[4] S. A. Ganebny, S. S. Kumkov, S. L. Ménec, and V. S. Patsko, "Model Problem in a Line with Two Pursuers and One Evader," Dynamic Games and Applications, vol. 2, no. 2, pp. 228–257, Jun. 1, 2012, issn: 2153-0785, 2153-0793. doi: 10.1007/s13235-012-0041-z.

[5] H. Huang, W. Zhang, J. Ding, D. M. Stipanović, and C. J. Tomlin, "Guaranteed Decentralized Pursuit-Evasion in the Plane with Multiple Pursuers," in Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference On, IEEE, 2011, pp. 4835–4840.

[6] E. Bakolas and P. Tsiotras, "Optimal pursuit of moving targets using dynamic Voronoi diagrams," in 49th IEEE Conference on Decision and Control (CDC), Dec. 2010, pp. 7431–7436. doi: 10.1109/CDC.2010.5717963.

[7] M. G. Earl and R. D'Andrea, "A decomposition approach to multi-vehicle cooperative control," vol. 55, no. 4, pp. 276–291, Apr. 30, 2007, issn: 0921-8890. doi: 10.1016/j.robot.2006.11.002.

[8] D. W. Oyler, P. T. Kabamba, and A. R. Girard, "Pursuit-evasion games in the presence of obstacles," Automatica, vol. 65, pp. 1–11, Supplement C Mar. 1, 2016, issn: 0005-1098. doi: 10.1016/j.automatica.2015.11.018.

[9] A. Festa and R. B. Vinter, "A decomposition technique for pursuit evasion games with many pursuers," in 52nd IEEE Conference on Decision and Control, Dec. 2013, pp. 5797–5802. doi: 10.1109/CDC.2013.6760803.

[10] J. F. Fisac and S. S. Sastry, "The pursuit-evasion-defense differential game in dynamic constrained environments," in 2015 54th IEEE Conference on Decision and Control (CDC), Dec. 2015, pp. 4549–4556. doi: 10.1109/CDC.2015.7402930.

[11] A. Festa and R. B. Vinter, "Decomposition of Differential Games with Multiple Targets," Journal of Optimization Theory and Applications, vol. 169, no. 3, pp. 848–875, Jun. 1, 2016, issn: 0022-3239, 1573-2878. doi: 10.1007/s10957-016-0908-z.

[12] W. Sun, P. Tsiotras, T. Lolla, D. N. Subramani, and P. F. Lermusiaux, "Multiple-Pursuer/One-Evader Pursuit–Evasion Game in Dynamic Flowfields," Journal of guidance, control, and dynamics, 2017.

[13] E. Garcia, Z. E. Fuchs, D. Milutinovic, D. W. Casbeer, and M. Pachter, "A Geometric Approach for the Cooperative Two-Pursuer One-Evader Differential Game," IFAC-PapersOnLine, 20th IFAC World Congress, vol. 50, no. 1, pp. 15 209–15 214, Jul. 1, 2017, issn: 2405-8963. doi: 10.1016/j.ifacol.2017.08.2366.

[14] W. A. Cheung, "Constrained pursuit-evasion problems in the plane," University of British Columbia, 2005.

[15] MOSEK ApS, MOSEK Modeling Cookbook. 2017.

[16] D. Oyler, "Contributions To Pursuit-Evasion Game Theory.," University of Michigan, 2016.

[17] E. Bakolas and P. Tsiotras, "Relay pursuit of a maneuvering target using dynamic Voronoi diagrams," Automatica, vol. 48, no. 9, pp. 2213–2220, Sep. 1, 2012, issn: 0005-1098. doi: 10.1016/j.automatica.2012.06.003.

[18] A. Pierson, Z. Wang, and M. Schwager, "Intercepting Rogue Robots: An Algorithm for Capturing Multiple Evaders With Multiple Pursuers," IEEE Robotics and Automation Letters, vol. 2, no. 2,

pp. 530–537, Apr. 2017, issn: 2377-3766. doi: 10.1109/LRA.2016.2645516.

[19] A. Dobrin, "A review of properties and variations of Voronoi diagrams," Whitman College, 2005.

[20] F. Aurenhammer and H. Edelsbrunner, "An optimal algorithm for constructing the weighted voronoi diagram in the plane," Pattern Recognition, vol. 17, no. 2, pp. 251–257, Jan. 1, 1984, issn: 0031-3203. doi: 10.1016/0031-3203(84)90064-5.

[21] K. Q. Brown, "Geometric Transforms for Fast Geometric Algorithms.," Carnegie-Mellon Univ Pittsburgh PA Dept of Computer Science, CMU-CS-80-101, Dec. 1979.

[22] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, Computational Geometry. Springer, Berlin, Heidelberg, 2008, isbn: 978-3-540-77973-5 978-3-540-77974-2. doi: 10.1007/978-3-540-77974-2_1.

[23] S. Fortune, "A sweepline algorithm for Voronoi diagrams," Algorithmica, vol. 2, no. 1-4, p. 153, Nov. 1, 1987, issn: 0178-4617, 1432-0541. doi: 10.1007/BF01840357.

[24] D. Ghose, "An Introduction to Proportional Navigation," in Guidance of Missiles, Guidance, Control, and Decision Systems Laboratory, Department of Aerospace Engineering, Indian Institute of Science, Bangalore, India, Mar. 2012.