# Multiplayer Target-Attacker-Defender Games: Pairing Allocations and Control Strategies for Guaranteed Intercept

Vishnu S. Chipade* and Dimitra Panagou†

*University of Michigan, Ann Arbor, MI, 48109, USA* ‡

**This paper discusses a multiplayer Target-Attacker-Defender (TAD) game. We capture multiple objectives for the players using approximations of the minimum and maximum functions, and derive analytical, Lyapunov-based control strategies for the players. The solutions obtained for 1 defender vs 1 attacker, and 1 defender vs 2 attackers TAD games, under mild assumptions, are utilized to pair defenders with attackers to maximize the number of attackers captured. A modified pairing algorithm is also proposed which considers the 1 defender vs 2 attackers cases and assigns pairings to minimize the total active time for the defenders. Some conditions for the guaranteed capture of the attackers are also provided.**

## Nomenclature

| | |
|---|---|
| $r_{A_i}$ | Position vector of the $i^{th}$ attacker in global frame |
| $r_{D_j}$ | Position vector of the $j^{th}$ defender in global frame |
| $r_{A_i0}$ | Initial position vector of the $i^{th}$ attacker in global frame |
| $r_{D_j0}$ | Initial position vector of the $j^{th}$ defender in global frame |
| $r_{D_jA_i}$ | Position vector joining the $j^{th}$ defender to the $i^{th}$ attacker |
| $r_{A_iT}$ | Position vector joining the $i^{th}$ attacker to the target |
| $R_c$ | Radius of capture |
| $R_s$ | Sensing radius |
| $R_{AD}$ | Minimum distance that the attackers want to keep from the defenders |
| $RoV_A$ | Region of victory for the attackers |
| $RoV_D$ | Region of victory for the defenders |
| $RoV_A^u$ | Estimate of the region of victory for the attackers |
| $RoV_D^l$ | Estimate of the region of victory for the defenders |
| $u_{A_i}$ | Control vector of the $i^{th}$ attacker |
| $u_{D_j}$ | Control vector of the $j^{th}$ defender |
| $u_{max_{A_i}}$ | Maximum bound on the norm of the $i^{th}$ attacker's control vector |
| $u_{max_{D_j}}$ | Maximum bound on the norm of the $j^{th}$ defender's control vector |
| $v_{ij}$ | Objective function for the $i^{th}$ attacker corresponding to the distance from the $j^{th}$ defender |
| $v_{iT}$ | Objective function for the $i^{th}$ attacker corresponding to the distance from the target |
| $\bar{v}_{ij}$ | Objective function for the $j^{th}$ defender corresponding to the distance from the $i^{th}$ attacker |
| $V_A$ | Collective objective function for the attacker's team |
| $V_D$ | Collective objective function for the defender's team |

*PhD Pre-candidate, Department of Aerospace Engineering, 1320 Beal Avenue, University of Michigan, Ann Arbor, MI 48109, vishnuc@umich.edu

†Assistant Professor, Department of Aerospace Engineering, 1320 Beal Avenue, University of Michigan, Ann Arbor, MI 48109, dpanagou@umich.edu

‡This work has been funded by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF Award No. 1738714 along with significant contributions from C-UAS industry members.

| | |
|---|---|
| $\beta^{ji}$ | Angle between the vectors from the $j^{th}$ defender to the $i^{th}$ attacker and from the $i^{th}$ attacker to the target |
| $\gamma_{ji}$ | Speed ratio of the $j^{th}$ defender and the $i^{th}$ attacker |

# I. Introduction

Pursuit-evasion differential games is a widely studied area.[1] Extending pursuit-evasion games to multiple players has also attracted a lot of attention from researchers.[2,3] A variation of the pursuit-evasion game is a target-attacker-defender (TAD) game, in which the attackers (evaders) aim to reach a target while evading from the defenders (pursuers), who aim to defend the target. The problem of defending a target area from a swarm of attacking agents is of great of importance today, particularly because of the increasing advancements in, and the popularity of swarm technologies. In general, solving the multiplayer TAD games is numerically intractable; approaches based on Hamilton–Jacobi–Isaacs (HJI) reachability analysis give optimal solutions to differential games,[4] but as the number of players increases the complexity grows exponentially. In fact, for more than 2 players in 2D, the approach becomes numerically intractable.[4] To tackle this problem, several approximate methods have been employed in the literature.[3,5–7] Although these approaches reduce the complexity of the problem, computational requirement still persists, except for the work in,[3] which provides Lyapunov-like solutions in closed-form. However, the work in[3] focuses on the pursuit-evasion problem, which is different from the TAD game that additionally considers the capturing of the target by the attackers, which in turn adds to the complexity of the problem.

In this paper, we use a "divide and conquer" approach similarly to.[6,7] In this earlier work, instead of solving the complete multiplayer differential game, the authors considered 1 defender vs 1 attacker games, and used the solutions to pair defenders with attackers so that the number of captured attackers is maximized. The pairing is done using the Hungarian algorithm. Based on this work, and to avoid the computational burden of solving for the 1 vs 1 optimal control problem, we propose suboptimal but numerically tractable control strategies for the players, based on the work in.[3,8] This yields pure pursuit control strategies for the defenders in the 1 defender vs 1 attacker game (called 1-1 TAD game hereafter). We solve 1-1 TAD games and pair defenders with attackers using the Hungarian algorithm. We also consider 1 defender vs 2 attackers games, without increasing much the computational burden at the pairing stage, and search for solutions that may reduce the total time for which the defenders are actively engaged in capturing the attackers (called as active time hereafter).

### Relevant Literature

Reachability analysis based on the HJI equation is one of the popular approaches. Authors in[4,9] discuss the "capture the flag" game between one attacker and one defender, solved using HJI equation-based reachable set analysis and level set methods.[10] However, scaling this approach to the multiplayer case suffers from the curse of dimensionality. As an alternative, researchers have utilized the HJI equation-based solution for 1-1 reach-avoid games to solve multiplayer reach-avoid games by using the idea of maximum matching on bipartite graphs, which consist of defenders and attackers as the nodes.[5,6] Several numerical tools such as[11,12] are available to solve the 1-1 dynamic game in an optimal setting. The authors of[6,13] also developed a path defense approach, which uses a 2D slice of the 4D HJI reach-avoid set to solve for 1-1 game and thus reduce the computational requirements. Building upon similar lines, authors in[7] developed time optimal control strategies for 1-1 TAD game using the isochrones-based method, and used the time information from these games to pair defenders with attackers. Although this method is less computationally intensive compared to HJI-based solutions, it yields open loop control strategies, and may require frequent pairing updates to improve performance.

Another approach of solving such problems uses optimal control theory.[14–16] Garcia et al.[14] developed a cooperative strategy for the team of one target and one defender aircraft against a pure-pursuit strategy of an attacking missile, whereas in[15] the attacker was assumed to use proportional navigation. The objective is to maximize the separation between target and attacker at the instant of capture of the attacker by the defender. However, these papers consider only one attacker and one defender, while the solution requires backwards integration of the system trajectories from an unknown final time. Authors in[17] considered one defender and two attackers. MPC-based control strategy is developed in[18] for harbor defense, wherein three different worst-case strategies for the attackers are considered, and MPC controllers are developed for the defenders. However, no analytical guarantees are provided on the solutions.

The authors in[19] have applied a Voronoi partitioning based approach for multiple pursuers and one evader game. The same has been extended to the case of multiple evaders in.[20] The capture of one or more evaders is guaranteed by minimizing the area of the Voronoi cells of the attackers.

To tackle the curse of dimensionality of the HJI-based approaches for multiplayer games, the authors in[3] used Lyapunov-like analysis to develop control strategies that are based on continuously differentiable approximations of the minimum and maximum functions. The control strategies are developed by either maximizing or minimizing the growth of the corresponding Lyapunov-like function.

In our work, we use an approach similar to[3] to develop control strategies for the players, and we pair the defenders with the attackers similarly to the work in.[6,7] The novelties and contributions of this paper compared to the similar literature are as follows:

- Suboptimal but numerically tractable control strategies for the players are proposed using approximation functions, for which analytical solutions can be obtained for the 1-1 TAD games under certain assumptions.
- The attacker's intention to stay safe by avoiding defenders while attacking the target is considered.
- A pairing algorithm is proposed that also considers 1-2 TAD games in addition to 1-1 TAD games, while pairing the defenders with the attackers to minimize the total time for which the defenders are actively engaged in capturing the attackers.
- For a given number of defenders, conditions on guaranteed capture of certain number of attackers are provided.

The paper is organized as follows: Section II describes the mathematical modeling and the problem statement. The proposed approach and the analysis is discussed in Section III, followed by simulation results in Section IV to demonstrate the efficacy of the approach. Section V provides conclusion and our thoughts on future work.

## II. Problem Formulation

In this paper, a TAD differential game between $N_A$ attackers, $N_D$ defenders and a stationary target located at the origin is considered, shown in Fig. 1. The agents are modeled as particles under single integrator dynamics with their velocities bounded, as it is described in Eqs. (1)-(3).

$$\dot{r}_{A_i} = \begin{bmatrix} \dot{x}_{A_i} \\ \dot{y}_{A_i} \end{bmatrix} = \begin{bmatrix} u_{x_{A_i}} \\ u_{y_{A_i}} \end{bmatrix} = u_{A_i} \tag{1}$$

$$\dot{r}_{D_j} = \begin{bmatrix} \dot{x}_{D_j} \\ \dot{y}_{D_j} \end{bmatrix} = \begin{bmatrix} u_{x_{D_j}} \\ u_{y_{D_j}} \end{bmatrix} = u_{D_j} \tag{2}$$

$$\left.\begin{array}{l} \|u_{A_i}\| \leq u_{max_{A_i}} \quad \forall i \in I_A = \{1, 2, ..., N_A\} \\ \left\|u_{D_j}\right\| \leq u_{max_{D_j}} \quad \forall j \in I_D = \{1, 2, ..., N_D\} \\ \min_{j \in I_D} u_{max_{D_j}} > \max_{i \in I_A} u_{max_{A_i}} \end{array}\right\}, \tag{3}$$

where $r_{A_i}$, $r_{D_j}$ are the position vectors of the $i^{th}$ attacker and the $j^{th}$ defender, respectively; $u_{A_i}$, $u_{D_j}$ are their control velocity vectors, respectively, whose norms are bounded by $u_{max_{A_i}}$ and $u_{max_{D_j}}$. The third constraint in Eq. (3) expresses that all defenders are assumed to be faster than the attackers.

We assume that all the defenders know the positions of all the attackers that are inside a sensing zone (SZ) around the target. The sensing zone is modeled as a circular region of sensing radius $R_s$ around the target, $SZ = \{r \in \mathbb{R}^2 | \; \|r\| \leq R_s\}$.

The objective of attackers' team is to capture the target while avoiding the defenders, and that of the defenders' team is to capture all the attackers before any of them captures the target. This paper considers soft capture, referred to as just 'capture' hereafter. The capture happens at time $t$ if the distance between the

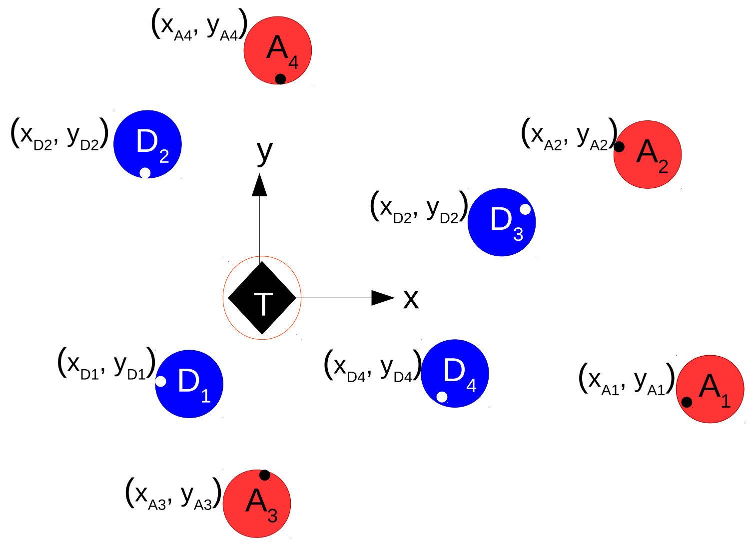



Figure 1: The global coordination frame and system configuration for the multi-player game.

corresponding agents becomes smaller than the capture radius $R_c$. For example, the $i^{th}$ attacker is captured by the $j^{th}$ defender at time $t$ if:

$$\sqrt{\left(x_{A_i}(t) - x_{D_j}(t)\right)^2 + \left(y_{A_i}(t) - y_{D_j}(t)\right)^2} \leq R_c,$$

or, the target is captured by the $i^{th}$ attacker if:

$$\sqrt{\left(x_{A_i}(t) - x_T\right)^2 + \left(y_{A_i}(t) - y_T\right)^2} \leq R_c.$$

In summary, the objectives of the attackers' team are:

- To ensure that at least one of the attackers captures the target, i.e., $\exists i \in \{1, 2, ..., N_A\}$ and $t_f \geq 0$ s.t. $\sqrt{\left(x_{A_i}(t_f) - x_T\right)^2 + \left(y_{A_i}(t_f) - y_T\right)^2} \leq R_c$.
- All the attackers should be at least $R_{AD}(> R_c)$ distance away from all the defenders at all times, i.e., $\forall i \in \{1, 2, ..., N_A\}$, $\sqrt{\left(x_{A_i}(t) - x_{D_j}(t)\right)^2 + \left(y_{A_i}(t) - y_{D_j}(t)\right)^2} \geq R_{AD}$, $\forall t$, $\forall j \in \{1, 2, ..., N_D\}$.

Similarly, the objectives of the defenders' team in the game are:

- To ensure that all the attackers are captured before any of them could reach the target, i.e., $\forall i \in \{1, 2, ..., N_A\}$, $\exists j \in \{1, 2, ..., N_D\}$ and $\exists t \geq 0$ s.t. $\sqrt{\left(x_{A_i}(t) - x_{D_j}(t)\right)^2 + \left(y_{A_i}(t) - y_{D_j}(t)\right)^2} \leq R_c$ AND $\sqrt{\left(x_{A_i}(t) - x_T\right)^2 + \left(y_{A_i}(t) - y_T\right)^2} \geq R_c$.

The goals of this paper are to develop control laws for 1-1 TAD games, and to pair defenders with attackers so that the number of captured attackers is maximized while the total active time of the defenders is minimized.

## III. Approach: Control Strategies and Pairing Algorithms

Solving multi-objective optimization problems is not easy in general, while the complexity of differential games increases significantly with the number of players. In this paper, we build upon the work by Stipanovic et al.,[3,8] which leverages approximations of the minimum and maximum functions to combine multiple objectives in a single scalar objective function that is then used to design collaborative control laws for the players. The approach yields a pure pursuit strategy for the defender in 1-1 TAD game.

For multi-player games we consider the following two cases: (a) Defenders use a collaborative control strategy obtained using the approximate functions approach. (b) Each defender applies a pure pursuit control corresponding to 1-1 TAD game with one of the attackers. It is found that the strategy in (b) performs better than the strategy in (a) when it comes to maximizing the number of captured attackers. For example, given equal number of attackers and defenders, capture of all the attackers starting in a region (discussed later in Theorem 1) is always guaranteed if the defenders apply the pure pursuit control corresponding to 1-1 TAD games with individual attackers. However, this is not true always for the collaborative control strategy based on approximate functions. As a consequence, we use pure pursuit control for the defenders, corresponding to 1-1 TAD game with individual attackers. We find the times of interception for 1-1 TAD games under the assumption of attackers not avoiding the defenders and use this knowledge to assign the attackers to the defenders. The defenders then can apply pure pursuit control corresponding to the 1-1 TAD game with the paired attacker.

In addition to this, we consider a game between 2 defenders and 2 attackers. Consider two cases: 1) Both defenders are active, i.e., one defender is chasing one attacker and the other defender is chasing the other attacker; 2) Only one defender is active, i.e., the active defender will chase the first attacker until it is captured and then it will chase the second attacker. For certain initial configurations of the attackers, it is possible that the total active time is less in case 2 as opposed to case 1. This can also be extended to the games with more than 2 defenders and attackers. The smaller active time for the defenders also reflects less energy consumption. So we also check for such 1 defender to 2 attackers (1→2) pairings which would reduce the total active time of the defenders while the number of captured attackers is maximized. We propose a modified pairing algorithm to check for 1→2 pairings of the defenders with the attackers in parallel to 1→1 pairings, and to assign pairings which minimize the total active time and maximize the number of captured attackers.

In the following section, we discuss the collaborative control strategy for attacker's team based on approximation functions. Although the global collaborative strategy for the defenders is not well suited in this case, we include that here for completeness.

### A. Control Strategies using Approximation Functions for Multiple Objectives

Every agent has multiple objectives to satisfy during the game. Accomplishment of these objectives can be ensured by developing control laws based on convergent approximations of minimum and maximum functions depending upon the formulation of the goals and objectives.[8]

Consider $N$ positive numbers $a_i$, $i \in \{1, 2, ..., N\}$. The minimum of these numbers can be approximated from below by a $\mathcal{C}^1$ function $\underline{\sigma}_\delta(.) : \mathbb{R}^N \to [0, +\infty)$ and from above by a $\mathcal{C}^1$ function $\overline{\sigma}_\delta(.) : \mathbb{R}^N \to [0, +\infty)$. For a given $\delta > 0$, $\underline{\sigma}_\delta$ and $\overline{\sigma}_\delta$ are given as:

$$\underline{\sigma}_\delta(a_1, a_2, ..., a_N) = \left(\frac{1}{\sum_{i=1}^N a_i^{-\delta}}\right)^{\frac{1}{\delta}} \qquad \& \qquad \overline{\sigma}_\delta(a_1, a_2, ..., a_N) = \left(\frac{N}{\sum_{i=1}^N a_i^{-\delta}}\right)^{\frac{1}{\delta}}. \tag{4}$$

Similarly, the maximum of those numbers can be approximated from below by a $\mathcal{C}^1$ function $\underline{\rho}_\delta(.) : \mathbb{R}^N \to [0, +\infty)$ and from above by a $\mathcal{C}^1$ function $\overline{\rho}_\delta(.) : \mathbb{R}^N \to [0, +\infty)$, which are given in Eq. (5),

$$\underline{\rho}_\delta(a_1, a_2, ..., a_N) = \left(\frac{\sum_{i=1}^N a_i^{\delta}}{N}\right)^{\frac{1}{\delta}} \qquad \& \qquad \overline{\rho}_\delta(a_1, a_2, ..., a_N) = \left(\sum_{i=1}^N a_i^{\delta}\right)^{\frac{1}{\delta}}. \tag{5}$$

Now consider that the $i^{th}$ agent has $N_i$ objectives, $v_{ij}(.) : \mathbb{R}^2 \to [0, +\infty)$, where $j$ denotes the $j^{th}$ objective and accomplishment of each objective is described by an inequality $v_{ij} \geq \epsilon_{ij}$ for some nonnegative numbers $\epsilon_{ij}$. If the $i^{th}$ agent wants to satisfy all the objectives, then the inequality constraint given in Eq. 6 is a conservative but sufficient condition for that,

$$\underline{\sigma}_\delta(v_{i1}, v_{i2}, ..., v_{iN_i}) \geq \max\{\epsilon_{i1}, \epsilon_{i2}, ..., \epsilon_{iN_i}\}. \tag{6}$$

Similarly, for the objectives of the type $v_{ij} \leq \epsilon_{ij}$ the sufficient condition is given in Eq. 7,

$$\overline{\rho}_\delta(v_{i1}, v_{i2}, ..., v_{iN_i}) \leq \min\{\epsilon_{i1}, \epsilon_{i2}, ..., \epsilon_{iN_i}\}. \tag{7}$$

On the other hand, if only one of the objectives is to be satisfied then the sufficient conditions for inequalities of the type $v_{ij} \geq \epsilon_{ij}$ and $v_{ij} \leq \epsilon_{ij}$ can be given respectively as:

$$\underline{\rho}_\delta(v_{i1}, v_{i2}, ..., v_{iN_i}) \geq \max\{\epsilon_{i1}, \epsilon_{i2}, ..., \epsilon_{iN_i}\}, \tag{8}$$

$$\overline{\sigma}_\delta(v_{i1}, v_{i2}, ..., v_{iN_i}) \leq \min\{\epsilon_{i1}, \epsilon_{i2}, ..., \epsilon_{iN_i}\}. \tag{9}$$

For more details, readers may refer to.[3,8] To incorporate the individual objectives of an attacker, we define two objective functions for each of the attackers as described in Eqs. (10) and (11). For attacker-defender pair, the objective function, defined in Eq. (10), is the ratio of the distance between the $i^{th}$ attacker and the $j^{th}$ defender and $R_{AD}$, where $R_{AD}$ is the distance that the attacker would want to maintain from the defender.

$$v_{ij} = \frac{\left\|r_{A_i} - r_{D_j}\right\|}{(R_{AD})} = \frac{\left(\sqrt{(x_{A_i} - x_{D_j})^2 + (y_{A_i} - y_{D_j})^2}\right)}{(R_{AD})} \geq 0. \tag{10}$$

To ensure that every objective combined using an approximation function is satisfied, the individual objectives require to satisfy same type of inequality. For example, to ensure that the $i^{th}$ attacker stays away from the $j^{th}$ defender we require $v_{ij} \geq 1$. If we want to combine $v_{iT}$ with $v_{ij}$ using approximation functions, then it is required to satisfy an inequality of type $v_{iT} \geq \epsilon_i$. So for attacker-target pair, the objective function $v_{iT}$ is defined as in Eq. (11) and it should satisfy $v_{iT} \geq 1$ for capturing the target.

$$v_{iT} = \frac{2R_c^2}{R_c^2 + \left\|r_{A_i} - r_T\right\|^2} = \frac{2R_c^2}{R_c^2 + (x_{A_i} - x_T)^2 + (y_{A_i} - y_T)^2} \geq 0. \tag{11}$$

Similarly for the defender, the objective function can be defined as in Eq. (12) and it should satisfy $\bar{v}_{ij} < 1$.

$$\bar{v}_{ij} = \frac{\left\|r_{A_i} - r_{D_j}\right\|}{(R_c)} = \frac{\left(\sqrt{(x_{A_i} - x_{D_j})^2 + (y_{A_i} - y_{D_j})^2}\right)}{(R_c)} \geq 0. \tag{12}$$

The objective of the $i^{th}$ attacker, for interception of the target and for evasion from the defenders is to ensure that $v_{iT} \geq 1$ **AND** $v_{ij} \geq 1$, $\forall j$ at some $t = t_f > 0$, whereas for interception of the $i^{th}$ attacker by the defenders before the attacker reaches the target, the defenders need to ensure that $\exists j \in \{1, 2, ..., N_D\}$ s.t. $v_{ij} \leq 1$ **AND** $v_{iT} < 1$ for some $t \in [0, t_f)$. In a collaborative manner, all the attackers would want to evade from the defenders and at least one of the attackers should capture the target. This cooperative objectives can be encoded using following set of approximation functions. For attackers, $V_A$ encodes all the cooperative objectives in one function,

$$\phi_\delta^{A_i} = \underline{\sigma}_\delta(v_{i1}, v_{i2}, ..., v_{iN_D}),$$
$$\eta_\delta^{A} = \underline{\rho}_\delta(v_{1T}, v_{2T}, ..., v_{N_AT}),$$
$$V_A = \underline{\sigma}_\delta(\phi_\delta^{A_1}, \phi_\delta^{A_2}, ..., \phi_\delta^{A_{N_A}}, \eta_\delta^{A}). \tag{13}$$

Here $\phi_\delta^{A_i}$ encodes the objective of the $i^{th}$ attacker of staying away from all the defenders using $\underline{\sigma}_\delta$, and $\eta_\delta^A$ combines the objective of at least one attacker capturing the target using $\underline{\rho}_\delta$. Since the attackers want to satisfy both of these objectives, they are combined together via $\underline{\sigma}_\delta$. Similarly, the defenders would want to intercept all the attackers before any of the attackers captures the target. For the team of defenders, the cooperative strategy can be encoded using $V_D$ as described in Eq. (14), where $\phi_\delta^{D_i}$ combines the objective of at least one defender capturing the $i^{th}$ attacker. Since all the attackers are to be captured, $\phi_\delta^{D_i}$'s are combined using $\overline{\rho}_\delta$. Defenders have no control over the distance of the attackers from the target, so there is no use of including that in the objective function.

$$\phi_\delta^{D_i} = \overline{\sigma}_\delta(\bar{v}_{i1}, \bar{v}_{i2}, ..., \bar{v}_{iN_D}),$$
$$V_D = \overline{\rho}_\delta(\phi_\delta^{D_1}, \phi_\delta^{D_2}, ..., \phi_\delta^{D_{N_A}}). \tag{14}$$

To satisfy the individual objectives, the attackers' team has to satisfy $V_A \geq 1$ whereas defenders' team has to make sure $V_D \leq 1$. This can be ensured, as proposed in,[8] by designing the control laws as follows:

$$\hat{u}_{A_i} = \underset{\left\|u_{A_i}\right\| \leq u_{max_{A_i}}}{\arg\max} \frac{dV_A}{dt} = u_{max_{A_i}} \left(\frac{\partial V_A}{\partial r_{A_i}}\right)^T \left\|\frac{\partial V_A}{\partial r_{A_i}}\right\|^{-1}, \tag{15}$$

$$\hat{u}_{D_j} = \underset{\left\|u_{D_j}\right\| \leq u_{max_{D_j}}}{\arg\min} \frac{dV_D}{dt} = -u_{max_{D_j}} \left(\frac{\partial V_D}{\partial r_{D_j}}\right)^T \left\|\frac{\partial V_D}{\partial r_{D_j}}\right\|^{-1}. \tag{16}$$

These control inputs basically drive the agents along the trajectories with the steepest change in the objective functions $V_A$ and $V_D$. The gradients of the objective functions in (13) and (14) are:

$$\frac{dV_A}{dr_{A_i}} = \frac{\partial V_A}{\partial \phi_\delta^{A_i}}\left[\sum_{j=1}^{N_D}\frac{\partial \phi_\delta^{A_i}}{\partial v_{ij}}\frac{\partial v_{ij}}{\partial r_{A_i}}\right] + \frac{\partial V_A}{\partial \eta_\delta^A}\frac{\partial \eta_\delta^A}{\partial v_{iT}}\frac{\partial v_{iT}}{\partial r_{A_i}}$$

$$\left.\begin{aligned}
\frac{\partial V_A}{\partial x_{A_i}} &= V_A^{(\delta+1)}\left[\sum_{j=1}^{N_D} v_{ij}^{(-\delta-1)}\left(\frac{x_{A_i}-x_{D_j}}{R_{AD}^2 v_{ij}}\right)\right] - V_A^{(\delta+1)}\frac{v_{iT}^{(-\delta-1)}\left(\eta_\delta^A\right)^{(\frac{1}{\delta}-\delta-2)}}{N_A}\left(\frac{(x_{A_i}-x_T)v_{iT}^2}{R_c^2}\right)\\
\frac{\partial V_A}{\partial y_{A_i}} &= V_A^{(\delta+1)}\left[\sum_{j=1}^{N_D} v_{ij}^{(-\delta-1)}\left(\frac{y_{A_i}-y_{D_j}}{R_{AD}^2 v_{ij}}\right)\right] - V_A^{(\delta+1)}\frac{v_{iT}^{(-\delta-1)}\left(\eta_\delta^A\right)^{(\frac{1}{\delta}-\delta-2)}}{N_A}\left(\frac{(y_{A_i}-y_T)v_{iT}^2}{R_c^2}\right)
\end{aligned}\right\} \quad (17)$$

Similarly for the defenders,

$$\frac{dV_D}{dr_{D_j}} = \sum_{k=1}^{N_A}\frac{\partial V_D}{\partial \eta_\delta^{D_k}}\frac{\partial \eta_\delta^{D_k}}{\partial \phi_\delta^{D_k}}\frac{\partial \phi_\delta^{D_k}}{\partial v_{kj}}\frac{\partial v_{kj}}{\partial r_{D_j}}$$

$$\left.\begin{aligned}
\frac{\partial V_D}{\partial x_{D_j}} &= \sum_{k=1}^{N_A}\frac{V_D^{(\frac{1}{\delta}-1)}\left(\phi_\delta^{D_k}\right)^{(\delta-1)}\left(\eta_\delta^{D_k}\right)^{(\frac{1}{\delta}+\delta-2)} v_{kj}^{(-\delta-1)}\left(\phi_\delta^{D_k}\right)^{(\delta+1)}}{N_D}\left(\frac{-(x_{A_k}-x_{D_j})}{R_{AD}^2 v_{kj}}\right)\\
\frac{\partial V_D}{\partial y_{D_j}} &= \sum_{k=1}^{N_A}\frac{V_D^{(\frac{1}{\delta}-1)}\left(\phi_\delta^{D_k}\right)^{(\delta-1)}\left(\eta_\delta^{D_k}\right)^{(\frac{1}{\delta}+\delta-2)} v_{kj}^{(-\delta-1)}\left(\phi_\delta^{D_k}\right)^{(\delta+1)}}{N_D}\left(\frac{-(y_{A_k}-y_{D_j})}{R_{AD}^2 v_{kj}}\right)
\end{aligned}\right\} \quad (18)$$

With these strategies, the attackers would move toward the target when defenders are not in their way, but they would deviate to avoid the defenders on the way in order to delay their capture by the defenders. However, for defenders, these strategies are basically weighted sum of the pure pursuit strategies corresponding to the individual attackers. As discussed earlier, we use pure pursuit strategies for the defenders corresponding to their 1-1 TAD games with individually paired attackers.

### B. Single Attacker-Target-Defender Game (1-1 TAD game): Strategy and Analysis

For a 1-1 TAD game, we have the objective functions as,

$$\left.\begin{aligned}
v_{11} &= \frac{\left\|r_{A_1}-r_{D_1}\right\|}{R_{AD}} = \frac{\sqrt{(x_{A_1}-x_{D_1})^2+(y_{A_1}-y_{D_1})^2}}{R_{AD}};\\
v_{1T} &= \frac{2R_c^2}{R_c^2+\left\|r_{A_1}-r_T\right\|^2} = \frac{2R_c^2}{R_c^2+(x_{A_1}-x_T)^2+(y_{A_1}-y_T)^2};\\
\bar{v}_{11} &= \frac{\left\|r_{A_1}-r_{D_1}\right\|}{R_c} = \frac{\sqrt{(x_{A_1}-x_{D_1})^2+(y_{A_1}-y_{D_1})^2}}{R_c}.
\end{aligned}\right\} \quad (19)$$

The overall objectives $V_A$ and $V_D$ are given as:

$$V_A = \left(\frac{1}{v_{11}^{-\delta}+v_{1T}^{-\delta}}\right)^{(1/\delta)}, \qquad V_D = \left(\bar{v}_{11}^{\delta}\right)^{(1/\delta)} = \bar{v}_{11}. \quad (20)$$

Their gradients are given as:

$$\frac{\partial V_A}{\partial x_{A_1}} = V_A^{(\delta+1)}\left(\frac{v_{11}^{-\delta-2}(x_{A_1}-x_{D_1})}{R_{AD}^2} - \frac{v_{1T}^{\frac{1}{\delta}-1}(x_{A_1}-x_T)}{R_c^2}\right), \qquad \frac{\partial V_A}{\partial y_{A_1}} = V_A^{(\delta+1)}\left(\frac{v_{11}^{-\delta-2}(y_{A_1}-y_{D_1})}{R_{AD}^2} - \frac{v_{1T}^{\frac{1}{\delta}-1}(y_{A_1}-y_T)}{R_c^2}\right); \quad (21)$$

$$\frac{\partial V_D}{\partial x_{D_1}} = \frac{-V_D^{(\delta+\frac{2}{\delta}-3)}\bar{v}_{11}^{\delta-2}(x_{A_1}-x_{D_1})}{R_{AD}^2}, \qquad \frac{\partial V_D}{\partial y_{D_1}} = \frac{-V_D^{(\delta+\frac{2}{\delta}-3)}\bar{v}_{11}^{\delta-2}(y_{A_1}-y_{D_1})}{R_{AD}^2}. \quad (22)$$

Let $\left\|\frac{\partial V_A'}{\partial r_{A_1}}\right\| = \frac{1}{V_A^{(\delta+1)}}\sqrt{\left(\frac{\partial V_A}{\partial x_{A_1}}\right)^2+\left(\frac{\partial V_A}{\partial y_{A_1}}\right)^2}$, then, for the attacker, the control velocities are given as:

$$\left.\begin{aligned}
u_{x_{A_1}} &= u_{max_{A_1}}\frac{\left(\frac{v_{11}^{-\delta-2}(x_{A_1}-x_{D_1})}{R_{AD}^2} - \frac{v_{1T}^{\frac{1}{\delta}-1}(x_{A_1}-x_T)}{R_c^2}\right)}{\left\|\frac{\partial V_A'}{\partial r_{A_1}}\right\|},\\
u_{y_{A_1}} &= u_{max_{A_1}}\frac{\left(\frac{v_{11}^{-\delta-2}(y_{A_1}-y_{D_1})}{R_{AD}^2} - \frac{v_{1T}^{\frac{1}{\delta}-1}(y_{A_1}-y_T)}{R_c^2}\right)}{\left\|\frac{\partial V_A'}{\partial r_{A_1}}\right\|}.
\end{aligned}\right\} \quad (23)$$

For the defender, the control velocities are given as:

$$
\left.\begin{aligned}
u_{x_{D_1}} &= u_{max_{D_1}} \frac{(x_{A_1}-x_{D_1})}{\sqrt{(x_{A_1}-x_{D_1})^2+(y_{A_1}-y_{D_1})^2}}, \\
u_{y_{D_1}} &= u_{max_{D_1}} \frac{(y_{A_1}-y_{D_1})}{\sqrt{(x_{A_1}-x_{D_1})^2+(y_{A_1}-y_{D_1})^2}}.
\end{aligned}\right\} \tag{24}
$$

For the defender, this approach yields pure pursuit control law corresponding to the attacker.

### 1. *Convergence Analysis*

In this subsection, we investigate the convergence of the defender and attacker to their respective objectives of capturing the attacker and the target. To analyze the capture of the attacker by the defender, we define a Lyapunov like function as given in Eq. (25) and analyze its time derivative.

$$
V_{DA} = \frac{\|r_{D_1A_1}\|^2}{2} = \frac{\|r_{A_1} - r_{D_1}\|^2}{2} = \frac{(x_{A_1} - x_{D_1})^2 + (y_{A_1} - y_{D_1})^2}{2} > 0 \qquad \& \; V_{DA} = 0 \text{ when } \|r_{D_1A_1}\| = 0 \tag{25}
$$

The time derivative of $V_{DA}$ is as follows:

$$
\begin{aligned}
\dot{V}_{DA} &= (x_{A_1} - x_{D_1})(u_{x_{A_1}} - u_{x_{D_1}}) + (y_{A_1} - y_{D_1})(u_{y_{A_1}} - u_{y_{D_1}}) \\
&\overset{(23),(24)}{=} u_{max_{A_1}} \frac{\left( \frac{v_{11}^{-\delta-2}(x_{A_1}-x_{D_1})^2}{R_{AD}^2} - \frac{v_{1T}^{\frac{1}{\delta}-1}(x_{A_1}-x_{D_1})(x_{A_1}-x_T)}{R_c^2} \right)}{\left\| \frac{\partial V_A'}{\partial r_{A_1}} \right\|} - u_{max_{D_1}} \frac{(x_{A_1}-x_{D_1})^2}{\sqrt{(x_{A_1}-x_{D_1})^2+(y_{A_1}-y_{D_1})^2}} \\
&\quad + u_{max_{A_1}} \frac{\left( \frac{v_{11}^{-\delta-2}(y_{A_1}-y_{D_1})^2}{R_{AD}^2} - \frac{v_{1T}^{\frac{1}{\delta}-1}(y_{A_1}-y_{D_1})(y_{A_1}-y_T)}{R_c^2} \right)}{\left\| \frac{\partial V_A'}{\partial r_{A_1}} \right\|} - u_{max_{D_1}} \frac{(y_{A_1}-y_{D_1})^2}{\sqrt{(x_{A_1}-x_{D_1})^2+(y_{A_1}-y_{D_1})^2}}, \\
&= \frac{u_{max_{A_1}} \left( v_{11}^{-\delta} - \frac{v_{1T}^{\frac{1}{\delta}-1}}{R_c^2} \left( (r_{A_1}-r_{D_1})^T (r_{A_1}-r_T) \right) \right)}{\left\| \frac{\partial V_A'}{\partial r_{A_1}} \right\|} - u_{max_{D_1}} \|r_{A_1} - r_{D_1}\|, \\
\dot{V}_{DA} &\leq u_{max_{A_1}} \|r_{D_1A_1}\| - u_{max_{D_1}} \|r_{D_1A_1}\|, \\
\dot{V}_{DA} &< 0, \quad \text{if } u_{max_{D_1}} > u_{max_{A_1}}.
\end{aligned} \tag{26}
$$

This implies that as long as the defender is faster than the attacker the distance between them would always keep decreasing with the given control laws i.e. $\exists \, t_c \geq 0$ s.t. $\|r_{D_1A_1}(t_c)\| \leq R_c$.

Similarly, to analyze the capture of the target by the attacker, we define a Lyapunov like function as given in Eq. (27) and analyze its time derivative.

$$
V_{AT} = \frac{\|r_{A_1T}\|^2}{2} = \frac{\|r_T - r_{A_1}\|^2}{2} = \frac{(x_T - x_{A_1})^2 + (y_T - y_{A_1})^2}{2} > 0 \qquad \& \; V_{AT} = 0 \text{ when } \|r_{A_1T}\| = 0 \tag{27}
$$

The time derivative of $V_{AT}$ is as follows:

$$
\begin{aligned}
\dot{V}_{AT} &= -(x_T - x_{A_1}) u_{x_{A_1}} - (y_T - y_{A_1}) u_{y_{A_1}} \\
&\overset{(23)}{=} u_{max_{A_1}} \frac{\left( \frac{-v_{11}^{-\delta-2}(x_{A_1}-x_{D_1})(x_T-x_{A_1})}{R_{AD}^2} - \frac{v_{1T}^{\frac{1}{\delta}-1}(x_T-x_{A_1})^2}{R_c^2} \right)}{\left\| \frac{\partial V_A'}{\partial r_{A_1}} \right\|} + u_{max_{A_1}} \frac{\left( \frac{-v_{11}^{-\delta-2}(y_{A_1}-y_{D_1})(y_T-y_{A_1})}{R_{AD}^2} - \frac{v_{1T}^{\frac{1}{\delta}-1}(y_T-y_{A_1})^2}{R_c^2} \right)}{\left\| \frac{\partial V_A'}{\partial r_{A_1}} \right\|} \\
\dot{V}_{AT} &= \frac{u_{max_{A_1}}}{\left\| \frac{\partial V_A'}{\partial r_{A_1}} \right\|} \left( - \frac{v_{11}^{-\delta-2}}{R_{AD}^2} \left( (r_{D_1A_1})^T (r_{A_1T}) \right) - \frac{v_{1T}^{\frac{1}{\delta}-1}}{R_c^2} \left( \|r_{A_1T}\| \right)^2 \right).
\end{aligned} \tag{28}
$$

In general, the definiteness of $\dot{V}_{AT}$ depends on the sign of the inner product $((r_{D_1A_1})^T(r_{A_1T}))$ and the relative magnitude of the two individual terms in the last expression of the Eq. (28). In particular, when the defender comes very close to the attacker and is in the direction of the target, then the first term would become positive and would dominate the second term thereby making the distance between the attacker and target grow instead of decreasing. This, in a sense, implies that the risk averse attacker, which also wants to stay away from the defender, focuses more on evading from the defender when the defender is close to delay its capture. Otherwise, it keeps chasing the target and would eventually capture the target because $\dot{V}_{AT}$ would be strictly less than 0 in this case.

*2. Regions of Victory for the Defender and the Attacker*

The notion of region of victory ($RoV$) for the defender is defined as the set of all initial conditions of the defender, for a given initial condition of the attacker, such that the defender captures the attacker before the attacker could reach the target. The $RoV$ for the defender can be defined as in Eq. (29)

$$RoV_D\left(r_{A_i0}\right) = \{r_{D0} \in \mathbb{R}^2 | \; \|r_{A_i}(t_f) - r_D(t_f)\| \leq R_c \; \& \; \|r_A(t_f) - r_T\| > R_c\}, \tag{29}$$

where $t_f$ is the final time called game time, at which the game would terminate in favour of either the attacker or the defender. Similarly, the $RoV$ for the attacker, for a given initial condition of the defender, can be defined as:

$$RoV_A\left(r_{D_j0}\right) = \{r_{A0} \in \mathbb{R}^2 | \; \|r_A(t_f) - r_T\| \leq R_c \; \& \; \left\|r_A(t_f) - r_{D_j}(t_f)\right\| > R_c\}. \tag{30}$$

In the next two subsections, we will find the estimates of the $RoV$'s for the defender and the attacker under the assumption that the attackers do not avoid the defenders and just move straight toward the target with maximum velocity. This estimates then will be used to guarantee capture of the attackers in the multiplayer game. The time information obtained from the analytic solutions discussed in the the next subsection would be used to pair the defenders with the attackers.

### a) Estimate of the Region of Victory for the Defender

If we assume that the attacker has no intention to deviate from the defender and it only wants to capture the target i.e. $R_{AD}$ tends to 0, then the control strategies of the attacker can be simplified to the form:

$$\begin{aligned} \dot{x}_{A_1} &= u_{max_{A_1}} \frac{-(x_{A_1} - x_T)}{\sqrt{(x_{A_1}-x_T)^2 + (y_{A_1}-y_T)^2}}, \\ \dot{y}_{A_1} &= u_{max_{A_1}} \frac{-(y_{A_1} - y_T)}{\sqrt{(x_{A_1}-x_T)^2 + (y_{A_1}-y_T)^2}}. \end{aligned} \tag{31}$$

This results in a pure pursuit problem with attacker moving toward the target in a straight line with defender's dynamics given as:

$$\begin{aligned} \dot{x}_{D_1} &= u_{max_{D_1}} \frac{(x_A - x_{D_1})}{\sqrt{(x_A-x_{D_1})^2 + (y_A-y_{D_1})^2}}, \\ \dot{y}_{D_1} &= u_{max_{D_1}} \frac{(y_A - y_{D_1})}{\sqrt{(x_A-x_{D_1})^2 + (y_A-y_{D_1})^2}}. \end{aligned} \tag{32}$$

With this control strategy, the time that the attacker would require to reach the target from the given initial position is given as:

$$t_{AT} = \frac{\sqrt{(x_{A_10}-x_T)^2 + (y_{A_10}-y_T)^2} - R_c}{u_{max_{A_1}}}, \tag{33}$$

which is the minimum time that attacker would require to reach the target. Similarly, after solving the pure-pursuit problem,[21] we can find the time of interception for the given initial position of the defender and the attacker using Eq. (34):

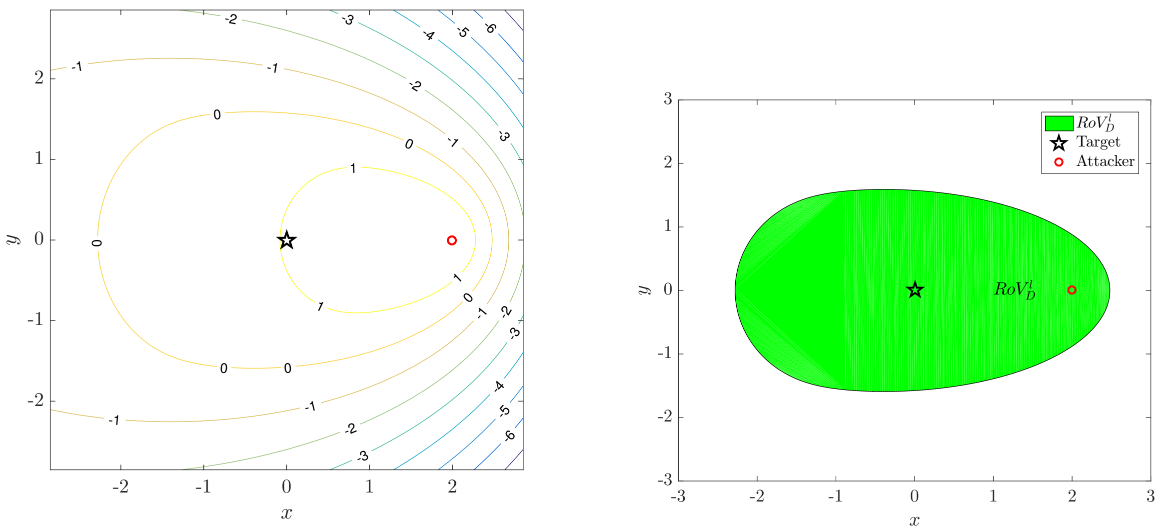
$$\begin{aligned} t_{DA} &= \frac{\left(\gamma_{11} + cos(\beta_0^{11})\right)\left\|r_{D_1A_1}(0)\right\| - \left(\gamma_{11} + cos(\beta^{11}(t_{DA}))\right)R_c}{u_{max_{A_1}}\left(\gamma_{11}^2 - 1\right)}, \\ \beta_0^{11} &= cos^{-1}\left(\frac{\left(r_{D_1A_1}(0)\right)^T\left(r_{A_1T}(0)\right)}{\left\|\left(r_{D_1A_1}(0)\right)\right\|\left\|\left(r_{A_1T}(0)\right)\right\|}\right), \\ \frac{sin\left(\beta^{11}(t_{DA})\right)}{tan^{\gamma}\left(\frac{\beta^{11}(t_{DA})}{2}\right)} &= \frac{\left\|r_{D_1A_1}(0)\right\| sin(\beta_0^{11})}{R_c tan^{\gamma}(\frac{\beta_0^{11}}{2})}, \end{aligned} \tag{34}$$

here the speed ratio of the defender and attacker ($\gamma_{11} = \frac{u_{max_{D_1}}}{u_{max_{A_1}}}$) is greater than 1. With this time information, we know that the defender would win the game if $t_{DA} < t_{AT}$. We can partition the state space into two regions: RoV of defender and RoV of the attacker, by evaluating the contours of the function in Eq. (35):

$$f_{RoV} = t_{AT} - t_{DA}. \tag{35}$$

The zero level contour would be the partitioning contour between the two RoV's. These contours can be obtained by numerically evaluating $f_{RoV}$. To find the contours of $f_{RoV}$, we create a grid around the initial location of the attacker and the target. The initial location of the defender is chosen on the grid points and $f_{RoV}$ is evaluated for all such initial locations of the defender on the grid. Then contour function in MATLAB is used to find contours of $f_{RoV}$ over the grid corresponding to different values. These contours are shown in Fig. 2 (a) with the attacker initially at $x_{A_1 0} = [2,\ 0]^T$ and for $u_{max_{D_1}} = 1.2$ and $u_{max_{A_1}} = 1$ i.e. $\gamma_{11} = 1.2$. In this figure, the positive contours correspond to the initial conditions of the defender for which the defender would win as $t_{DA} < t_{AT}$ and negative contours correspond to the attacker's victory as $t_{AT} < t_{DA}$. The region of victory for this initial location of the attacker is explicitly shown in Fig. 2 (b).

This estimate of the region of victory $RoV_D^l$ is always smaller than the actual one because, if the attacker also tried to avoid the defender using the control law given in Eq. (23), then that would result in attacker moving away from the defender when the defender is close. This would deviate the attacker from the straight line path toward the target delaying the capture of the target by the attacker. The defender would have more time to capture the attacker which means that the defender could start a little farther than the boundary of $RoV_D^l$ and still capture the attacker. In a sense, it is a worst case $RoV$ for the defender. This is evident from Fig. 3 which shows the $RoV_D$ when attacker uses control action as given in Eq. (23) with $R_{AD} = 2$ along with $RoV_D^l$. This $RoV_D$ is calculated by numerically simulating the system for various initial conditions of the defender on the grid around the given initial position of the attacker and the target.




(a) Contours of $f_{RoV}$

(b) Region of victory

Figure 2: Region of victory of the defender, for the given position of the attacker

### b) Estimate of the Region of Victory for the Attacker

Using the similar idea as discussed above, we can find an estimate, $RoV_A^u$, of the $RoV_A$. In this case, the initial position of the defender is fixed and the initial position of the attacker is chosen on the grid points. Figure 4 shows the contours of the function in Eq. 35 for the defender initially located at $x_{D_1 0} = [3,\ 0]^T$. If the attacker starts inside the region bounded by zero level contour, then the attacker would win if it does not avoid the defender as $t_{AT} < t_{DA}$ in that region. However, if the attacker starts outside the zero level contour then the defender would win as $t_{DA} < t_{AT}$.

For the attacker, this $RoV_A^u$ is in fact an over estimate of the actual $RoV_A$ because this corresponds to a straight line motion of the attacker toward the target at maximum speed which requires minimum time to capture the target. If the attacker also avoids the defender to delay its capture by the defender when the defender is nearby, then attacker would require more time to capture the target. In this case, capture of the target can not always be guaranteed if the attacker starts at the boundary of $RoV_A^u$ i.e. it may have to start well inside $RoV_A^u$. This is evident from Fig. 5 which shows the $RoV_A$ when the attacker uses control law as in Eq. (23) with $R_{AD} = 2$ along with the estimate $RoV_A^u$. The $RoV_A$ is smaller than the estimate $RoV_A^u$ corresponding to the straight line motion of the attacker.

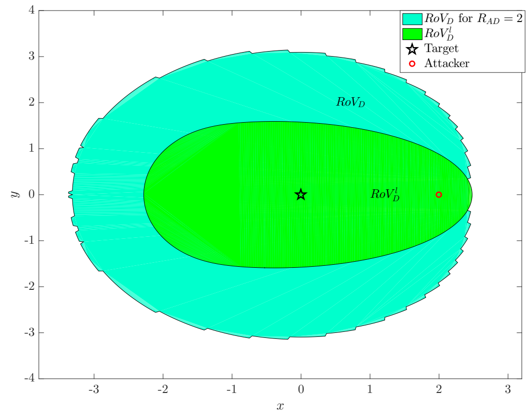
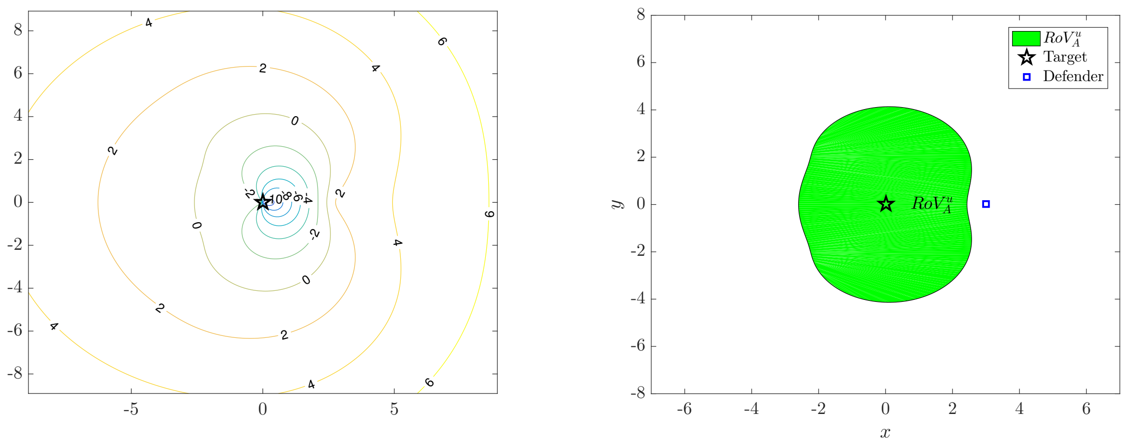



Figure 3: Regions of victory for the defender, for different control strategies of the attacker




(a) Contours of $f_{RoV}$

(b) Region of victory

Figure 4: Region of victory of the attacker

**Proposition 1.** *If an attacker starts outside the* $RoV_A^u(r_{D0})$ *and uses the control law given by Eq.* (23)*, then the attacker is guaranteed to be captured by the defender starting at* $r_{D0}$*.*

*Proof.* The proof follows directly from the definition of the $RoV_A^u$ and it being an over estimate of the $RoV_A$ and the fact that the defender is faster than the attacker. □

**Proposition 2.** *The proposition 1 holds true even if the attacker uses collaborative control strategy as given in Eq.* (15)*.*

*Proof.* This again is because the fastest any attacker could reach the target is by moving straight toward the target at maximum speed. This means that any other control strategy would require attacker more time to capture the target. Since the defender can capture the attacker starting outside $RoV_A^u$ with straight motion toward the target, it also has enough time to capture the attacker with any other control strategy. □

### C. Single Defender vs Two Attackers Game (1-2 TAD game)

In this section, we investigate the advantage of allowing one defender to capture two attackers by applying pure pursuit corresponding to the attackers sequentially. Under the same assumption as in Subsection B.2.a) i.e. the attackers are not trying to avoid defenders, consider two defenders and two attackers at $r_{D_10}$, $r_{D_20}$

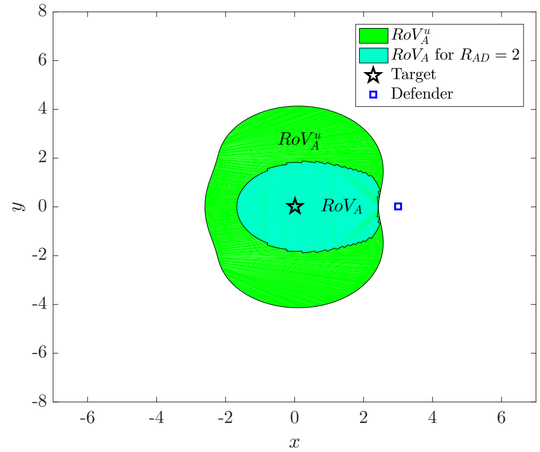



Figure 5: Regions of victory for the attacker, for different control strategies of the attacker

and $r_{A_10}$, $r_{A_20}$, respectively. Assume that $A_1$ can be captured by $D_1$ using pure pursuit control in $t_{11}$ s and $A_2$ by $D_2$ in $t_{22}$ s. Also assume that after $D_1$ captures $A_1$, it can capture $A_2$ in $\bar{t}_{12}$ s by applying pure pursuit control corresponding to $A_2$. Then for given positions of $D_1, D_2$ and $A_1$, all the initial positions of $A_2$ satisfying the condition in Eq. 36 yield $t_{22}+t_{11} > \bar{t}_{12}+t_{11}$ i.e. total active time when both the defenders go after the attackers individually is greater than the time when only one defender goes after both the attackers in sequence.

$$\frac{\left(\gamma_{22}+cos(\beta_0^{22})\right)\|r_{D_2A_2}(0)\|-\left(\gamma_{22}+cos(\beta^{22}(t_{22}))\right)R_c}{u_{max_{A_2}}\left(\gamma_{22}^2-1\right)} > \frac{\left(\gamma_{12}+cos(\beta^{12}(t_{11}))\right)\|r_{D_1A_2}(t_{11})\|}{u_{max_{A_2}}\left(\gamma_{12}^2-1\right)} - \frac{\left(\gamma_{12}+cos(\beta^{12}(\bar{t}_{12}+t_{11}))\right)R_c}{u_{max_{A_2}}\left(\gamma_{12}^2-1\right)}, \tag{36}$$

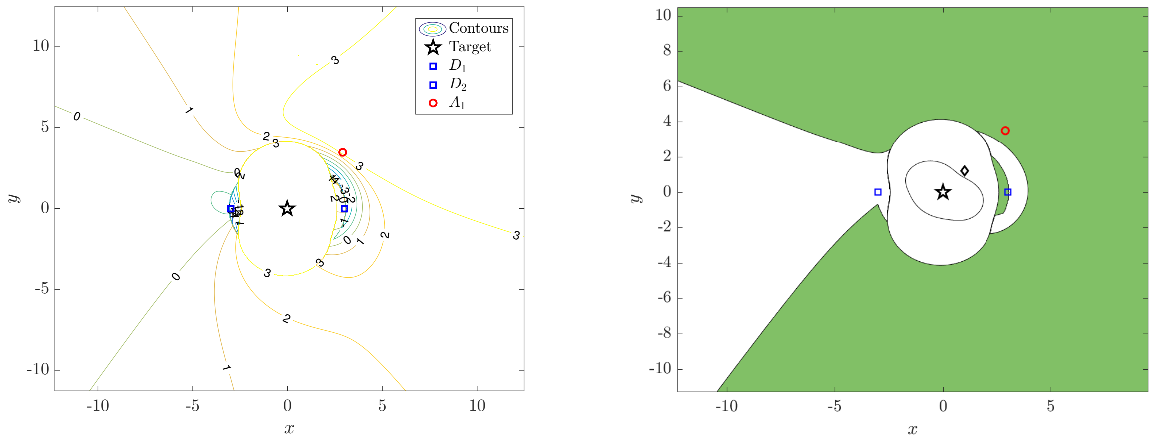
where $\|r_{D_1A_2}(t_{11})\|$ and $\beta^{12}(t_{11})$ is the distance between $D_1$ and $A_2$ and angle between the vectors joining $D_1$ to $A_2$ and $A_2$ to target at the instance when $A_1$ is captured by $D_1$. This values can be obtained by similar formulae as in Eq. (34) by assuming straight line motion of the attackers.

This region is shown in Fig. 6 (dark green region) which is obtained using the similar idea of contours described in Subsection B.2.a). For the given positions of $D_1$, $D_2$ and $A_1$, if $A_2$ starts in the green region then the total time required by $D_1$ to capture $A_1$ and $A_2$ using pure pursuit applied sequentially is smaller than the total time required when $D_1$ applies pure pursuit corresponding to $A_1$ and $D_2$ applies the same corresponding to $A_2$. This means that the total active time, which also reflects the energy consumption, of the defenders' team can be minimized by assigning two attackers to a single defender and allowing other defender to stay inactive when initial configurations of the two attackers lie in a particular set satisfying Eq. 36. This 1→2 pairing idea can be easily generalized to a game with more than 2 defenders and attackers.

## D. Multiplayer Game with Modified Pairings

In this section, we use the knowledge of interception time of the attackers in 1-1 and 1-2 TAD games, under the above stated assumption, to collaboratively pair defenders with attackers in multiplayer game. Researchers[6,7] have used 1 →1 pairings for which they used Hungarian matching algorithm to make pairings to maximize the number of captured attackers in multiplayer game. We propose a modified pairing strategy to pair defenders with attackers in which the defenders' team also considers the possible 1→2 pairings of the defenders to the attackers in addition to 1→1 pairings. The pairing strategy makes pairings which minimize the overall nominal time of interception. By nominal time, we mean the time at which the attacker would be captured given that the attacker does not avoid the defenders. Let us define the total nominal time of interception of the attackers as:

$$J = \sum_{i=1}^{N_{pi}} t_i^n, \tag{37}$$




(a) Contours of $(t_{22} - \bar{t}_{12})$

(b) Region

Figure 6: Region of better performance for 1 vs 2 pairing

where $N_{pi}$ is the total number of attackers that can be captured and $t_i^n$ is the nominal time of interception of the $i^{th}$ attacker. When the $i_1^{th}$ and the $i_2^{th}$ attackers are assigned to the $j^{th}$ defender then $t_{i_1}^n + t_{i_2}^n = t_{ji_1} + \bar{t}_{ji_2}$. The goal of the defenders' team is to make pairings of the defenders with the attackers to maximize $N_{pi}$ and to minimize $J$.

To check for the best 1→1 pairings, Hungarian algorithm (Munkres algorithm), as in,[6,7] is used. To compare 1→1 and 1→2 pairings, Algorithm 1 (given as a pseudo-code) is proposed. This algorithm finds 1→1 pairings using Hungarian algorithm and 1→2 pairings by doing exhaustive search over all the possible 1→2 pairings if $N_A$ is even and less than $2N_D$. Then it chooses the pairings that maximize $N_{pi}$ and if $N_{pi}$ is same for both the pairings then chooses the ones which minimize $J$.

When $N_A$ is odd and if we assign one of the attackers to one of the defenders and then search for 1→2 pairings for the rest of the players, then the worst case complexity, when $N_D = \frac{N_A+1}{2}$, is $O(N_A^6)$ which is much higher than $O(N_A^4)$ when $N_A$ is even for larger $N_A$. So we use only 1→1 pairings when $N_A$ is odd. Also, when $N_A$ is greater than $2N_D$, not all of the attackers would be paired so it is better to avoid huge computations in that case as well.

**Algorithm 1** Pseudo Code for Pairing Defenders to Attackers

**Inputs**: Initial positions and numbers of the defenders and the attackers: $(r_{D0}, r_{A0}, N_D, N_A)$
**Outputs**: The assignments, total nominal time of interception, and $N_{pi}$: (assign, costT, $N_{pi}$)

**procedure** ASSIGN D TO A$(r_D, r_A, N_D, N_A)$
  **if** $N_A$ is even AND less than $2N_D$ **then**
    Find T$[j, i]$= timeIntercept$(r_{D_j0}, r_{A_i0})$, $\forall j \in I_d$ and $\forall i \in I_A$
    Find AttPairs = set of all possible pairs of the attackers
    Set Def =[1, 2, ..., $N_D$]
    Find T2$[i, j]$= timeIntercept2$(r_{D_j0}, [r_{A_{i1}0}, r_{A_{i2}0}])$ (Where $ik$=AttPairs$(i, k)$ $\forall k = \{1, 2\}$)
    [assign1, costT1, $N_{pi}^1$]=HungarianMatch(T, $N_A$, $N_D$,)
    [assign2, costT2, $N_{pi}^2$]=Pair2(T2, AttPairs, Def, $N_A$, $N_D$)
    **if** $N_{pi}^1 == N_{pi}^2$ **then**
      **if** costT1 $<$ costT2 **then**
        assign=assign1; costT=costT1; $N_{pi}$=$N_{pi}^1$;
      **else**
        assign=assign2; costT=costT2; $N_{pi}$=$N_{pi}^2$;
      **end if**
    **else if** $N_{pi}^1 > N_{pi}^2$ **then**
      assign=assign1; costT=costT1; $N_{pi}$=$N_{pi}^1$;

**else**
assign=assign2; costT=costT2; $N_{pi}$=$N_{pi}^2$;
**end if**
**return** assign, costT, $N_{pi}$.
**else**
Find T[$j,i$]= timeIntercept($r_{D_{j0}}$, $r_{A_{i0}}$)
[assign, costT, $N_{pi}$]=HungarianMatch(T)
**end if**
**return** assign, costT, $N_{pi}$.
**end procedure**

---

**Remark 1.** *'timeIntercept' gives time of interception for the given initial positions of the defender and the attacker. It yields 'Inf' if no interception is possible. 'timeIntercept2' gives total time of interception for the given defender intercepting the two attackers in that order. It yields 'Inf" if either of the interceptions is not possible.*

**Remark 2.** *The function 'HungarianMatch' is a standard routine which uses Hungarian matching algorithm. The function 'PAIR2' finds 1→2 pairings and is described in Appendix as a Pseudo-code.*

**Remark 3.** *The same approach could be extended to 1 to n (1→n) pairing but it rapidly increases the computational complexity of the pairing algorithm. In fact, the 1→n pairing assignment problem is NP hard in general.*

#### *1. Re-assignment Protocol*

The pairing assignment is done at the beginning of the game but all the attackers may not be paired with the defenders depending upon their number and initial locations. When there are still some unpaired attackers, the assignment is performed again when one of the already paired attackers is captured. In addition to this, if there is 1→2 pairing, assignment is performed again every time the smallest of the nominal times of interception has passed but no interception occurred. This is to improve performance in case one defender is paired with two attackers and the attackers also try to avoid the defenders.

### E. Guarantees on the Capture of the Attackers

**Theorem 1.** *Let us assume that $N_D$ defenders are deployed around the target. If $N_A$ attackers are initially located such that $r_{A_{i0}} \notin RoV_A^u(r_{D_{10}}) \cup RoV_A^u(r_{D_{20}}) \cup ... \cup RoV_A^u(r_{D_{N_D0}}), \quad \forall i \in I_A$, then*

*a) at least $N_D$ of them (if $N_A \geq N_D$)* ***OR***

*b) all of them (if $N_A < N_D$)*

*would be captured by the defenders with a proper (1→1) pairing of the defenders with the attackers, when each defender applies pure pursuit control corresponding to 1-1 TAD game with the paired attacker, even if the attackers avoid the defenders using the controller in Eq.* (15).

*Proof.* We are given that

$$\begin{aligned} & r_{A_{i0}} \notin RoV_A^u(r_{D_{10}}) \cup RoV_A^u(r_{D_{20}}) \cup ... \cup RoV_A^u(r_{D_{N_D0}}), \quad \forall i \in I_A \\ \Longrightarrow \quad & r_{A_{i0}} \notin RoV_A^u(r_{D_{j0}}), \quad \forall i \in I_A, \quad \forall j \in I_D \\ \Longrightarrow \quad & \exists j \in I_D \text{ s.t. } r_{A_{i0}} \notin RoV_A^u(r_{D_{j0}}), \quad \forall i \in I_A \end{aligned}$$

From Proposition 2

$$\Longrightarrow \exists j \in I_D \text{ and } \exists t_f^i > 0 \text{ s.t.} \left\| r_{D_j}(t_f^i) - r_{A_i}(t_f^i) \right\| \leq R_c \text{ when } u_{D_j} = u_{max_{D_j}} \frac{r_{A_i} - r_{D_j}}{\left\| r_{A_i} - r_{D_j} \right\|}, \quad \forall\, i \in I_A$$

In other words, for any attacker there is a defender which can capture it. So if $N_A$ is less than $N_D$, then all of the attackers can be captured by the defenders by pairing $N_A$ defenders with the attackers. In case $N_A$

is greater than $N_D$, only $N_D$ of them could be paired with the $N_D$ defenders and hence at least $N_D$ could be captured.

For illustration purpose, consider 4 defenders situated at $[3,0]^T$, $[0,3]^T$, $[-3,0]^T$, $[0,-3]^T$. Figure 7 shows the $RoV_A^u$'s of the attackers corresponding to this defenders. Given that the attackers lie outside the union of this regions there is at least one defender which is capable of capturing that attacker. Pairing that particular defender with the attacker will guarantee that the attacker is captured. This way, in this particular example, at least 4 of the attackers starting outside the $\cup RoV_A^u$'s (green region) are guaranteed to be captured by the defenders. $\square$

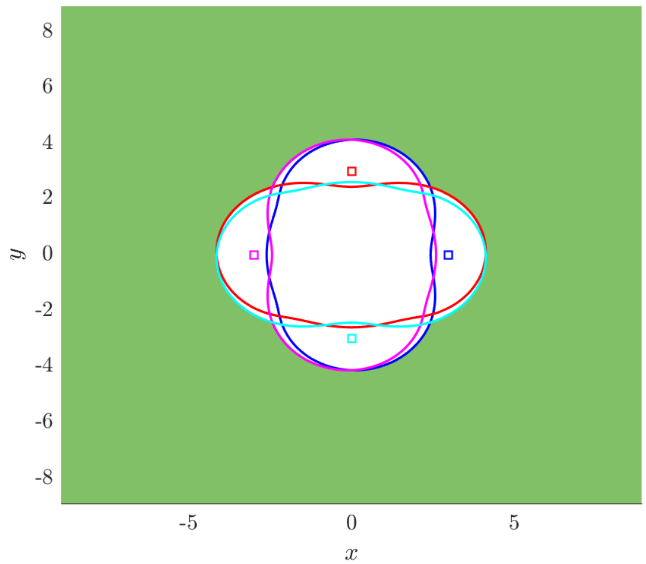



Figure 7: Region of guaranteed victory for the defenders

**Remark 4.** *The capture can also happen in some specific cases where the initial locations of the attackers are inside the union but outside the intersection of this $RoV_A^u$'s but this can not be guaranteed for all such possible cases.*

## IV. Simulations and Results

In this section, we provide simulations to demonstrate the efficacy of the proposed algorithm. All the defenders are assumed to be identical and the same is true for the attackers. The maximum speed of all the defenders is assumed to be $u_{max_{D_j}} = 1.2\ m/s$ and that of the attackers is $u_{max_{A_i}} = 1\ m/s$. Other parameters are: $R_c = 0.1$; $\delta = 5$. Consider $N_D = 4$ defenders situated at $r_{D0} = [(3,0)^T,\ (0,3)^T,\ (-3,0)^T,\ (0,-3)^T]$. For comparison purpose, we also define actual total active time as in Eq. 38, where $t_j^a$ is total active time of the $j^{th}$ defender.

$$J_a = \sum_{j=1}^{N_D} t_j^a \tag{38}$$

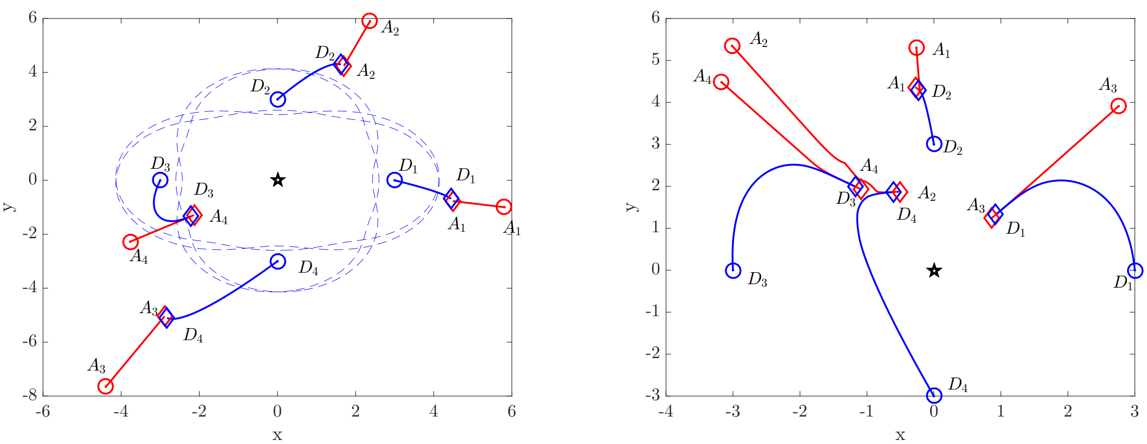
In all the following figures, the blue color corresponds to the defenders and the red color corresponds to the attackers. The circles show the initial locations of the corresponding agents and the diamonds show the final locations of the agents. The target is shown as a solid black star at the origin.

### A. Guaranteed Capture with 1→1 Pairings

1) When the attackers do not avoid the defenders: Figure 8 (a) shows a game with $N_D = 4$ and $N_A = 4$ when the attackers move straight toward the target and do not avoid the defenders. As observed in Fig. 8 (a), since the attackers initially located at $r_{A0} = [(5.81,-0.99)^T,(2.37,5.90)^T,(-4.39,-7.65)^T,(-3.77,-2.28)^T]$

start outside the $\bigcup_{j=1}^{4} RoV_A^u(r_{D_j 0})$ of the defenders (enclosed by blue dotted lines), the defenders are able to capture them with 1→1 pairings.

2) When the attackers avoid the defenders ($R_{AD} = 2$): Even when the attackers use control law given in Eq. (15) with $R_{AD} = 2$ and try to avoid the defenders, the defenders are able to capture the attackers, as can be observed in Fig. 8 (b). The attackers in this case start at $r_{A0} = [(-0.26, 5.30)^T, (-3.01, 5.35)^T, (2.76, 3.91)^T, (-3.17, 4.48)^T]$;




(a) Attackers do not avoid defenders

(b) Attackers avoid defenders

Figure 8: Multiplayer Game with 1→1 pairings

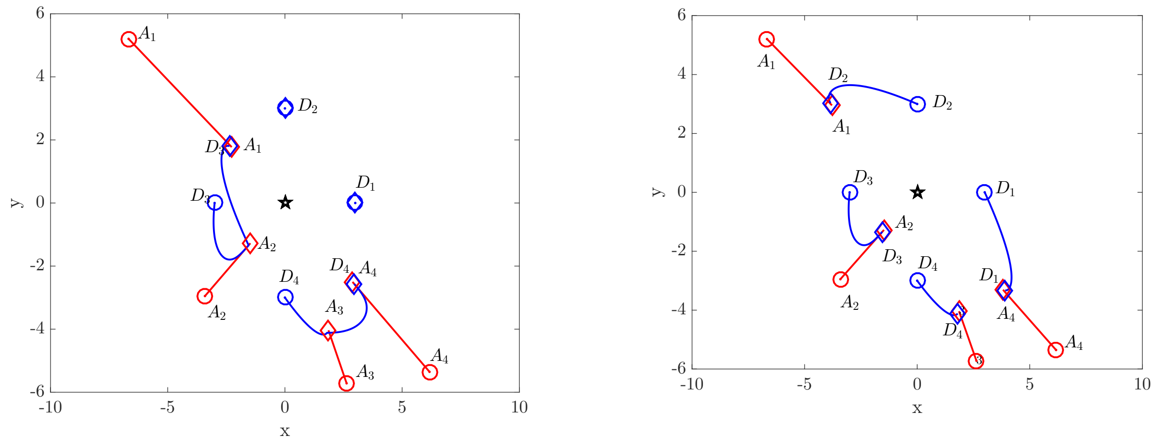
## B. Capture with Modified Pairings

1) When the attackers do not avoid the defenders: Figure 9 shows comparison of 1→1 and 1→2 pairings for the attackers initially located at $r_{A0} = [(-6.67, 5.20)^T, (-3.41, -2.95)^T, (2.63, -5.72)^T, (6.17, -5.35)^T]$. The attackers are captured using both the pairings as observed in Figs. 9 (a) and (b). As shown in Fig. 9 (a), using 1→2 pairings, $D_3$ and $D_4$ chase ($A_1$, $A_2$) and ($A_3$, $A_4$), respectively, while $D_1$ and $D_2$ remain inactive. These 1→2 pairings do better in terms of total active time as they require total active time $J_a = 9.93$ s whereas 1→1 pairings shown in Fig. 9 (b) require $J_a = 11.19$ s.

2) When the attackers avoid the defenders ($R_{AD} = 2$): Similar to the previous case, as shown in Fig. 10 (a) and (b), all the attackers are captured even when they try to avoid the defenders using control action given by Eq. (15) with 1→1 and 1→2 pairings. In Fig. 10 (a), only $D_1$ and $D_4$ are paired with the attackers and other two defenders are inactive. On the other hand, as in Fig. 10 (b), all the defenders are paired with one attacker each. Although in this particular case, the 1→2 pairings ($J_a$=13.29 s) do better in terms of total active time than 1→1 pairings ($J_a$=18.45) but in general this cannot be guaranteed when the attackers avoid the defenders.

## C. When $N_A$ is Greater than $N_D$

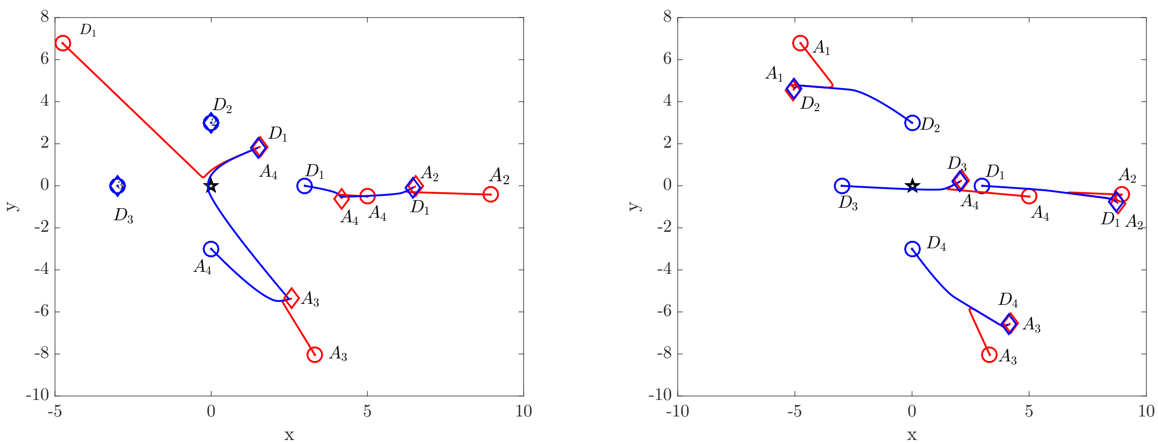
1) When the attackers do not avoid the defenders: As observed from Fig. 11, even when $N_A \in [N_D,\ 2N_D]$, all the attackers are captured by the defenders when the attackers only move straight toward the target and do not avoid the defenders. In Fig. 11 (a), $D_2$, $D_3$ and $D_4$ are assigned two attackers each and $D_1$ is inactive. $D_2$, $D_3$ and $D_4$ capture their paired attackers by sequentially applying pure pursuit control corresponding to them. In Fig. 11 (b), however, all the defenders are assigned one attacker each in the beginning. Once $D_2$ and $D_3$ capture their assigned attackers, each of them is re-assigned a new attacker which they are able to capture. In Fig. 11 (a), the total active time is 14.17 s whereas in Fig. 11 (b) it is 14.83 s, which means modified pairing improves the total active time.




(a) 1 to 2 pairing, J' = 9.93 s

(b) 1 to 1 pairing, J' = 11.19 s

Figure 9: Multiplayer Game with 1→2 vs 1→1 pairings (attackers do not avoid defenders)




(a) 1 to 2 pairing, $J_a$ = 13.29 s

(b) 1 to 1 pairing, $J_a$ = 18.45 s

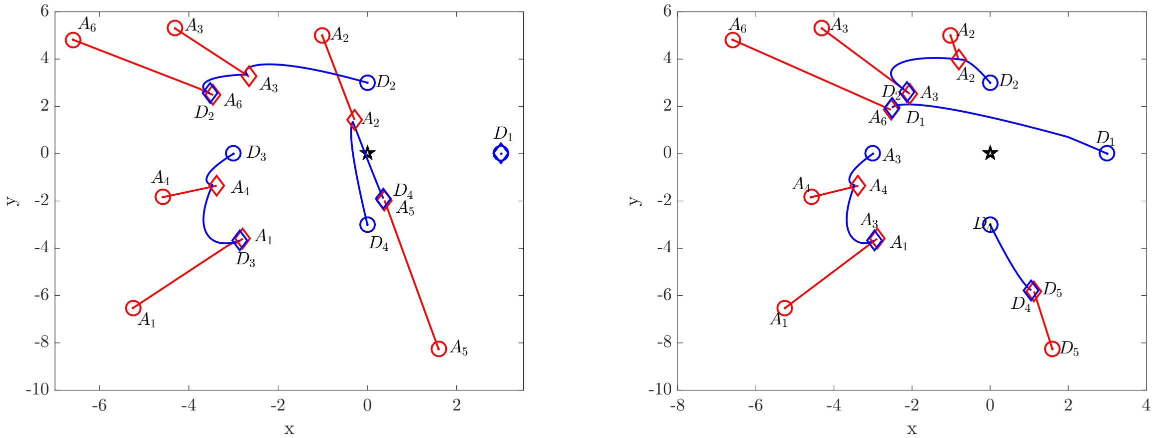
Figure 10: Multiplayer Game with 1→2 vs 1→1 pairings (attackers avoid defenders)

2) When the attackers avoid the defenders ($R_{AD} = 2$): Even if the attackers try to avoid the defenders using control action given in Eq. (15), the capture happens for $N_A \in [N_D,\ 2N_D]$ as shown in Fig. 12. As can be observed from Fig. 12 (a), initially only $D_2$, $D_3$, $D_4$ were assigned the attackers and only they were moving, $D_1$ was inactive. But when the minimum nominal time of interception had passed without any interception because of attackers' avoidance from the defenders, the assignment was performed again in which $A_3$ was assigned to $D_1$ thereby making it active and other pairings were also modified.

Although the capture was guaranteed because of the re-assignment, there was no improvement in the active time because of attackers' avoidance. In general, modified pairing algorithm is beneficial if the attackers do not avoid the defenders but it may or may not be so when the attackers do avoid the defenders.

## V. Conclusions

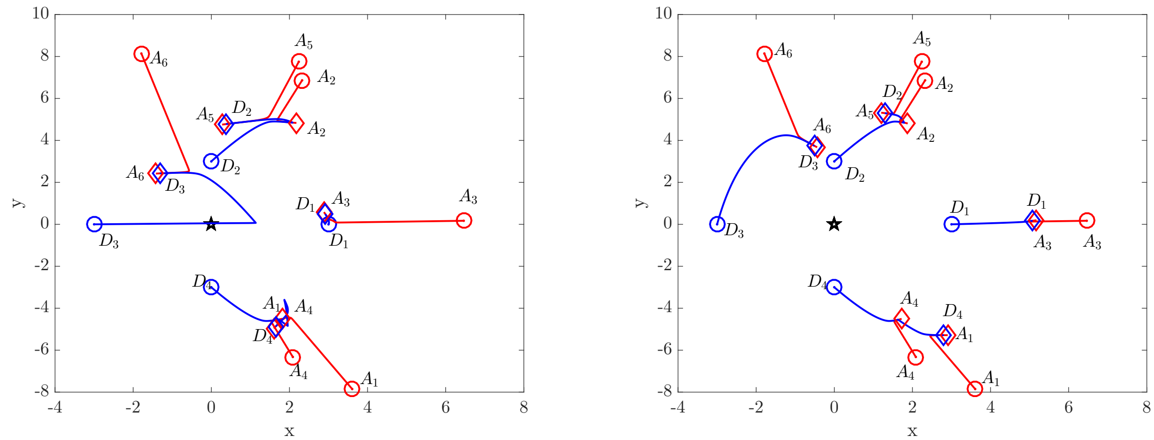
In this paper, an algorithm is presented which combines different conflicting objectives of the opposing players in multiplayer TAD games into a single scalar approximation function which is used to develop control strategies for the players. The strategy for the defender in 1-1 TAD game simplifies to a pure pursuit strategy. Knowledge of the time of interception of the attackers when they do not avoid the defenders is used




(a) 1 to 2 pairing, $J_a$ = 14.17 s

(b) 1 to 1 pairing, $J_a$ = 14.83 s

Figure 11: Multiplayer Game with 1→2 vs 1→1 pairings (attackers do not avoid defenders)




(a) 1 to 2 pairing, $J_a$ = 15.60 s

(b) 1 to 1 pairing, $J_a$ = 12.62 s

Figure 12: Multiplayer Game with 1→2 vs 1→1 pairings (attackers avoid defenders)

to collaboratively pair defenders with attackers to maximize the number of captured attackers. In addition to this, 1-2 TAD games under a sequential pure pursuit strategy are also considered during pairing phase to minimize the active time of the defenders if the same number of attackers can be captured. A modified pairing algorithm is proposed to do the collaborative pairings. Simulations are provided to demonstrate the efficacy of the algorithm and also to demonstrate the improvement in performance with modified pairings. The modified pairing strategy is found to reduce the active time of the defenders when the attackers do not avoid the defenders but it may or may not do so when the attackers avoid the defenders.

In future, we would like to investigate scalability of the proposed algorithm for 1 defenders to n attackers (1→n) pairing, and trade-off between the performance and the computational cost for the 1→n pairing.

# Appendix

**Algorithm 2** Pseudo-code for Defenders to Attackers (1→2) Pairings

**function** PAIR2(T2, AttPairs, Def, $N_A$, $N_D$)
    Create an empty cell array 'assign' of size $(\frac{N_A}{2} - 1)$ to store pairings of size less than $N_A$;

```
    Create an array 'costT' of size (N_A/2 − 1) to store J values corresponding to the pairings in assign;
    N̄_pi=[2, 4, 6, ..., N_A − 2];
    Initialize assign0 and costT0 to store pairings of size N_A and corresponding J value;
    [assign0, costT0, assign, costT]=Pair(T2, AttPairs, Def, N_A, N_D, assign, costT);
    if assign0 is empty then
        assign2 ← Non-empty cell with largest number of pairings from 'assign';
        Corresponding values from 'costT' and 'N̄_pi' are stored in costT2 and N_pi^2;
    else
        assign2=assign0, costT2=costT0, N_pi^2=N_A;
    end if
     return assign2, costT2, N_pi^2;
end function

function Pair(T2, AttPairs, Def, N_A, N_D, assign, costT)
    if N_A > 2 then
        costT0=Inf;
        for i = 1 to N_A(N_A − 1) do
            for j = 1 to N_D do
                Remove all the attacker pairs containing the attackers in the i^th pair → AttPairs_new;
                Remove the j^th defender and corresponding entries from T2 → Def_new, T2_new;
                [assign00, costT00, assign, costT]=Pair(T2_new, AttPairs_new, Def_new, N_A − 2, N_D − 1,
assign, costT);
                if costT00+T2[i,j] < costT0 then
                    costT0=costT00+T2[i,j];
                    assign0 ← ((j, AttPairs(i)), assign00);
                end if
                if costT00 < costT[N_A/2-1] then
                    costT[N_A/2-1] = costT00;
                    assign{N_A/2 − 1} = assign00;
                end if
            end for
        end for
    else
        assign0 ← ordered pair of attackers and defender with minimum time of interception;
        costT0 ← corresponding minimum time;
    end if
     return assign0, costT0, assign, costT;
end function
```

## References


[1] Isaacs, R., *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*, Courier Corporation, 1999.

[2] Huang, H., Zhang, W., Ding, J., Stipanović, D. M., and Tomlin, C. J., "Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers," *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, IEEE, 2011, pp. 4835–4840.

[3] Stipanović, D. M., Melikyan, A., and Hovakimyan, N., "Guaranteed strategies for nonlinear multi-player pursuit-evasion games," *International Game Theory Review*, Vol. 12, No. 01, 2010, pp. 1–17.

[4] Huang, H., Ding, J., Zhang, W., and Tomlin, C. J., "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 1451–1456.

[5] Chen, M., Zhou, Z., and Tomlin, C. J., "Multiplayer reach-avoid games via low dimensional solutions and maximum matching," *American Control Conference (ACC), 2014*, IEEE, 2014, pp. 1444–1449.

[6] Chen, M., Zhou, Z., and Tomlin, C. J., "Multiplayer reach-avoid games via pairwise outcomes," *IEEE Transactions on Automatic Control*, Vol. 62, No. 3, 2017, pp. 1451–1457.

[7] Coon, M. and Panagou, D., "Control strategies for multiplayer target-attacker-defender differential games with double integrator dynamics," *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, IEEE, 2017, pp. 1496–1502.

[8]Stipanović, D. M., Tomlin, C. J., and Leitmann, G., "Monotone approximations of minimum and maximum functions and multi-objective problems," *Applied Mathematics & Optimization*, Vol. 66, No. 3, 2012, pp. 455–473.

[9]Huang, H., Ding, J., Zhang, W., and Tomlin, C. J., "Automation-assisted capture-the-flag: A differential game approach," *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 3, 2015, pp. 1014–1028.

[10]Mitchell, I. M., "A toolbox of level set methods," *UBC Department of Computer Science Technical Report TR-2007-11*, 2007.

[11]Mitchell, I. M., Bayen, A. M., and Tomlin, C. J., "A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, Vol. 50, No. 7, 2005, pp. 947–957.

[12]Osher, S., Fedkiw, R., and Piechor, K., "Level set methods and dynamic implicit surfaces," *Applied Mechanics Reviews*, Vol. 57, 2004, pp. B15.

[13]Chen, M., Zhou, Z., and Tomlin, C. J., "A path defense approach to the multiplayer reach-avoid game," *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, IEEE, 2014, pp. 2420–2426.

[14]Garcia, E., Casbeer, D. W., Pham, K., and Pachter, M., "Cooperative aircraft defense from an attacking missile," *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, IEEE, 2014, pp. 2926–2931.

[15]Garcia, E., Casbeer, D., Pham, K. D., and Pachter, M., "Cooperative aircraft defense from an attacking missile using proportional navigation," *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 0337.

[16]Pachter, M., Garcia, E., and Casbeer, D. W., "Differential Game of Guarding a Target," *Journal of Guidance, Control, and Dynamics*, 2017, pp. 1–8.

[17]Fuchs, Z. E., Khargonekar, P. P., and Evers, J., "Cooperative defense within a single-pursuer, two-evader pursuit evasion differential game," *Decision and Control (CDC), 2010 49th IEEE Conference on*, IEEE, 2010, pp. 3091–3097.

[18]Lee, S., Polak, E., and Walrand, J., "On the use of min-max algorithms in receding horizon control laws for harbor defense," *Engineering Optimization*, Vol. 2014, 2014, pp. 211.

[19]Huang, H., Zhou, Z., Zhang, W., Ding, J., Stipanovic, D. M., and Tomlin, C. J., "Safe-reachable area cooperative pursuit," *IEEE Transactions on Robotics*, 2012.

[20]Pierson, A., Wang, Z., and Schwager, M., "Intercepting rogue robots: An algorithm for capturing multiple evaders with multiple pursuers," *IEEE Robotics and Automation Letters*, Vol. 2, No. 2, 2017, pp. 530–537.

[21]Shneydor, N. A., *Missile guidance and pursuit: kinematics, dynamics and control*, Elsevier, 1998.