

The Multi-Pursuer Single-Evader Game

A Geometric Approach

Alexander Von Moll · David Casbeer · Eloy Garcia ·
Dejan Milutinović · Meir Pachter

January 2019

Abstract We consider a general pursuit-evasion differential game with three or more pursuers and a single evader, all with simple motion (fixed-speed, infinite turn rate). It is shown that traditional means of differential game analysis is difficult for this scenario. But simple motion and min-max time to capture plus the two-person extension to Pontryagin's maximum principle imply straight-line motion at maximum speed which forms the basis of the solution using a geometric approach. Safe evader paths and policies are defined which guarantee the evader can reach its destination without getting captured by any of the pursuers, provided its destination satisfies some constraints. A linear program is used to characterize the solution and subsequently the saddle-point is computed numerically. We replace the numerical procedure with a more analytical geometric approach based on Voronoi diagrams after observing a pattern in the numerical results. The solutions derived are open-loop optimal, meaning the strategies are a saddle-point equilibrium in the open-loop sense.

Keywords pursuit-evasion · differential game · Voronoi diagram · optimization

Mathematics Subject Classification (2010) 49N70 · 49N90 · 49N75

1 Introduction

In adversarial settings for autonomous vehicles, robots, and systems, pursuit-evasion formulations are an interesting research direction. These formulations started with the seminal contributions by

This paper is based on work performed at the Air Force Research Laboratory (AFRL) *Control Science Center of Excellence*. Distribution Unlimited. 21 August 2018. Case #88ABW-2018-4153.

A. Von Moll, D. Casbeer, E. Garcia
Controls Science Center of Excellence
Air Force Research Laboratory
2210 8th St. WPAFB, OH 45433
E-mail: alexander.von_moll@us.af.mil

D. Milutinović
Department of Computer Engineering
UC Santa Cruz
Santa Cruz, CA 95064
E-mail: dmilutin@ucsc.edu

M. Pachter
Department of Electrical Engineering
Air Force Institute of Technology
WPAFB, OH 45433
E-mail: meir.pachter@us.af.mil

Isaacs [18, 19]. Not long after, Breakwell and Hagedorn [4] investigated a dynamic game seeking the minimum time successive capture of two slower evaders by a faster pursuer. Using [4] as a motivation, the extension to capture multiple evaders by a fast pursuer was addressed [20].

References [15] and [17] looked at the case of multiple pursuers seeking to capture an evader. The concept of a dynamic Voronoi diagram was used to define the closed domain in which evader capture occurred by these several pursuers. Reference [9] finds the intercept set of evaders, assuming that the evaders' goals were known to the pursuers. A related work, where the objective is to rescue certain agents with interference of obstacles was modeled in the Prey, Protector, and Predator game [22]. In this paper, we continue this investigation of capture of a single evader in minimum time by multiple pursuers.

The central contribution of this work is to provide open-loop optimal pursuit and evasion strategies to the M-pursuer one-evader differential game. These strategies, while useful, are not necessarily the solution to the M-pursuer one-evader differential game in the sense of the feedback saddle-point (see [3]). Traditional differential game analysis *a la* Isaacs' method [19] is difficult, as is shown. In order to proceed, we employ the two-person extension to Pontryagin's maximum principle to establish the necessary conditions for optimality pertaining to the pursuers' strategy.

The pursuers' objective is to intercept the evader in minimum time by cooperating as a team. The agents are assumed to have full access to the state of the system, namely, the positions of each agent. We show that, for general initial positions of the agents, cooperation among the pursuers can significantly reduce the capture time of the evader compared to operating in isolation. The state of the system is of high dimension due to the number of agents – some work has been done on decomposition methods to ameliorate this issue [10–12]. However, our approach does not rely on decomposition and therefore considers full cooperation among pursuers. This work extends the solution to the one-on-one and two-on-one games by allowing the evader to stand still when it is advantageous. In these cases, the evader would only worsen its capture time by moving from this point. Due to the proposed pursuer strategy which consists of straight-line paths, the solution lends itself to a geometric interpretation with many interesting properties. Several algorithms are presented for the efficient computation of the evader's region of dominance as well as the optimal capture point under the proposed strategies.

Although the solution presented in this paper is open-loop optimal, the analysis is a useful step in fully solving and verifying the M-pursuer one-evader differential game. The solution of the latter opens up the possibility of analyzing the seemingly intractable (and ambitious) M-pursuer N-evader differential game by breaking the game down into instances of M-pursuer one-evader games and considering combinations of pursuer assignments. The advantage, then, is the removal of the burden of a very high dimensional state space in the differential game analysis.

This paper is a significant enhancement over [25]. Among the enhancements are the introduction of a relative coordinate system, a derivation of the necessary conditions for optimality, proofs of safety for a class of evader behaviors, an alternative linear program formulation, illustrative examples, and additional figures for clarity. Finally, we make it clear that the policies proposed herein are not necessarily the feedback saddle-point equilibrium policies.

The remainder of the paper is organized as follows. Section 2 contains the problem formulation. Section 3 elaborates on the formulation and introduces some solution methods. Section 3.3 presents a new geometric approach, Section 4 defines several efficient algorithms, and Section 5 contains simulation results. Section 6 concludes with some remarks about areas for further research.

2 Problem Formulation

We consider a pursuit-evasion scenario with a single evader, E , and multiple pursuers, P_i , for $i = 1, \dots, M$. We are interested in the case where $M \geq 3$ as the single- and two-pursuer scenarios have been addressed in [16, 19]. The objective of the pursuers is to capture the evader in minimum time, whereas the evader tries to delay capture as long as possible. We consider the case where the pursuers

are faster than the evader. In the case of equal speeds (or a fast evader) one must first determine where in the state space capture is even possible (i.e., solve the game of kind) before solving the capture time problem (the game of degree). Accompanying the speed advantage, the pursuers have an advantage in numbers. There is a marked reduction in capture time in the two-pursuer scenario compared to having just one pursuer [16]. The intent of this work is to show even greater improvement when $M > 2$.

The state of the system is determined solely by the positions of each of the agents on the Euclidean plane in two dimensions

$$\begin{aligned} E &= (x_E, y_E), \\ P_i &= (x_{P_i}, y_{P_i}), \quad \forall i \in \{1, \dots, M\}, \end{aligned}$$

so the state space's dimension is $2(M + 1)$. All the agents have simple motion, meaning their control input at every time instant is their heading angle. For the evader, we include speed as a control input bounded by $V_{E_{max}}$; the pursuers all share a constant velocity, $V_P > V_{E_{max}}$. Thus the kinematics of the system can be written as

$$\begin{aligned} \dot{x}_E &= u_E, \\ \dot{y}_E &= v_E, \\ \dot{x}_{P_i} &= u_i \quad i = 1, \dots, M, \\ \dot{y}_{P_i} &= v_i \quad i = 1, \dots, M, \end{aligned} \tag{1}$$

$$\begin{aligned} s.t. \quad &u_E^2 + v_E^2 \leq V_{E_{max}}^2, \\ &u_i^2 + v_i^2 = V_P^2 \quad i = 1, \dots, M, \end{aligned}$$

where u_E and v_E are the evader's velocity components in the x and y direction, and, similarly, u_i and v_i for the i th pursuer.

Remark 1 Although it appears that the pursuers have two control variables they only have one since the choice of either u_i or v_i completely determines the other by the final constraint in (1).

In the realistic plane, there are $2M + 2$ states, 2 for each agent. However, it is possible to reduce the number of states to $2M$ by considering a relative coordinate system. In this relative coordinate system, the evader's position is always $(0, 0)$ and thus the $2M$ states correspond to the x and y coordinate of each pursuer relative to the evader. The transformation is given by

$$\begin{aligned} x_i &= x_{P_i} - x_E \quad i = 1, \dots, M, \\ y_i &= y_{P_i} - y_E \quad i = 1, \dots, M, \end{aligned} \tag{2}$$

where (x_i, y_i) are the coordinates of the i th pursuer relative to the evader. Substituting (1) into (2) yields the following expressions for the kinematics in the reduced state space

$$\begin{aligned} \dot{x}_i &= u_i - u_E \quad i = 1, \dots, M, \\ \dot{y}_i &= v_i - v_E \quad i = 1, \dots, M, \end{aligned} \tag{3}$$

$$\begin{aligned} s.t. \quad &u_E^2 + v_E^2 \leq V_{E_{max}}^2, \\ &u_i^2 + v_i^2 = V_P^2 \quad i = 1, \dots, M. \end{aligned}$$

Figure 1 shows the two coordinate systems and how they are related.

The agents have at their disposal the current state of the system (i.e., full information) but not the current control action of the other agents. One may be tempted to consider a scenario wherein the agents also have access to the history of the system's evolution (including past control actions of all the agents). In this alternate formulation, agents may form some belief about their opponent's next action that is conditioned on the state history and previous control actions. Isaacs discusses this

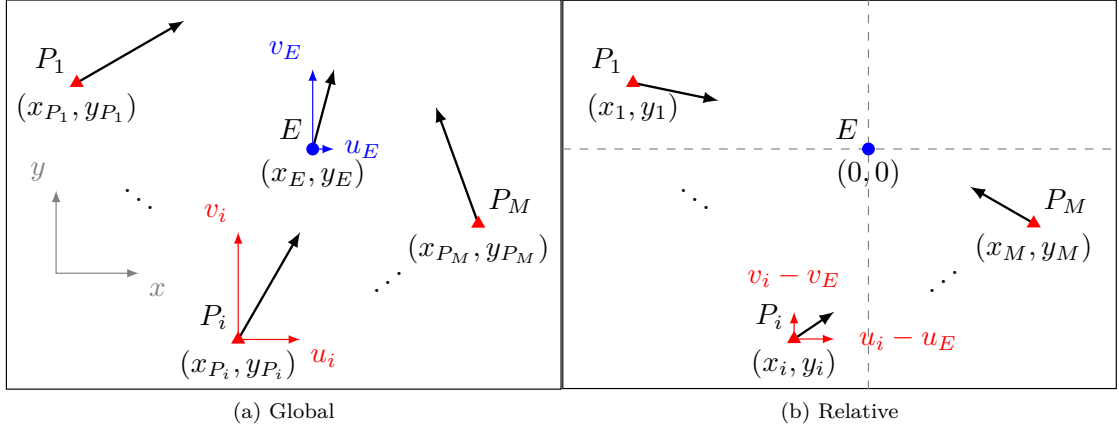


Fig. 1 Coordinate systems

possibility and reasons that, because the agents' control input can change abruptly without notice, it is impossible to rely on any prediction of the opponent's future position [19]. If an agent truly did adhere to some behavior beyond what is specified in the kinematics, then perhaps its opponent *could* improve its performance if it could correctly ascertain the behavior. What is gained in performance is lost in robustness, as is generally the case in the real world.

Let the state of the system be represented by $\mathbf{x} \triangleq ((x_1, y_1), \dots, (x_M, y_M))$. Similarly, let $\mathbf{u}_E \triangleq (u_E, v_E)$ and $\mathbf{u}_P \triangleq (u_1, v_1, \dots, u_M, v_M)$ represent the control inputs of the evader and pursuers, respectively. Note that each part of the state is a function of time, though we will not notate it explicitly. The set of terminal states for the scenario is defined by the requirement of point capture:

$$\Lambda = \{\mathbf{x} \mid \exists i, 1 \leq i \leq M \text{ s.t. } (x_i, y_i) = (0, 0)\}. \quad (4)$$

Remark 2 Eq. (4) allows one or more pursuers to reach the position $(0, 0)$ at time T .

Another way of denoting the set of the terminal conditions is by defining

$$\Psi(\mathbf{x}) = \prod_{i=1}^M (x_i^2 + y_i^2), \quad (5)$$

and setting

$$\Lambda = \{\mathbf{x} \mid \Psi(\mathbf{x}) = 0\}. \quad (6)$$

The condition $\Psi(\mathbf{x}) = 0$ is akin to a terminal manifold of dimension $2M - 1$.

The terminal time T , or capture time, is the first time such that the system state enters Λ ,

$$T = \inf\{t \mid \mathbf{x}(t) \in \Lambda\}. \quad (7)$$

Let the set of interceptors be those pursuers whose positions are $(0, 0)$ at time T , i.e.,

$$\mathcal{I} = \{i \mid (x_i(T), y_i(T)) = (0, 0)\}. \quad (8)$$

The terminal time T is also the cost (or payoff) of the game:

$$J(\mathbf{u}_E(\mathbf{x}), \mathbf{u}_P(\mathbf{x})) = \int_0^T d\tau, \quad (9)$$

with $\mathbf{u}_E, \mathbf{u}_P$ subject to (3). The value function describes the minimax value of the cost function starting from the point $\mathbf{x}_0 = \mathbf{x}(0)$ at time $t = 0$ [3]

$$V(\mathbf{x}_0) = \min_{\mathbf{u}_P} \max_{\mathbf{u}_E} \int_0^T d\tau. \quad (10)$$

For such a function to exist the min and max must be interchangeable. The existence of this function or even of a saddle-point in the cost function is not guaranteed, however we will proceed under the assumption that at least the saddle-point exists. That is, we seek the control inputs that satisfy

$$J(\mathbf{u}_E, \mathbf{u}_P^*) \leq J(\mathbf{u}_E^*, \mathbf{u}_P^*) \leq J(\mathbf{u}_E^*, \mathbf{u}_P). \quad (11)$$

The control policies in (9) are state feedback policies, and thus $J(\mathbf{u}_E^*, \mathbf{u}_P^*)$ represents a feedback saddle-point equilibrium. The Isaacs Equation [19] can be written as

$$\min_{\mathbf{u}_P} \max_{\mathbf{u}_E} \sum_{i=1}^M V_{x_i}(u_i - u_E) + V_{y_i}(v_i - v_E) + 1 = 0, \quad (12)$$

where

$$V_{x_i} = \frac{\partial V}{\partial x_i},$$

$$V_{y_i} = \frac{\partial V}{\partial y_i}.$$

For many problems in [19], the Isaacs Equation (12), along with information about the terminal surface, is sufficient to generate game-optimal trajectories. Indeed, this is also the approach taken in [14]. If we attempt this here, the curse of dimensionality bites us and an impasse is reached.

To circumvent the need to deal directly with the value function itself, we turn our attention to an analysis based upon open-loop strategies. As in many of the examples in [3] we drop the explicit dependence of the control policies on the state

$$\mathbf{u}_P(t, \mathbf{x}) = \hat{\mathbf{u}}_P(t), \quad \mathbf{u}_E(t, \mathbf{x}) = \hat{\mathbf{u}}_E(t). \quad (13)$$

Here, $\hat{\mathbf{u}}_P(t)$ and $\hat{\mathbf{u}}_E(t)$ represent open-loop controls for the pursuers and evader, respectively.

2.1 Necessary Conditions for Optimality

Under the assumptions that the pair $(\mathbf{u}_P^*, \mathbf{u}_E^*)$ provides a saddle-point solution in feedback strategies and the corresponding open-loop representation $(\hat{\mathbf{u}}_P^*, \hat{\mathbf{u}}_E^*)$ provides a saddle-point solution in open-loop policies then Theorem 2 of [3] provides a framework for deriving the necessary conditions for optimality. The procedure is based upon the two-person extension to Pontryagin's maximum principle. First let the Hamiltonian be given by

$$H(t, \lambda(t), \hat{\mathbf{u}}_P, \hat{\mathbf{u}}_E) = 1 + \lambda^\top \begin{bmatrix} u_1 - u_E \\ v_1 - v_E \\ u_2 - u_E \\ v_2 - v_E \\ \vdots \\ u_M - u_E \\ v_M - v_E \end{bmatrix}, \quad (14)$$

where λ is the costate vector in \mathbb{R}^{2M}

$$\lambda = (\lambda_{x_1}, \lambda_{y_1}, \dots, \lambda_{x_M}, \lambda_{y_M}).$$

Thus the Hamiltonian becomes

$$H = 1 + \sum_{i=1}^M \lambda_{x_i} (u_i - u_E) + \lambda_{y_i} (v_i - v_E), \quad (15)$$

where the elements u_i and v_i , $i = 1, \dots, M$, and u_E and v_E are taken to mean the corresponding elements in the open-loop policies $\hat{\mathbf{u}}_P$ and $\hat{\mathbf{u}}_E$, respectively. The costate variables satisfy

$$\dot{\lambda}_{x_i} = -\frac{\partial H}{\partial x_i} = 0, \quad \dot{\lambda}_{y_i} = -\frac{\partial H}{\partial y_i} = 0, \quad \text{for } i = 1, \dots, M. \quad (16)$$

Eq. (14) does not depend explicitly on the state \mathbf{x} . This is a result of the fact that the kinematics are only a function of the control inputs. Thus, we have $\dot{\lambda} = 0$ which implies the costate variables are constant w.r.t. time (e.g., $\lambda_{x_1}(t) = \lambda_{x_1} = \text{const}$). The minimizing controls for the pursuers can easily be obtained from (3) and (15) as

$$u_i^* = -\frac{\lambda_{x_i} V_P}{\sqrt{\lambda_{x_i}^2 + \lambda_{y_i}^2}}, \quad v_i^* = -\frac{\lambda_{y_i} V_P}{\sqrt{\lambda_{x_i}^2 + \lambda_{y_i}^2}}, \quad (17)$$

which are also constant. Because the optimal control policy for the pursuers is constant, their state trajectories are straight lines in the global coordinate system. The only caveat is that the optimal control policy is defined only if $\lambda_{x_i} \neq 0$ or $\lambda_{y_i} \neq 0$. When $\lambda_{x_i} = \lambda_{y_i} = 0$ the implication is that the value of the game is not sensitive to pursuer i . When the i th pursuer participates in the capture of the evader (i.e., $x_i(T) = y_i(T) = 0$), either one of λ_{x_i} or λ_{y_i} must be nonzero.

3 Solution

We now return to the original coordinate system, and without loss of generality consider the starting position of the evader to be the origin and the evader's speed to be unity. Let the ratio of evader's max speed to pursuers' speed be given by $\alpha = V_{E_{max}}/V_P < 1$. The necessary conditions for optimality derived above help to justify the following:

Proposition 1 *The optimal trajectories of the agents are straight line paths.*

Moreover, under optimal play, the pursuers' heading should not change; that is, their trajectory is comprised of a single straight line segment. Thus, we repose the differential game problem of min-max capture time as finding the coordinates $I = (x_I, y_I)$ that maximize the evader's life assuming all the pursuers head directly to that point starting at $t = 0$. Then the terminal time (which is also the cost/payoff of the game in the preceding section) can be written simply as

$$\bar{T}(\mathbf{x}, I) = \min_i \alpha \sqrt{(x_{P_i} - x_I)^2 + (y_{P_i} - y_I)^2}, \quad (18)$$

which is the smallest time for any of the pursuers to reach the designated intercept point. There is a subtle shift here from the traditional game-theoretic framework wherein the agents have absolutely no knowledge of the opponents' present or future control action. Here, the understanding is that the point I represents a designated intercept point that both the evader and pursuers have knowledge of. Thus (18) is the capture time assuming that all the agents head directly to the designated (or 'agreed upon') point. Suppose the evader were to choose the point I , then (assuming I could be reached safely by the evader) (18) gives the worst-case capture time from the perspective of the evader. Conversely,

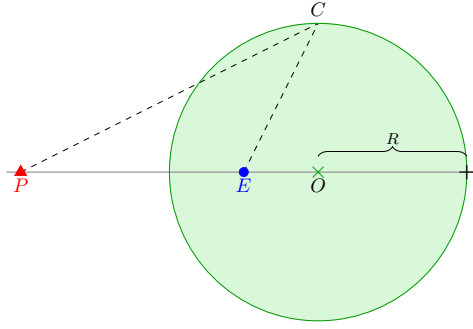


Fig. 2 Apollonius circle for a single pursuer.

if the pursuers were to select point I , it is in the evader's best interest to flee from this point rather than aim towards it. Therefore, we treat the point I as if it is the choice of the evader:

$$\begin{aligned} T(\mathbf{x}) &= \max_{(x_I, y_I)} \bar{T}(\mathbf{x}, I) \\ &= \max_{(x_I, y_I)} \min_i \alpha \sqrt{(x_{P_i} - x_I)^2 + (y_{P_i} - y_I)^2}, \end{aligned} \quad (19)$$

wherein the evader is selecting the best choice among all the worst-case capture times, \bar{T} . Eq. (19) is *analogous* to the value of the game in (10), but not necessarily equivalent. As hinted previously, the order of max-min in (19) is not interchangeable. In fact, interchangeability of max and min is *required* to obtain a saddle point solution to a differential game (known as Isaacs Condition) [19].

3.1 Constraints

Eq. (19) is not particularly useful because the evader could designate an intercept point at infinity and thus the capture time would also be infinite. In other words, (19) is unconstrained. We should amend this by stating (1) the evader must be able to reach point I in T time and (2) the evader must be able to reach point I *safely*, that is, capture of the evader *en route* to I ought not be possible. Constraint (1) is easily formulated as [25]

$$x_I^2 + y_I^2 \leq T^2, \quad (20)$$

since the evader's speed and starting position are one and $(0, 0)$, respectively. The need for (2) is due to the fact that if capture is possible prior to the evader reaching I , then the associated terminal time T is meaningless in the sense that it is no longer analogous to a game-theoretic solution. To formalize (2), the concept of the Apollonius circle is useful. In the context of pursuit and evasion, the Apollonius circle is the locus of points such that the ratio of distances to the evader and pursuer is equal to their speed ratio [19]. Thus, the Apollonius circle defines the points that can be reached simultaneously by the pursuer and evader assuming they each head directly to the point at their maximum speeds. Inside the circle (i.e., the Apollonius *disk*) are points in which the evader can reach before the pursuer under the same assumptions. When the speed ratio is unity, the Apollonius circle becomes the orthogonal bisector of the segment \overline{EP} . Figure 2 displays the Apollonius circle. Note, the optimal capture point for this configuration is marked with a $+$. The center of the Apollonius circle, O , marked with a green \times , lies on the line passing through the evader and pursuer. The definition given above stipulates that, for any point C on the circle, the ratio of $\overline{EC}/\overline{PC} = \alpha$. This relation allows one to express the location

of the circle's center as well as its radius

$$\overline{EO} = \frac{\alpha^2}{1 - \alpha^2} \overline{PE}, \quad (21)$$

$$R = \frac{\alpha}{1 - \alpha^2} \overline{PE}. \quad (22)$$

Putting the circle center associated with pursuer i into the global coordinate system gives

$$x_{O_i} = -\frac{\alpha^2}{1 - \alpha^2} x_{P_i}, \quad (23)$$

$$y_{O_i} = -\frac{\alpha^2}{1 - \alpha^2} y_{P_i}, \quad (24)$$

$$R_i = \frac{\alpha}{1 - \alpha^2} \sqrt{x_{P_i}^2 + y_{P_i}^2}. \quad (25)$$

The set describing the Apollonius disk associated with pursuer i is

$$\mathcal{D}_i = \left\{ (x, y) \mid (x - x_{O_i})^2 + (y - y_{O_i})^2 \leq R_i^2 \right\}. \quad (26)$$

For a point to be safely reachable by the evader, that is, no pursuer can reach the point or any point along the evader's path before the evader, the point and the evader's path to the point must lie inside the Apollonius disk associated with each pursuer

$$\begin{aligned} I &\in \bigcap_{i=1}^M \mathcal{D}_i, \\ (x_I - x_{O_i})^2 + (y_I - y_{O_i})^2 &\leq R_i, \quad i = 1, \dots, M. \end{aligned} \quad (27)$$

Let the intersection of Apollonius disks be represented by $\mathcal{S} = \bigcap_{i=1}^M \mathcal{D}_i$; we refer to \mathcal{S} as the evader's *safety region*. Eq. (27) is only a necessary condition, however, it does not guarantee safety. First of all, it only stipulates that the point I is inside every Apollonius disk. It is possible for the evader to take a path which may leave the set \mathcal{S} , in which case there exists a valid pursuer path which leads to capture at the boundary, $\partial\mathcal{S}$. Moreover, one must recompute \mathcal{S} at each instant of time as the game is played; thus under optimal play, \mathcal{S} shrinks over time as the pursuers approach the evader. To guarantee safety, then, the point I must lie inside the instantaneous region $\mathcal{S}(t)$ for all $t \leq T$; at T the region \mathcal{S} collapses to a point.

Property 1 A safe evader path is one in which, given the initial positions of all the agents, \mathbf{x}_0 , and their velocities, V_E and V_P , there does not exist a pursuer path that captures the evader *en route* to its destination.

Lemma 1 \mathcal{S} is a convex set.

Proof \mathcal{S} is the intersection of Apollonius disks, and each Apollonius disk is a convex set, thus for any two points $p_1, p_2 \in \mathcal{S}$,

$$p_1, p_2 \in \mathcal{D}_i, \quad i = 1, \dots, M,$$

from the definition of set intersection and,

$$\mu p_1 + (1 - \mu)p_2 \in \mathcal{D}_i, \quad i = 1, \dots, M, \quad 0 \leq \mu \leq 1,$$

by convexity of the \mathcal{D}_i . Therefore, by set intersection, all of these points are also in \mathcal{S} , implying convexity [7]. \square

Corollary 1 Any straight-line path starting inside and ending inside \mathcal{S} lies entirely in \mathcal{S} .

Theorem 1 *Given an evader and pursuers at P_i , $i = 1, \dots, M$ and point $I \in \mathcal{S}$, if the evader travels directly towards I at maximum speed then the evader will reach I safely. There does not exist a pursuer path that can intercept the evader before \bar{T} according to (18).*

Proof By construction, the evader's initial position of $(0,0)$ lies inside \mathcal{S} . Every point along the straight-line path from $(0,0)$ to I is inside \mathcal{S} from Corollary 1. The evader can reach I under the prescribed evader policy at or before the time that any pursuer can reach it by construction of the Apollonius disks \mathcal{D}_i and the fact that $I \in \mathcal{S}$. This implies that the point I lies inside $\mathcal{S}(t)$ for $t \leq t_1$ where

$$t_1 = \sqrt{x_I^2 + y_I^2} \leq T,$$

is the time at which the evader reaches I . The last inequality is enforced by (20). For $t_1 \leq t \leq T$ the evader's position $E = I$, thus $I \in \mathcal{S}$ for $t_1 \leq t \leq T$ which completes the proof. \square

Theorem 1 specifies an evader policy which is guaranteed to be safe in the sense that (18) gives a worst-case capture time. However, this policy is not unique and there may be many alternative policies which are also safe [24, 25].

Proposition 2 *Given an evader at E and pursuers at P_i , $i = 1, \dots, M$ and point $I \in \mathcal{S}$, if the evader travels directly towards I at speed s then the evader will reach I safely, where*

$$s \triangleq \frac{1}{\bar{T}} \sqrt{x_I^2 + y_I^2}. \quad (28)$$

That is, there does not exist a pursuer path that can intercept the evader before \bar{T} according to (18).

Another policy, which is analogous to the policy in Theorem 1 is for the evader to head to I at maximum speed and then switch heading between 0 and π infinitely often while remaining at maximum speed [25]. In fact, allowing the evader to modulate its heading continuously alleviates entirely the need to consider its speed as a control variable. Oddly enough, a straight-line path is not strictly necessary to guarantee the evader can reach I safely (nor is it sufficient). An example scenario is included in Section 5 wherein the evader takes a safe, non-straight-line path to I corresponding to the arg max in (19). For the remainder, we restrict the evader to employ the policy prescribed in Theorem 1 since it is safe and also consistent with Proposition 1.

3.2 Linear Program with Quadratic Constraints

The multiple pursuer single evader game has been re-posed as solving (19) subject to the constraints (20) and (27) with the understanding that all the agents head directly to I corresponding to the arg max of (19). In order to solve this new problem, we formulate it as a linear program:

$$\max_{\mathbf{z}} \mathbf{c}^\top \begin{bmatrix} \mathbf{z} \\ \mathbf{s} \end{bmatrix}, \quad (29)$$

with slack variables $\mathbf{s} = [s_1 \ s_2 \ \dots \ s_M]^\top$ subject to the constraints

$$g_i(\mathbf{z}) = 0, \quad i = 1, \dots, M, \quad (30)$$

$$-g_E(\mathbf{z}) \leq 0, \quad (31)$$

$$-s_i \leq 0, \quad i = 1, \dots, M, \quad (32)$$

where $\mathbf{c}^\top = [1 \ 0 \ \dots \ 0]_{1 \times (M+3)}$, $\mathbf{z}^\top = [m \ x \ y]$. The functions g are defined as

$$g_i(\mathbf{z}) = \frac{m}{\alpha^2} - (x - x_{P_i})^2 - (y - y_{P_i})^2 + s_i, \quad i = 1, \dots, M, \quad (33)$$

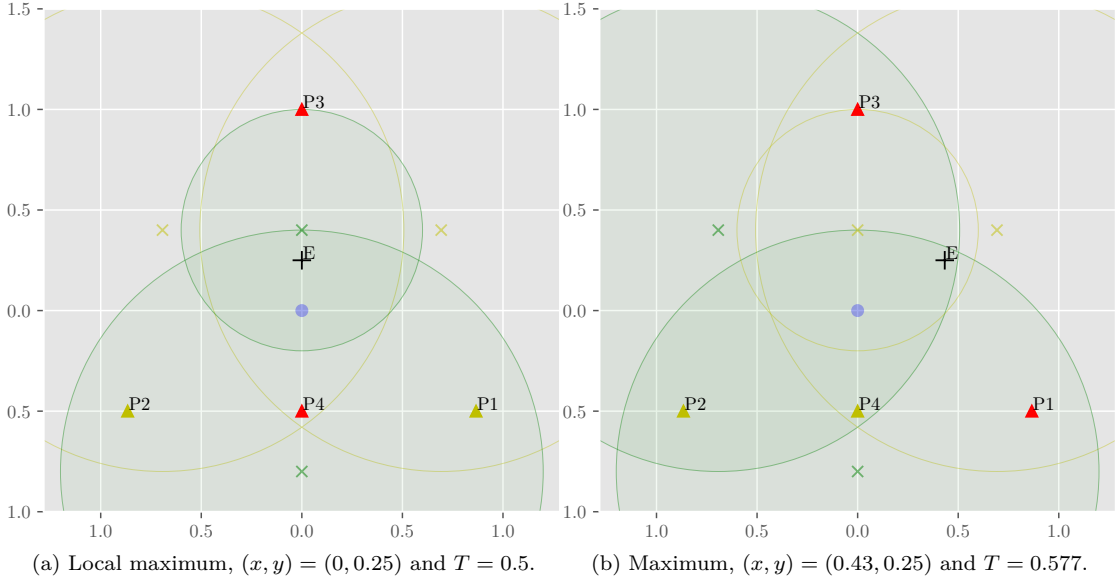


Fig. 3 Numerical results for Example 1

and

$$g_E(\mathbf{z}) = m - x^2 - y^2. \quad (34)$$

Note that the problem is about the maximization of a linear function with quadratic constraints. Furthermore, the constraints (30) are active whether or not the point (x, y) is on $\partial\mathcal{S}$ or $\mathcal{S} \cap \overline{\partial\mathcal{S}}$ (i.e., on the boundary or interior of \mathcal{S}). The set of pursuers which intercept the evader at (x, y) is given as,

$$\mathcal{I} = \{i \mid s_i = 0, i = 1, \dots, M\} \quad (35)$$

hence, s_i represents the remaining time required for pursuer i to reach (x, y) after the evader and interceptors have reached it. Thus, (33) with (32) is analogous to the Apollonius circle constraint introduced in (27). In other words, $s_i > 0$ holds for any point inside pursuer i 's Apollonius circle. Similarly, (34) and (31) correspond to the reachability constraint (20). This constraint is always satisfied, however, by the definition of the Apollonius circle and the fact that (30) and (32) constrain the point to be inside \mathcal{S} .

This linear program is now amenable to solution using a generic numerical optimization scheme. Consider the following example, whose numerical results are shown in Figure 3.

Example 1

$$\begin{aligned} M &= 4 \\ P_1 &= (\cos(-\pi/6), \sin(-\pi/6)) \\ P_2 &= (\cos(7\pi/6), \sin(7\pi/6)) \\ P_3 &= (0, 1) \\ P_4 &= (0, -0.5) \\ V_P &= 1.5 \end{aligned}$$

In Figure 3 and those to follow the pursuers are indicated with triangles and colored according to whether or not they are in the set \mathcal{I} (red if so, yellow if not). The initial position of the evader is

marked by a blue circle and the capture point is marked with $+$ and E . The Apollonius circles for those pursuers in \mathcal{I} are shown as translucent green and for those not in \mathcal{I} just the border is marked with yellow. Finally, the circle centers are marked with \times . As previously mentioned, the set I is computed from (35). When $s_i = 0$, the constraints g_E and g_i define the Apollonius circle associated with pursuer i . Thus adding the Apollonius circle constraints, (27), will not change the optimization problem. In fact, once we introduce the Apollonius circle constraints, we can remove the slack variables from our problem formulation and use $g_i(\mathbf{z}) \leq 0$.

For the scenario in Example 1 there is a local maxima (shown in Figure 3a) and two global maxima which have equal value. One of the global maxima is shown in Figure 3b, and the other is symmetric about the y axis. The matter of local maxima and the possibility for multiple global maxima presents a practical issue. In general, it is not known how many maxima or local maxima may be present in (29). One may need to initialize the numerical optimization procedure many times to uncover all the different maxima, and even then an upper bound on this number is not known at this time. The two initial conditions used for Example 1 are $\mathbf{z} = [0.1, 0.1, 1]^\top$ and $[0, 0.1, 1]^\top$, but these may not be sufficient to find the true global maximum. In summary, the numerical optimization process is blind to any special structure of the problem and sensitive to the presence of local maxima.

3.3 Geometric Approach

The linear program introduced in the previous section must search over a continuous space; it was mentioned that the linear program does not make use of any special structure to search the space intelligently. Consider the following as a motivating example for finding some additional useful information embedded in the problem:

Example 2

$$\begin{aligned} M &= 4 \\ P_1 &= (\cos(-\pi/6), \sin(-\pi/6)) \\ P_2 &= (\cos(7\pi/6), \sin(7\pi/6)) \\ P_3 &= (0, 1) \\ V_P &= 1.5 \end{aligned}$$

Note, this example is the same as Example 1 but with P_4 removed.

Now, we repeat the numerical optimization of the linear program as before, but with initial $z = [0, 0, 1]^\top$ and $[0.502, 0.29, 1]^\top$. Figure 4 shows the results of the numerical optimization. Like Example 1 there are a number of local maxima and global maxima. In this case, there are three local maxima, one of which is shown in Figure 4a. The other two local maxima are radially symmetric w.r.t the origin. Now, the global maximum is unique and lies in $\mathcal{S} \cap \partial\mathcal{S}$, that is, in the interior of the region \mathcal{S} whereas before the global maximum was near (essentially on) the border $\partial\mathcal{S}$. The former case is easy to understand – it corresponds to the furthest distance from the evader to a point on $\partial\mathcal{S}$. In the latter case it becomes optimal for the evader to not move at all.

Remark 3 Simultaneous capture by two pursuers, an example of which is shown in Figure 4a, occurs at the intersection of Apollonius circles associated with the intercepting pursuers.

Figure 4b, on the other hand, depicts simultaneous capture by three pursuers.

When Isaacs first posed the two cutters and fugitive ship problem [19] he posited that the optimal strategy would be for all the agents to head to the further of the two Apollonius circle intersection points. Indeed, this was proven to be true in certain regions of the state space [16]. Of course, the game may also degenerate to capture by a single pursuer, which also holds true for the multiple pursuer

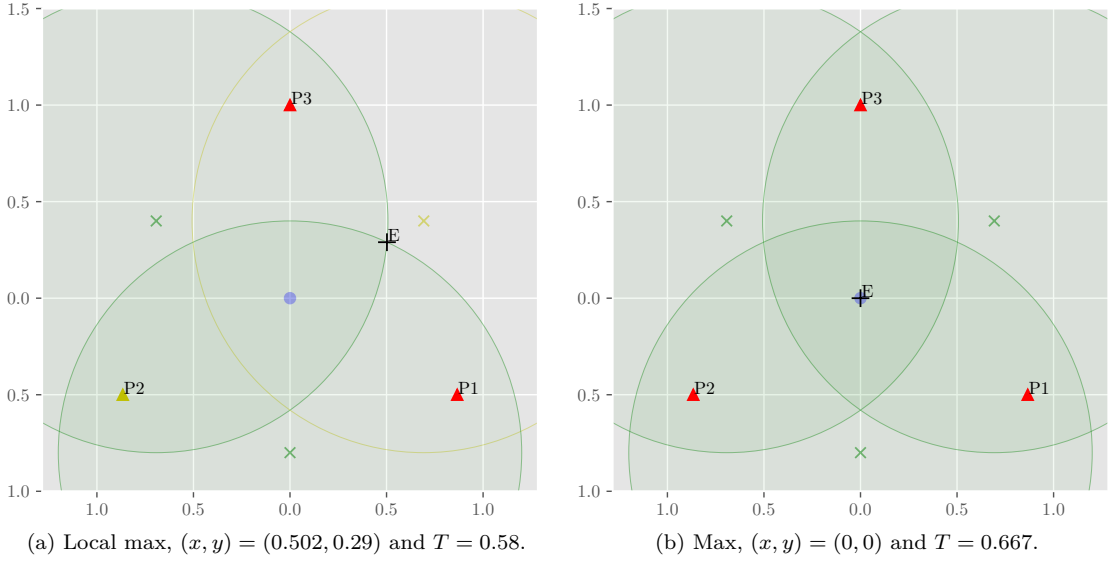


Fig. 4 Numerical results for Example 2 highlighting some special properties of the solution.

case as we shall explore later on. The geometry and solution of the two pursuer game is driven by the fact that optimality dictates straight-line paths. From Proposition 1 and its preceding analysis, we place ourselves in this same realm of geometric analysis in which the Apollonius circles are of chief importance. However, from Figure 4b it is plain that the optimal intercept point $(0, 0)$ has little to do with the Apollonius circles themselves. Up to now we have only considered dividing the state space according to regions dominated by the evader versus a particular pursuer. Now consider the region dominated by a particular pursuer versus all the other pursuers; that is, the region of the state space where pursuer i can reach before any other pursuer $j \neq i$. This partitioning of the state space (which, at the moment, leaves out any consideration of the evader) is precisely a Voronoi diagram [8]. Because the pursuers share the same velocity, the diagram is comprised of straight line segments which partition the entire xy plane. Figure 5 shows the same setup but with the pursuer Voronoi diagram overlaid.

Remark 4 The Apollonius circle intersections lie on the edges of the pursuer Voronoi diagram. Thus, for simultaneous capture by two pursuers to be optimal it must occur on an edge of the pursuer Voronoi diagram.

Remark 5 The optimal capture point for Example 4b which represents simultaneous capture by three pursuers coincides with the vertex of the pursuer Voronoi diagram.

This last observation, in particular, drives the remainder of the analysis and allows us to generalize the multiple pursuer single evader game to any number of pursuers. Also, note that the application of Voronoi diagrams to the analysis of pursuit-evasion games is not novel and has been explored extensively in the literature (c.f. [2, 6, 17, 21, 23]). 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. This construct is particularly useful in the present context as we have established that the optimal trajectories ought to be comprised of constant-heading paths. We define two different Voronoi diagrams, each parameterized as a set of vertices, a set of edges, and a set of agent positions (which

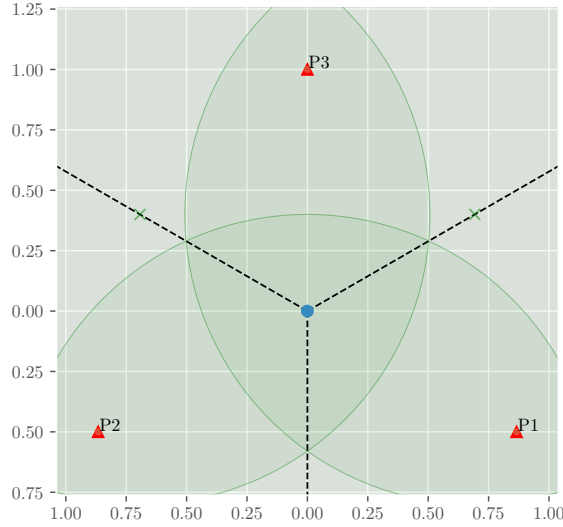


Fig. 5 Example 2 with the pursuer Voronoi diagram overlaid. The vertex is colored blue and the cell borders are shown as dashed lines.

correspond to so-called generator points) [25],

$$\begin{aligned}\mathbb{V}_E &= (\mathcal{V}_E, \mathcal{E}_E, \{E, P_1, \dots, P_M\}), \\ \mathbb{V}_P &= (\mathcal{V}_P, \mathcal{E}_P, \{P_1, \dots, P_M\}).\end{aligned}$$

Note, the only difference between the two is that \mathbb{V}_E includes the evader as a generator point, and \mathbb{V}_P use only the pursuer positions as generator points. The latter describes a partitioning wherein the pursuers can reach points in their own cell before any other pursuer. Because the pursuers share the same velocity the edges \mathcal{E}_P are segments of the perpendicular bisectors between neighboring pursuers. The same is not true, however, for the edges of the evader's cell in \mathbb{V}_E , since E is slower than the neighboring pursuers. Typically, when different weights (velocities) are involved one may consider \mathbb{V}_E to be a multiplicatively-weighted Voronoi diagram [8]. Here, we note that the evader's cell in \mathbb{V}_E is exactly the safety region, \mathcal{S} , which we have already defined using the Apollonius circles. Now let \mathcal{S} be parameterized as

$$\mathcal{S} = (\mathcal{V}_S, \mathcal{E}_S), \quad (36)$$

where \mathcal{V}_S is an ordered set of vertices of \mathcal{S} , and \mathcal{E}_S is an ordered set of arcs [25]. The ordering of these two sets is such that the i th edge in \mathcal{E}_S connects the $i - 1$ th and i th Apollonius circle intersection in \mathcal{V}_S .

3.4 Types of Solutions

In the introduction we noted that for the case of $M > 2$ there is a new type of solution over the previously derived solutions to the one-on-one and two-on-one scenarios. First, let us briefly recount these known solutions as they are still solutions to the M -on-one scenario for particular configurations. As example, consider a case with $M > 2$ wherein one pursuer is very close to the evader and the other $M - 1$ pursuers are very (let us say “infinitely”) far away. Then the solution, obviously, degenerates to the solution of the one-on-one scenario between the close pursuer and the evader. The solution, in this case, is given by pure pursuit: the evader's and pursuer's heading should be along the line-of-sight [19]. Capture occurs on the point on the Apollonius circle which is antipodal to the pursuer.

Hugo Steinhaus and Rufus Isaacs each proposed the case of two pursuers against one evader. Isaacs referred to this scenario as the “two cutters and fugitive ship problem” [19]. In his book, Isaacs posited that the solution of the game was for all three agents to head to the intersection of Apollonius circles furthest from the evader. Note, however, that Isaacs did not mention the cases in which the two-on-one scenario degenerates to one-on-one. Nonetheless, using the geometric intuition of Isaacs, the Value function and saddle-point strategies were derived only recently in [16].

As will be shown in the following sections, it is not necessary for some pursuers to be “infinitely far” away from the evader for the scenario to degenerate to two-on-one or one-on-one. Instead, we retain the solutions to all of these sub-scenarios as candidate solutions. Note that each of the candidates described above necessarily occur on the boundary of the evader’s dominance region $\partial\mathcal{S}$. When $M \geq 3$ we must consider new candidate solution, one in which capture occurs in the interior of \mathcal{S} , as opposed on the boundary $\partial\mathcal{S}$. We claim that when the solution to (19) occurs in the interior of \mathcal{S} , the point necessarily corresponds to a vertex of \mathbb{V}_P , the Voronoi diagram for the pursuers [25]

$$(x^*, y^*) \notin \partial\mathcal{S} \implies (x^*, y^*) \in \mathcal{V}_P, \quad (37)$$

and thus implies simultaneous capture by three or more pursuers. Now, with all possible candidate solutions described, we include one of the main results in [25] which states that the solution of (19) is among this finite set of candidates comprised of points representing capture by a single pursuer (\mathcal{P}_S^1), capture by two pursuers simultaneously (\mathcal{V}_S), and capture by three or more pursuers simultaneously (\mathcal{V}_{P_S}).

Theorem 2

$$(x^*, y^*) \in \mathcal{P}_S^1 \cup \mathcal{V}_S \cup \mathcal{V}_{P_S}, \quad (38)$$

where

$$\begin{aligned} (x^*, y^*) &= \arg \max_{(x,y) \in \mathcal{S}} \min_{i \in \{1, \dots, M\}} \|(x, y) - P_i\|_{\alpha_P}, \\ \mathcal{P}_S^1 &= \left\{ x_i, y_i \left| x_i, y_i = R_i(1 + \alpha_P) \frac{E - P_i}{\|E - P_i\|}, \right. \right. \\ &\quad \left. \left. x_i, y_i \in \mathcal{S}, i = 1, \dots, M \right\}, \\ \mathcal{V}_{P_S} &= \mathcal{V}_P \cap \mathcal{S}. \end{aligned} \quad (39)$$

Proof The interested reader is referred to [25]. \square

3.5 Categories of Pursuers

One of the interesting consequences of Theorem 2 is the fact that not all the pursuers have an effect on the ploy of the game. For example, a pursuer that is very far away from the evader compared to the other pursuers may not be able to reach the evader before capture occurs. This is certainly the case when the Apollonius circle of a pursuer completely contains the evader’s safety region, \mathcal{S} . As mentioned in [25] the set of pursuers can be broken up into four disjoint sets. The membership of the pursuers is a function of the optimal capture point I (i.e., the solution to the first equation in (39)). The first set contains those pursuers who reach I at \bar{T} , according to (18). Earlier, we referred to this set as “interceptors” using the notation \mathcal{I} . The set of pursuers whose Apollonius circle completely contains the safety region \mathcal{S} is given by

$$\mathcal{I}^- = \{i \mid \partial\mathcal{S} \cap \partial\mathcal{D}_i = \emptyset, i = 1, \dots, M\}. \quad (40)$$

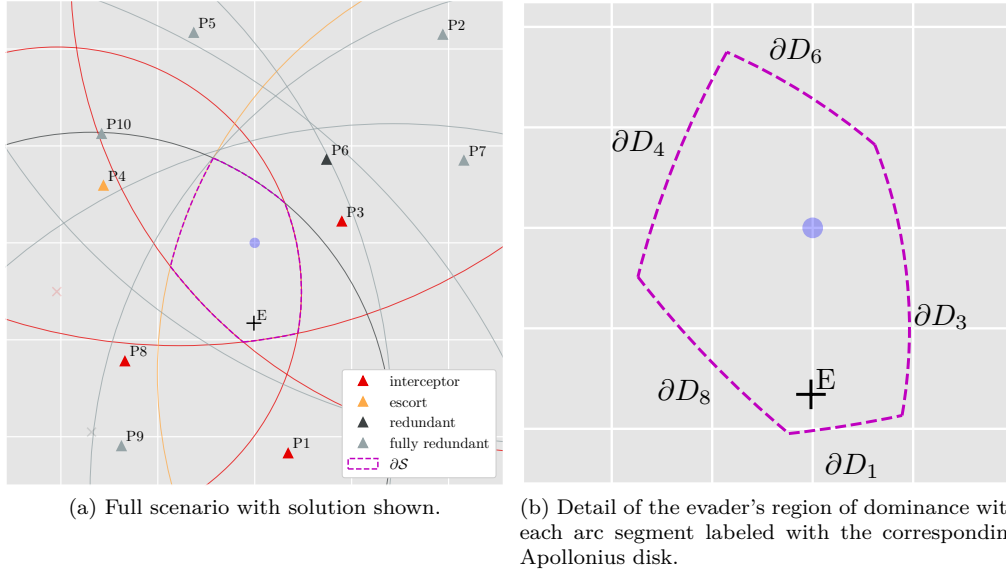


Fig. 6 Categories of pursuers. The color of each pursuer with its Apollonius circle denote which category it belongs to. The boundary of the evader’s safety region is rendered in dashed magenta.

If capture is constrained to occur in \mathcal{S} , then these pursuers have no effect on the game. They can be discarded completely; we refer to this set as “fully redundant”. Then the set of pursuers who share an edge of the evader’s cell of \mathbb{V}_E is given by

$$\mathcal{I}_{\mathcal{S}} = \{1, \dots, M\} \setminus \mathcal{I}^-. \quad (41)$$

The set $\mathcal{I}_{\mathcal{S}} \setminus \mathcal{I}$ contains pursuers which neighbor the evader in \mathbb{V}_E and can be further broken down by recomputing the solution (i.e., optimal intercept point I and capture time T) and checking if the new solution changed. Let the new solution of the game with the i th pursuer removed be denoted as $I^{\setminus i}$ and $T^{\setminus i}$ for the intercept point and capture time, respectively. If $T^{\setminus i} = T$ and $I^{\setminus i} = I$, removal of pursuer i had no effect on the game and thus it is “redundant”; we denote these pursuers as \mathcal{I}_R . Alternatively, we have $T^{\setminus i} > T$ or $I^{\setminus i} \neq I$, and thus removal of pursuer i led to a decrease in the pursuers’ performance or a change in the optimal intercept point, respectively. These pursuers are referred to as “escorts”, denoted \mathcal{I}_E . It is necessary for the escorts to “play the game” by heading towards I ; alternatively they could simply implement pure pursuit. Their purpose is to control the dynamic shape of \mathcal{S} so that no new Apollonius circle intersection becomes feasible or advantageous for the evader. Because they do not actually participate in capture, it is difficult to say precisely what is optimal for these pursuers [25]. One interesting research question is what an escort could or should do when it is an escort for more than one evader in an M on N pursuit-evasion scenario.

Figure 6 contains an example wherein all the categories are represented. In this example, simultaneous capture is achieved by P1, P3, and P8. If P4 were removed from the game, the evader could flee to the Apollonius intersection directly to its left and reach it safely, thus increasing capture time. Therefore, P4 must play the game by shaping \mathcal{S} . P6 neighbors E in \mathbb{V}_E which is homologous to stating that part of the boundary of its Apollonius circle coincides with the boundary of \mathcal{S} . Removal of P6 from the game would make other Apollonius circle intersections safe/feasible for the evader, however these points are suboptimal. Thus, P6 does not have any effect on the solution provided P1, P3, P4, and P8 behave optimally.

Fortunately, the set \mathcal{I}^- can be determined prior to solving (39). In Figure 6, the safety region \mathcal{S} lies entirely inside the interior of the fully redundant pursuers’ Apollonius circles. Thus, in this

case $M = 10$, five may be discarded immediately. The following section presents an algorithm for computing the set \mathcal{I}_S , which are the pursuers who are neighbors of the evader in \mathbb{V}_E . This reduced set includes all the pursuers who could possibly affect the solution. Interestingly, in this example, using the linear program method yields a different capture point wherein P3, P4, and P8 are interceptors and the capture time is 0.4% worse for the evader.

4 Geometric Algorithms

For general initial conditions, the size of the set of pursuers who are neighbors of E in \mathbb{V}_E , namely $|\mathcal{I}_S|$, is usually close to four. This could be true even when M is very large (say, in the thousands). However, $|\mathcal{I}_S|$ may be large when the positions of the pursuers are highly correlated. The extreme case would be when the pursuers lie on a ring (i.e., are equidistant) from the evader. In that case, $|\mathcal{I}_S| = M$ and all the Apollonius circle will contribute edges to \mathcal{S} . For the algorithm to follow, this will yield worst-case performance. In this section, we briefly describe and summarize two algorithms: an algorithm which simultaneously computes \mathcal{I}_S and \mathcal{S} , and another which computes the optimal intercept point (according to (39)) given \mathcal{S} . The paper [25] contains a complete specification of these algorithms including some supporting lemmas and analysis of the computational complexity.

First note that \mathcal{S} corresponds to the evader's cell of \mathbb{V}_E , which is the multiplicatively-weighted Voronoi diagram consisting of all the agents [25]. Part of the motivation for computing \mathcal{S} quickly is the fact that existing approaches (like computing the entire multiplicatively-weighted Voronoi diagram) can be $O(M^2)$ [1]. Note also that to construct \mathcal{S} , we are essentially computing the union of M disks, for which there is an algorithm which takes $\Theta(M \log M)$ time [5]. The algorithm described here differs in that we do not use any geometric transformations. We refer to the algorithm which computes \mathcal{S} and \mathcal{I}_S as `EVADERCELL`. The `EVADERCELL` algorithm takes as input the positions of all the agents and the speed ratio α . First, the Apollonius circles for each pursuer are computed. Then the pursuers are ranked according to the minimum distance from the evader to a point on their Apollonius circle. This measure is useful because it takes into account both the distance from the pursuer to the evader as well as its speed. The latter piece makes this algorithm applicable in the case where the pursuers have different velocities. However, for the present case of equal-speed pursuers, we could simply rank pursuers according to distance from the evader. Note, for computational speed this ranking is achieved using a heap, in lieu of a full sort, since, in the general case, not all the pursuers require consideration. `EVADERCELL` constructs \mathcal{S} iteratively, considering a single Apollonius circle in each iteration. Initially, the region \mathcal{S} is initialized as the Apollonius circle of the closest pursuer. Then, as pursuers (and their Apollonius circles) are dequeued from the heap `EVADERCELL` computes the intersection of the new Apollonius circle with the current region \mathcal{S} . Eventually, the closest point on a pursuer's Apollonius circle will be further from the evader than the furthest point on \mathcal{S} , then we know that this pursuer, and all subsequent pursuers in the heap, fall into the *fully redundant* category [25]. Therefore, the construction of \mathcal{S} is complete and \mathcal{I}_S is simply the set of pursuers that had been dequeued previously. Thus, `EVADERCELL` returns the ordered sets of vertices and arc segments $(\mathcal{V}_S, \mathcal{E}_S)$ and \mathcal{I}_S .

Now, we summarize several of the lemmas from [25] which affirm the correctness of `EVADERCELL`.

1. The pursuer closest to E is in \mathcal{I}_S and has an Apollonius circle which comes the closest to E .
2. The closest distance of the pursuers' Apollonius circles is monotonically non-decreasing with each iteration (due to the queue/heap property).
3. The point on $\partial\mathcal{S}$ furthest from the evader is either at the intersection of two Apollonius circles or corresponds to the solution to one of the one-on-one games.
4. The distance from the evader to the point on $\partial\mathcal{S}$ furthest away is monotonically non-increasing with each iteration.
5. None of the remaining pursuers' Apollonius circles intersect \mathcal{S} once we reach a pursuer whose Apollonius circle's minimum distance to the evader is greater than the furthest point from the evader on $\partial\mathcal{S}$.

6. There can be at most one one-on-one optimal intercept point on $\partial\mathcal{S}$.
7. If a one-on-one optimal intercept point lies on $\partial\mathcal{S}$, it is the optimal intercept point for the overall game.

Once \mathcal{S} has been constructed computing the optimal intercept point, which is the solution to (39), is very straightforward. We refer to this process as MPURSUELEVADER [25]. The MPURSUELEVADER algorithm is based entirely off of Theorem 2, and therefore proceeds by computing all the candidate solutions and comparing the distance from the nearest pursuer to each candidate. From Theorem 2 the optimal intercept point is

$$(x^*, y^*) = \underset{(x,y) \in \mathcal{P}_S^1 \cup \mathcal{V}_S \cup \mathcal{V}_{P_S}}{\arg \max} \min_{i \in \{1, \dots, M\}} \|(x, y) - P_i\| \alpha_P, \quad (42)$$

where \mathcal{P}_S^1 is the set of single pursuer optimal intercept points in \mathcal{S} , \mathcal{V}_S are the vertices of the evader's region of dominance (given by EVADERCELL), and \mathcal{V}_{P_S} is the set of vertices of the pursuer-only Voronoi diagram inside \mathcal{S} . Due to the seventh item in the previous paragraph, the first thing to do is check whether any of the one-on-one optimal intercept points are on $\partial\mathcal{S}$. Because the pursuers share the same speed, it is either the case that the solution to the one-on-one game between the pursuer closest to the evader and the evader is on $\partial\mathcal{S}$ or none of the one-on-one solutions are on $\partial\mathcal{S}$. Thus this step is trivial. If it is not the case that the optimal intercept point of a one-on-one game is on $\partial\mathcal{S}$, then one must turn to the other candidates. As mentioned previously, the points \mathcal{V}_S are given by EVADERCELL. The last subset of candidate solutions to compute, then, is \mathcal{V}_{P_S} for which we first need to compute the Voronoi vertices of \mathbb{V}_P . Because we assume that EVADERCELL is used prior to this point we can essentially forget about any of the pursuers not in the set \mathcal{I}_S . To compute the Voronoi vertices between the pursuers we employ Fortune's Algorithm [13] over the pursuers in \mathcal{I}_S . Then, to get \mathcal{V}_{P_S} we simply check to see if each Voronoi vertex is inside or outside of \mathcal{S} . Finally, the distance from each candidate in $\mathcal{V}_S \cup \mathcal{V}_{P_S}$ to its nearest pursuer is computed (RHS of (42)), and the optimal intercept point is the candidate which maximizes this distance. The terminal time is simply the travel time of the pursuer nearest that point via a straight-line path.

5 Results

The goal of this investigation is to consider full cooperation among the pursuers. Alternatively, the pursuers could, for example, act entirely independently of one another. In the latter case, each pursuer may implement its best strategy in the sense of one-on-one, that is Pure Pursuit (PP). We refer to the approach detailed here and in [25] as the geometric (G) policy. Note that we did not claim that G policy, if implemented by the pursuers and the evader, is a saddle-point pair of strategies in the sense of a feedback Nash equilibrium. Therefore, some numerical simulations are included here in order to demonstrate some of the merits of the G policy in comparison to other strategies, namely PP. The comparison of these two policies is done using a discrete time numerical simulation in which agents evaluate their respective control policies at each time step. For those agents implementing the G policy, the optimal intercept point (x^*, y^*) is computed using the current positions of the agents, and then those agents' heading would aim towards this point. Also, if the evader, implementing G, reaches the intercept point before any pursuer, then it will stand still (provided that point remains the solution to (42)).

Figure 7 displays the trajectories produced by each policy pair. In these simulations, the time step Δt is 0.001, the pursuers speed is 1, and the evader speed is 0.8 (and $\alpha = 0.8$ as well). From $t = 0$ the solution to (42) is the Voronoi vertex created by P1, P2, and P3. Thus, all three pursuers are required to achieve the capture time of 0.57 seen in Figure 7a. Consequently, if the pursuers act independently and implement PP, the evader, implementing G, can increase the capture time to 0.86, an increase of over 50%. Observe that in Figures 7a and 7b P2's trajectory is quite similar. In the latter, however, P2 is not able to catch all the way up to E because P1 and P3 do not block the evader from passing

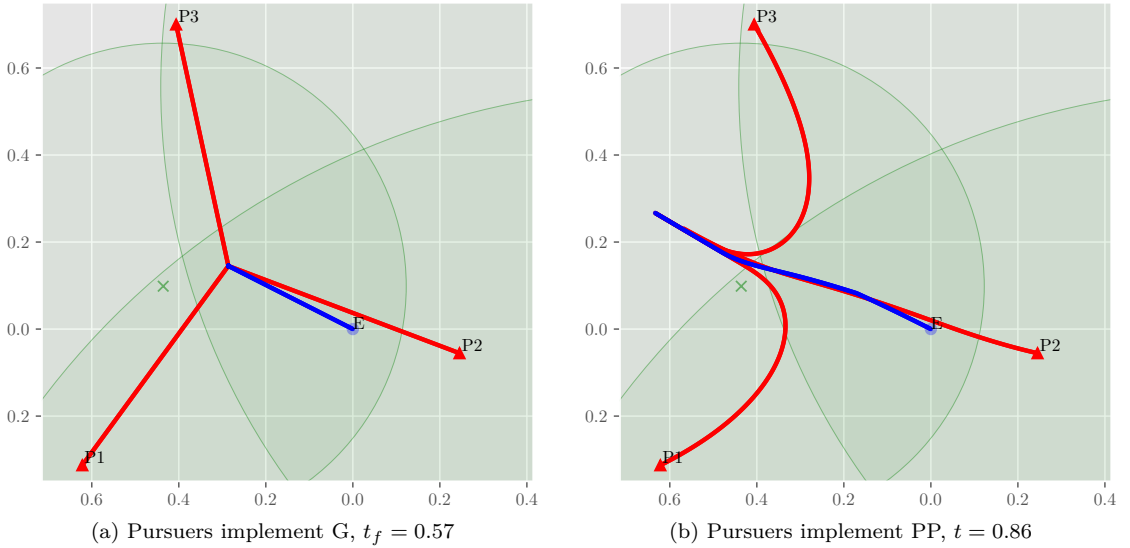


Fig. 7 Simulation results; evader implements G and pursuers implement G (a) or PP (b).

between them. The drastic difference between P1 and P3's headings with respect to E and with respect to the Voronoi vertex at $t = 0$ is what allows the evader to capitalize on their behavior. These results are included merely to highlight the benefits of cooperation and to demonstrate how the G policy achieves it.

The next set of examples demonstrate the non-uniqueness of the evader's trajectory in cases when the solution to (19) is a Voronoi vertex, v^* . In this case, all the evader must do is take a safe path to v^* (according to Property 1) in order to guarantee capture at time t^* . Meanwhile, the pursuers will implement the G policy – that is, at every time step, they will call `MPURSUER1EVADER` to compute their headings. As a consequence of the evader taking a safe path, the point (x^*, y^*) will remain invariant and the pursuers' trajectories will be straight. Figure 8 shows the trajectories for two different simulations. In Figures 8a and 8b, the evader's policy is a random walk with some additional logic to ensure v^* remains inside the instantaneous safe region \mathcal{S} . In Figures 8c and 8d, the evader's policy is such that the evader spirals in towards v^* ; once the evader can reach v^* within a single time step, it simply heads there at maximum speed. It is difficult, in general, to guarantee the safety of a path, *a priori*, and hence the evader policy prescribed by Theorem 1 wherein the evader takes a straight-line path at maximum speed to v^* and then stands still is useful.

The selection of the timestep Δt certainly has an effect on the payout of the game, and is an important consideration not only for simulation but also for real-world control implementations. For sufficiently large timesteps, capture may not even be possible! However, for the scenarios included in this section we note that the results do not change appreciably for $\Delta t = 0.01$ versus $\Delta t = 0.001$. For even smaller timesteps ($\Delta t = 1e-4$), the difference is even smaller. Therefore, as $\Delta t \rightarrow 0$, the capture time asymptotically approaches the value it would have in continuous time. We claim that the results we present for $\Delta t = 0.001$ are sufficiently close to this asymptotic value.

6 Conclusion

In this paper, we considered the problem of multiple agents pursuing a single evader wherein all the agents have simple motion. This problem is a direct extension of the two-pursuer one-evader problem originally posed by Isaacs [19] and verified formally in [16]. The intent is to exploit the benefits of

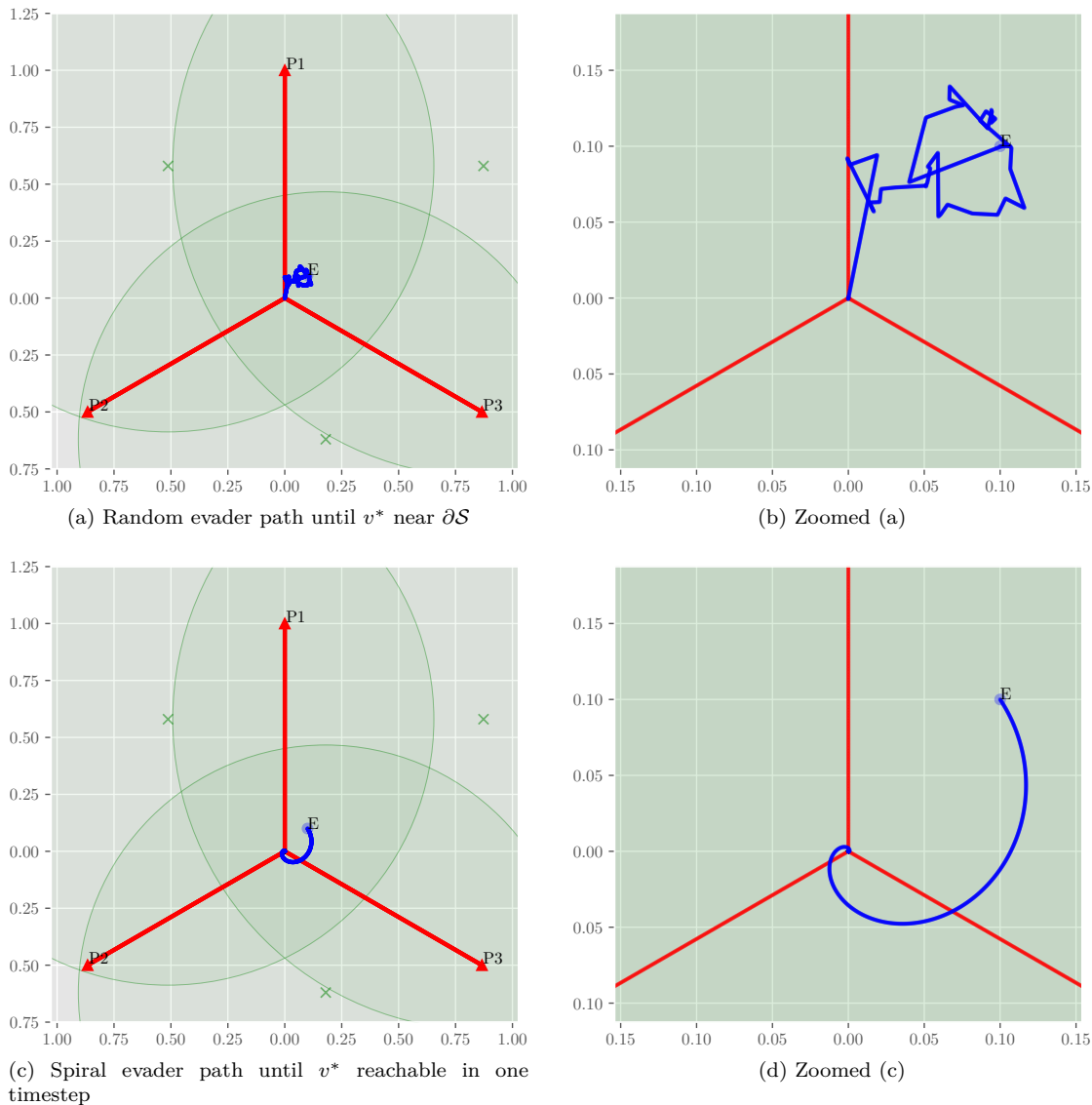


Fig. 8 Examples of ‘optimal’ evader trajectories resulting in capture at the Voronoi vertex demonstrating non-uniqueness of the path. Evader begins at $(0.1, 0.1)$, $V_P = 1$, $\alpha = 2/3$, and $t^* = 1$ s.

cooperation among a team of three or more pursuers. Intuitively, the presence of additional pursuers was shown to reduce the capture time of the evader.

The initial problem formulation and analysis highlights the difficulty in analyzing this problem using the techniques of Isaacs. Part of the issue is the curse of dimensionality brought about by including additional pursuers. A general strategy in differential games is to reduce the state-space into at most three dimensions, which cannot be done here. In lieu of a full verified feedback-optimal differential game solution we pursued a route of analysis which yielded open-loop optimal strategies for the pursuers and evader. These strategies are open-loop in the sense that they depend only on the initial conditions and are not necessarily robust to all other choice of opponent strategy when

implemented as a state-feedback policy. Isaacs' methods pertain to analysis of regular solutions; special care must be given to singularities wherein Isaacs' methods may not yield an optimal control action for one or more agents. It is likely, with the increased number of agents, that there are more singular surfaces in the M-pursuer one-evader differential game, requiring a more careful analysis.

Subsequent analysis relies on the pursuers taking straight-line paths to the capture point, which is suggested by the derived necessary conditions for optimality as well as the solutions to both the one-pursuer and two-pursuer versions of the game. Then, a linear program was posed to determine which capture location is optimal. However, a strong geometric interpretation of the problem was observed which resulted from the fact that pursuers take straight-line paths. Thus, a geometric solution was prescribed based on two Voronoi diagrams which allows searching over a discrete set of candidate solutions, as opposed to a continuous space as in the linear program case. Finally, algorithms were presented to compute the solution efficiently while also providing the precise shape of the evader's region of dominance.

As presented, the pursuer and evader strategies are very useful, however a full solution to the M-pursuer one-evader differential game is still sought. The results of this paper will help in addressing the M-pursuer N-evader scenario which may be broken down into M-pursuer one-evader subgames whose outcomes determine some sort of optimal pursuer assignments and teaming combinations.

References

1. Aurenhammer, F., Edelsbrunner, H.: An optimal algorithm for constructing the weighted voronoi diagram in the plane. *Pattern Recognition* **17**(2), 251–257 (1984). DOI 10.1016/0031-3203(84)90064-5
2. Bakolas, E., Tsiotras, P.: Relay pursuit of a maneuvering target using dynamic Voronoi diagrams. *Automatica* **48**(9), 2213–2220 (2012). DOI 10.1016/j.automatica.2012.06.003
3. Başar, T., Olsder, G.J.: Chapter 8: Pursuit-Evasion Games. In: *Mathematics in Science and Engineering, Dynamic Noncooperative Game Theory*, vol. 160, pp. 344–398. Elsevier (1982). DOI 10.1016/S0076-5392(08)62960-4
4. Breakwell, J.V., Hagedorn, P.: Point capture of two evaders in succession. *Journal of Optimization Theory and Applications* **27**(1), 89–97 (1979). DOI 10.1007/BF00933327
5. Brown, K.Q.: Geometric Transforms for Fast Geometric Algorithms. Tech. Rep. CMU-CS-80-101, Carnegie-Mellon Univ Pittsburgh PA Dept of Computer Science (1979)
6. Cheung, W.A.: Constrained pursuit-evasion problems in the plane. Ph.D. thesis, University of British Columbia (2005)
7. Conway, J.B.: *A Course in Functional Analysis*. Graduate Texts in Mathematics. Springer-Verlag, New York (1985)
8. Dobrin, A.: A review of properties and variations of Voronoi diagrams. Whitman College (2005)
9. Earl, M.G., D'Andrea, R.: A decomposition approach to multi-vehicle cooperative control. *Robotics and Autonomous Systems* **55**(4), 276–291 (2007). DOI 10.1016/j.robot.2006.11.002
10. Festa, A., Vinter, R.B.: A decomposition technique for pursuit evasion games with many pursuers. In: 52nd IEEE Conference on Decision and Control, pp. 5797–5802 (2013). DOI 10.1109/CDC.2013.6760803
11. Festa, A., Vinter, R.B.: Decomposition of Differential Games with Multiple Targets. *Journal of Optimization Theory and Applications* **169**(3), 848–875 (2016). DOI 10.1007/s10957-016-0908-z
12. Fisac, J.F., Sastry, S.S.: The pursuit-evasion-defense differential game in dynamic constrained environments. In: 2015 54th IEEE Conference on Decision and Control (CDC), pp. 4549–4556 (2015). DOI 10.1109/CDC.2015.7402930
13. Fortune, S.: A sweepline algorithm for Voronoi diagrams. *Algorithmica* **2**(1-4), 153 (1987). DOI 10.1007/BF01840357
14. Fuchs, Z.E., Khargonekar, P.P., Evers, J.: Cooperative defense within a single-pursuer, two-evader pursuit evasion differential game. In: 49th IEEE Conference on Decision and Control (CDC), pp. 3091–3097 (2010). DOI 10.1109/CDC.2010.5717894
15. Ganebny, S.A., Kumkov, S.S., Le Méneç, S., Patsko, V.S.: Model Problem in a Line with Two Pursuers and One Evader. *Dynamic Games and Applications* **2**(2), 228–257 (2012). DOI 10.1007/s13235-012-0041-z
16. Garcia, E., Fuchs, Z.E., Milutinović, D., Casbeer, D.W., Pachter, M.: A Geometric Approach for the Cooperative Two-Pursuer One-Evader Differential Game. *IFAC-PapersOnLine* **50**(1), 15209–15214 (2017). DOI 10.1016/j.ifacol.2017.08.2366
17. Huang, H., Zhang, W., Ding, J., Stipanović, D.M., Tomlin, C.J.: 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*, pp. 4835–4840. IEEE (2011)

18. Isaacs, R.: Games of Pursuit. Product Page P-257, RAND Corporation, Santa Monica, CA (1951)
19. Isaacs, R.: Differential Games: A Mathematical Theory with Applications to Optimization, Control and Warfare. Wiley, New York (1965)
20. Liu, S.Y., Zhou, Z., Tomlin, C., Hedrick, K.: Evasion as a team against a faster pursuer. In: 2013 American Control Conference, pp. 5368–5373 (2013). DOI 10.1109/ACC.2013.6580676
21. Oyler, D.: Contributions To Pursuit-Evasion Game Theory. Ph.D. thesis, University of Michigan (2016)
22. Oyler, D.W., Kabamba, P.T., Girard, A.R.: Pursuit-evasion games in the presence of obstacles. *Automatica* **65**(Supplement C), 1–11 (2016). DOI 10.1016/j.automatica.2015.11.018
23. Pierson, A., Wang, Z., Schwager, M.: Intercepting Rogue Robots: An Algorithm for Capturing Multiple Evaders With Multiple Pursuers. *IEEE Robotics and Automation Letters* **2**(2), 530–537 (2017). DOI 10.1109/LRA.2016.2645516
24. Sun, W., Tsiotras, P., Lolla, T., Subramani, D.N., Lermusiaux, P.F.: Multiple-Pursuer/One-Evader Pursuit–Evasion Game in Dynamic Flowfields. *Journal of guidance, control, and dynamics* (2017)
25. Von Moll, A., Casbeer, D.W., Garcia, E., Milutinović, D.: Pursuit-evasion of an Evader by Multiple Pursuers. In: 2018 International Conference on Unmanned Aircraft Systems (ICUAS). Dallas, TX (2018). DOI 10.1109/ICUAS.2018.8453470