

# Multi-Agent Reach-Avoid Games: Two Attackers Versus One Defender and Mixed Integer Programming

Hanyang Hu\*, Minh Bui\* and Mo Chen

**Abstract**—We propose a hybrid approach that combines Hamilton-Jacobi (HJ) reachability and mixed-integer optimization for solving a reach-avoid game with multiple attackers and defenders. The reach-avoid game is an important problem with potential applications in air traffic control and multi-agent motion planning; however, solving this game for many attackers and defenders is intractable due to the adversarial nature of the agents and the high problem dimensionality. In this paper, we first propose an HJ reachability-based method for solving the reach-avoid game in which 2 attackers are playing against 1 defender; we derive the numerically convergent optimal winning sets for the two sides in environments with obstacles. Utilizing this result and previous results for the 1 vs. 1 game, we further propose solving the general multi-agent reach-avoid game by determining the defender assignments that can maximize the number of attackers captured via a Mixed Integer Program (MIP). Our method generalizes previous state-of-the-art results and is especially useful when there are fewer defenders than attackers. We validate our theoretical results in numerical simulations.

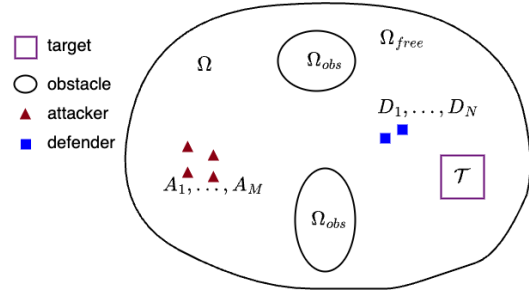
## I. INTRODUCTION

The multi-agent reach-avoid game is a differential game between two adversarial teams. One team, called the attackers, aims to reach a target set of states without being captured by the opposing team while also avoiding obstacles. The opposing team, called the defenders, aims to stop as many attackers as it can from reaching the target while avoiding obstacles. The multi-agent reach-avoid game is an essential research tool, which has a wide range of applications in the area of drone control and robot logistics [1]–[4]

Because of the competitive nature of the game and the high-dimensional joint state space, it is challenging to solve multi-agent reach-avoid games optimally. Some prior works have solved the game under various assumptions. In [5], one team in the adversarial game is simplified to using simple motions, and the other team is assumed to be controlled under the trajectory primitive decomposition approach. In [6], only one of the pursuers could be allowed to pursue one evader in each time interval with the integral constraints on control functions. In [7], both attackers and defenders are assumed to have complete knowledge of other players within a designed scope, and the attacker-defender pairs are decided based on the intersection of the isochrones. These mentioned methods work well in their respective scenarios under their assumptions. However, they sometimes cannot be generalized easily under a less restrictive set of assumptions.

\*H. Hu and M. Bui contributed equally to this work. All authors are with the School of Computing Science, Simon Fraser University. {hha160, buiminhb, mochen}@sfu.ca.

This work received support from the SFU-Huawei Joint Lab.



**Fig. 1:** In a multi-agent reach-avoid game, the attackers (red triangle) attempt to reach the target while avoiding the obstacles or being captured by the defenders along the way. The defenders (blue square) will adversarially design a strategy to stop as many attackers from reaching the target.

Some researchers try to find analytical solutions to the multi-agent reach-avoid game based on geometric features. In [8], the authors introduce a geometric method that considers the geometric relation between the target region and the reaching region in a 2 vs. 1 reach-avoid game. In [9], [10], Rui et al propose different analytical solutions based on disparate geometric features of a 2 vs. 1 game. In [11], the authors propose three different control strategies based on three different stages between the single attacker and the single defender. These methods could obtain suboptimal or optimal analytical solutions through case-by-case discussions. However, these methods depend on the geometry of the games and can have difficulty generalizing to different environments.

Hamilton-Jacobi (HJ) reachability analysis is a powerful method that can provide general optimal strategies for both attackers and defenders in a multi-agent reach-avoid game [12], [13]. The game result only depends on the initial joint states of all attackers and defenders, and optimal joint state feedback strategies can be derived for both teams. However, HJ reachability analysis scales poorly with the problem dimensionality, restricting the previous application of the method to only the 1 vs. 1 game, a 4-dimensional problem. By leveraging the recently developed OptimizedDP library [14], computations up to 7 dimensions become possible, allowing the numerically convergent optimal solution to be computed for games involving 3 agents (with single integrator dynamics, a common assumption in the reach-avoid game literature). However, despite these advances, when the dimensionality of the joint state of all players is greater than 6, the multi-agent reach-avoid game again becomes computationally intractable.

Methods that use high-level logic to scale up low-dimensional solutions of HJ reachability have been effective in obtaining approximate solutions to high-dimensional problems. In [15]–[18], the authors propose using the maximum matching method to extend the result of a 1 vs. 1 reach-avoid game to a multi-agent case through a maximum matching approach. In [19], a MIP is proposed to decide the assignment logic between pairs of vehicles when there are more than two vehicles to guarantee safety. These methods are based on the result of a 1 vs. 1 pursuit-evasion game and could be improved by combining more tractable HJ reachability results.

In this paper, we propose to combine recent computational advances of high-dimensional HJ reachability analysis using OptimizedDP toolbox [14] with a new MIP that provides high-level logic to solve and scale up the solutions of multi-agent reach-avoid games. Specifically, our contributions are as follows. First, we propose a formulation based on HJ reachability for optimally solving a reach-avoid game involving 2 attackers playing against 1 defender. Our solution involves computing a reach-avoid set in a 6-dimensional state space, enabled by OptimizedDP [14] as well as the reach-avoid set for a 1 vs. 1 game. Given the 1 vs. 1 and 2 vs. 1 game solutions, we propose a MIP that maximizes the number of attackers that are guaranteed to be captured by the defenders. The MIP allows a single defender to be assigned to defend against up to 2 attackers. We demonstrate and validate our theoretical contributions in numerical simulations of various game configurations.

The rest of this paper is organized as follows. In Section II, the multi-agent reach-avoid game is formulated along with the player dynamics. In Section III, we introduce the HJ reachability analysis method for solving reach-avoid games and an optimization program based on a MIP that maximizes the number of captured attackers. In Section IV, we show the performance of our proposed method in simulation with different configurations. In Section V, we draw conclusions and discuss possible future work.

## II. PROBLEM FORMULATION

In the multi-agent reach-avoid game, we consider  $M$  attackers,  $\{A_i\}_{i=1}^M = \{A_1, A_2, \dots, A_M\}$  and  $N$  defenders,  $\{D_j\}_{j=1}^N = \{D_1, D_2, \dots, D_N\}$ . Both attackers and defenders are confined to some compact domain  $\Omega \subset \mathbb{R}^2$ . The domain  $\Omega$  is divided into two subsets:  $\Omega_{free}$  and  $\Omega_{obs}$ . Each player can move freely in  $\Omega_{free}$  but should not enter  $\Omega_{obs}$ . We assume the dynamics of attackers and defenders are as follows:

$$\begin{aligned} \dot{\mathbf{x}}_{A_i}(t) &= v_A \mathbf{a}_i(t), & \mathbf{x}_{A_i}(0) &= \mathbf{x}_{A_i}^0, & i &= 1, 2, \dots, M \\ \dot{\mathbf{x}}_{D_j}(t) &= v_D \mathbf{d}_j(t), & \mathbf{x}_{D_j}(0) &= \mathbf{x}_{D_j}^0, & j &= 1, 2, \dots, N \end{aligned} \quad (1)$$

where  $\mathbf{x}_{A_i}$  and  $\mathbf{x}_{D_j}$  are the 2-dimensional states, representing positions of attackers and defenders respectively,  $\mathbf{x}_{A_i}(0)$  and  $\mathbf{x}_{D_j}(0)$  are the initial positions of attackers and defenders respectively,  $v_A$  and  $v_D$  are the maximum speed of attackers and defenders respectively,  $\mathbf{a}_i(t)$  and  $\mathbf{d}_j(t)$  are

the control inputs of attacker  $i$  and defender  $j$  respectively at the time  $t$ .

In the reach-avoid game, the attackers aim to reach some target  $\mathcal{T} \subset \Omega_{free}$ , which is a compact subset of the free region, while avoiding obstacles and being captured by defenders. On the other hand, the defenders will try to prevent the attackers from reaching the target by capturing them or delaying them indefinitely. An attacker  $A_i$  is considered to be captured by the defender  $D_j$  when their relative distance is within a capture distance  $R_C$ . With that, we define the capture sets  $\mathcal{C}_{ij} \subset \Omega_{free}$  for each attacker-defender pair  $(A_i, D_j)$ ,  $i = 1, \dots, M, j = 1, \dots, N$  as  $\mathcal{C}_{ij} = \{(\mathbf{x}_{A_i}, \mathbf{x}_{D_j}) \mid \|\mathbf{x}_{A_i} - \mathbf{x}_{D_j}\|_2 \leq R_C\}$ .

In multi-agent reach-avoid game, we are most interested in the number of attackers that the defenders are guaranteed to be able to prevent from reaching the target. An illustration of our multiplayer reach-avoid game is shown in Fig. 1.

## III. METHODOLOGY

HJ reachability analysis is a numerically convergent framework for obtaining optimal solutions to differential games [12], [20]–[22]. We will first introduce the general background of HJ reachability and how it is used to generate the closed-loop feedback control inputs. Then, we describe the details of solving a 1 vs. 1 and a 2 vs. 1 reach-avoid game using HJ reachability. Finally, we will propose a MIP to solve the general  $M$  vs.  $N$  reach-avoid game utilizing the solutions of the 1 vs. 1 and the 2 vs. 1 game.

### A. Hamilton-Jacobi Reachability

In HJ reachability analysis, one models the dynamics of a system using the following set of differential equations:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{d}), \mathbf{x}(0) = \mathbf{x}^0 \quad (2)$$

where  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{u} \in \mathcal{U}$  is the joint control input of all attackers,  $\mathbf{d} \in \mathcal{D}$  is the joint control input of all defenders. The sets  $\mathcal{U}$  and  $\mathcal{D}$  are the sets of measurable control functions for attackers and defenders respectively. In the case of the reach-avoid game we are considering, we have the following:  $n = 2(M + N)$  is the joint states of  $M$  attackers and  $N$  defenders.

In the reach-avoid game, we need to specify the target set  $R$  which is the winning condition of the attackers. Besides, we also need to specify the constraint which we should obey during the backward propagation from the target set  $R$ . The constraint is described by the avoid set  $A$ . The details of these two sets are in Section III-B and Section III-C.

Specifically, given the target set  $R$ , the level set representation of  $R$  is the function  $l(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  such that  $R = \{\mathbf{x} \in \mathbb{R}^n \mid l(\mathbf{x}) \leq 0\}$ . Similarly, the level set representation of the avoid set  $A$  is the function  $g(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  such that  $A = \{\mathbf{x} \in \mathbb{R}^n \mid g(\mathbf{x}) > 0\}$ . Based on these notations, let the value function  $\Phi : \mathbb{R}^n \times [-T, 0] \rightarrow \mathbb{R}$  be the viscosity solution to the Hamilton-Jacobi-Isaacs (HJI) variational

inequality [20]:

$$\max\left\{\min\left\{\frac{\partial\Phi}{\partial t} + H\left(\mathbf{x}, \frac{\partial\Phi}{\partial\mathbf{x}}\right), l(\mathbf{x}) - \Phi(\mathbf{x}, t)\right\}, g(\mathbf{x}) - \Phi(\mathbf{x}, t)\right\} = 0, \quad t \in [-T, 0], \mathbf{x} \in \mathbb{R}^n, \quad (3)$$

where the optimal Hamiltonian  $H$  is calculated as:

$$H(\mathbf{x}, p) = \min_{\mathbf{u} \in \mathbb{U}} \max_{\mathbf{d} \in \mathbb{D}} p^T f(\mathbf{x}, \mathbf{u}, \mathbf{d})$$

where  $p = \frac{\partial\Phi}{\partial\mathbf{x}}$ .

Analytical solutions to Eq. 3 are rarely feasible, and typically solved using numerical methods [23], which depends on the discretion of the state space and dynamic programming iteration; therefore, the computational and space complexity grows exponentially as the number of states increases. Software toolbox such as OptimizedDP [14], utilizes modern computing power to solve Eq. 3 for up to  $n = 7$  with decent grid resolutions.

Once the solution  $\Phi$  has been obtained, the set of initial positions where attackers are guaranteed to win within the time interval  $T$  is as follows [13], [24]:

$$\mathcal{R}\mathcal{A}_T(R, A) := \{\mathbf{x} \in \mathbb{R}^n \mid \Phi(\mathbf{x}, -T) \leq 0\} \quad (4)$$

In reachability literature, such set obtained from Eq. 3 is called reach-avoid tube. Next, the optimal controls for all attackers and defenders can be derived from the solution  $\Phi$  respectively [21]:

$$\begin{aligned} \mathbf{u}^*(\mathbf{x}, t) &= \arg \min_{\mathbf{u} \in \mathbb{U}} \max_{\mathbf{d} \in \mathbb{D}} p(\mathbf{x}, -t)^\top f(\mathbf{x}, \mathbf{u}, \mathbf{d}), t \in [0, T] \\ \mathbf{d}^*(\mathbf{x}, t) &= \arg \max_{\mathbf{d} \in \mathbb{D}} p(\mathbf{x}, -t)^\top f(\mathbf{x}, \mathbf{u}^*, \mathbf{d}), t \in [0, T] \end{aligned} \quad (5)$$

where  $p = \frac{\partial\Phi}{\partial\mathbf{x}}$ . Furthermore, if we take the time interval  $T \rightarrow \infty$ , we will obtain the set of initial positions where attackers are guaranteed to win which is denoted as  $\mathcal{R}\mathcal{A}_\infty(R, A)$ . A more in-depth introduction to HJ reachability can be found in [12], [25].

For practical problems that involve multiple defenders and attackers (where  $n > 7$ ), it is intractable to obtain  $\Phi$  numerically. In the next few sections, we will describe how we solve Eq. 3 for small sub-problems, which are then used to approximate a winning control strategy for the defenders in larger problem instances. In this paper, we will shortly denote 1 vs. 1 to refer to a game where one attacker is against one defender and 2 vs. 1 for a game where two attackers are up against one defender.

### B. The 1 vs. 1 Reach-Avoid Game ( $n = 4$ )

In a 1 vs. 1 reach-avoid game, the attacker wins if the attacker can reach the target set and loses the game if it is captured by the defender. Following the work in [15], the target set  $R^{11}$  and the avoid set  $A^{11}$  for a 1 vs. 1 reach-avoid game are defined as follows:

$$\begin{aligned} R^{11} &= \{\mathbf{x} \in \Omega^2 \mid \mathbf{x}_A \in \mathcal{T} \wedge \|\mathbf{x}_A - \mathbf{x}_D\|_2 > R_C\} \\ &\cup \{\mathbf{x} \in \Omega^2 \mid \mathbf{x}_D \in \Omega_{obs}\} \\ A^{11} &= \{\mathbf{x} \in \Omega^2 \mid \|\mathbf{x}_A - \mathbf{x}_D\|_2 \leq R_C\} \\ &\cup \{\mathbf{x} \in \Omega^2 \mid \mathbf{x}_A \in \Omega_{obs}\} \end{aligned} \quad (6)$$

Given these sets, we can solve Eq. 3 to obtain the 4-dimensional reach-avoid tube  $\mathcal{R}\mathcal{A}_\infty^{11}(R^{11}, A^{11})$  for the 1 vs. 1 game. An attacker wins if and only if the initial joint state  $(\mathbf{x}_A^0, \mathbf{x}_D^0)$  lies within the reach-avoid tube  $\mathcal{R}\mathcal{A}_\infty^{11}(R^{11}, A^{11})$  under the optimal control [15]:

$$\mathbf{a}^*(\mathbf{x}_A, \mathbf{x}_D, t) = -v_A \frac{p_a(\mathbf{x}_A, \mathbf{x}_D, -t)}{\|p_a(\mathbf{x}_A, \mathbf{x}_D, -t)\|_2} \quad (7)$$

where  $p = (p_a, p_d) = \frac{\partial\Phi^{11}}{\partial(\mathbf{x}_A, \mathbf{x}_D)}$  with  $p_a, p_d \in \mathbb{R}^2$  representing components of the gradient corresponding to the states of the attacker and defender, respectively.

If the initial joint state  $(\mathbf{x}_A^0, \mathbf{x}_D^0)$  lies out of the reach-avoid tube  $\mathcal{R}\mathcal{A}_\infty^{11}(R^{11}, A^{11})$ , the defender will win the game with the optimal control [15]:

$$\mathbf{d}^*(\mathbf{x}_A, \mathbf{x}_D, t) = v_D \frac{p_d(\mathbf{x}_A, \mathbf{x}_D, -t)}{\|p_d(\mathbf{x}_A, \mathbf{x}_D, -t)\|_2} \quad (8)$$

### C. The 2 vs. 1 Reach-Avoid Game ( $n = 6$ )

With the recent advances in solving high-dimensional HJ PDE [14], [26], [27], we are not limited to just solving a 1 vs. 1 reach-avoid game (by solving a 4-dimensional HJI PDE) [15]. In this paper, we will utilize the OptimizedDP toolbox [14] to obtain optimal solutions to the 2 vs. 1 game, in which one defender can be playing against up to 2 attackers in a 6-dimensional joint state space. In this 2 vs. 1 game, a defender wins the game if it is guaranteed to capture and/or stop both attackers from reaching the target set.

In the 2 vs. 1 game, the joint state is  $\mathbf{x} = (\mathbf{x}_{A_1}, \mathbf{x}_{A_2}, \mathbf{x}_D)$ . Since we are interested in guaranteeing that the defender can stop both attackers and win the game, we will consider a 2 vs. 1 reach-avoid game that computes the region of states where **at least** one of the attackers is guaranteed to reach the target, the complement of this region is a set of states where none of the attackers can arrive at the target or the defender is guaranteed to win.

First, we define the target set  $R^{21}$  and the avoid set  $A^{21}$  as the following:

$$\begin{aligned} R^{21} &= \{\mathbf{x} \in \Omega^3 \mid \mathbf{x}_{A_1} \in \mathcal{T} \wedge \|\mathbf{x}_{A_1} - \mathbf{x}_D\|_2 > R_C\} \\ &\cup \{\mathbf{x} \in \Omega^3 \mid \mathbf{x}_{A_2} \in \mathcal{T} \wedge \|\mathbf{x}_{A_2} - \mathbf{x}_D\|_2 > R_C\} \\ &\cup \{\mathbf{x} \in \Omega^3 \mid \mathbf{x}_D \in \Omega_{obs}\} \\ A^{21} &= L_1 \cup L_2 \end{aligned} \quad (9)$$

where

$$\begin{aligned} L_1 &= (\{\mathbf{x} \in \Omega^3 \mid \|\mathbf{x}_{A_1} - \mathbf{x}_D\|_2 \leq R_C\} \\ &\cup \{\mathbf{x} \in \Omega^3 \mid \mathbf{x}_{A_1} \in \Omega_{obs}\}) \\ &\cap \{\mathbf{x} \in \Omega^3 \mid (\mathbf{x}_{A_2}, \mathbf{x}_D) \notin \mathcal{R}\mathcal{A}_\infty^{11}\} \\ L_2 &= (\{\mathbf{x} \in \Omega^3 \mid \|\mathbf{x}_{A_2} - \mathbf{x}_D\|_2 \leq R_C\} \\ &\cup \{\mathbf{x} \in \Omega^3 \mid \mathbf{x}_{A_2} \in \Omega_{obs}\}) \\ &\cap \{\mathbf{x} \in \Omega^3 \mid (\mathbf{x}_{A_1}, \mathbf{x}_{A_2}) \notin \mathcal{R}\mathcal{A}_\infty^{11}\} \end{aligned} \quad (10)$$

In the above definition,  $R^{21}$  comprises states where at least one attacker arrives at the target set  $\mathcal{T}$  while not being captured by the defender, or the defender is obstructed by  $\Omega_{obs}$ . The avoid set  $A^{21}$  is the union of  $L_1$  and  $L_2$ .  $L_1$  is a set of states where the attacker  $A_1$  has been captured or hit

obstacles and the attacker  $A_2$  be captured in the future time steps, which happens when  $(\mathbf{x}_{A_2}, \mathbf{x}_D) \notin \mathcal{R}\mathcal{A}_\infty^{11}$ . Similarly,  $L_2$  is a set of states where  $A_2$  has been captured and  $A_1$  is captured in future time steps. The avoid set  $A^{21}$  is then the union of  $L_1$  and  $L_2$ , which cover all the states leading both attackers getting captured. Note that the definition of  $L_1$  and  $L_2$  involve the 4-dimensional reach-avoid tube  $\mathcal{R}\mathcal{A}_\infty^{11}$ , which allows us to circumvent complications involving defining a time-varying avoid set that encodes the defender capturing the attackers not at the same time, but one after another.

By solving the Eq. 3, we can obtain the reach-avoid tube  $\mathcal{R}\mathcal{A}_\infty^{21}$ , which is a 6-dimensional value function that represents states where at least one attacker can reach the target safely without being captured. In this case, if the initial state does not lie in the reach-avoid tube  $\mathbf{x}^0 \notin \mathcal{R}\mathcal{A}_\infty^{21}$ , then the defender is guaranteed to capture both attackers successfully using the optimal controls:

$$\mathbf{d}^*(\mathbf{x}_{A_1}, \mathbf{x}_{A_2}, \mathbf{x}_D, t) = v_D \frac{p_d(\mathbf{x}_{A_1}, \mathbf{x}_{A_2}, \mathbf{x}_D, -t)}{\|p_d(\mathbf{x}_{A_1}, \mathbf{x}_{A_2}, \mathbf{x}_D, -t)\|_2} \quad (11)$$

where  $p = (p_{a1}, p_{a2}, p_d) = \frac{\partial \Phi^{21}}{\partial (\mathbf{x}_{A_1}, \mathbf{x}_{A_2}, \mathbf{x}_D)}$  with  $p_{a1}, p_{a2}, p_d \in \mathbb{R}^2$  representing components of the gradient corresponding to the states of the first attacker, second attacker, and defender, respectively.

#### D. Mixed Integer Programming for Assigning Attackers

HJ reachability analysis is computationally intractable when the total number of attackers and defenders is greater than three. Hence, to utilize the results of the 1 vs. 1 and 2 vs. 1 game via HJ reachability, we propose a MIP that maximizes the number of attackers that can be assigned to a defender, which can be conveniently solved by software library such as the Python-MIP [28]–[30] Mathematically, this optimization is defined as the following:

$$\text{maximize} \quad \sum_{i,j} e_{ij} \quad (12a)$$

$$\text{subject to} \quad e_{ij} \in \{0, 1\} \quad \forall i, j, \quad (12b)$$

$$\sum_i e_{ij} \leq 2, \quad \forall j, \quad (12c)$$

$$\sum_j e_{ij} \leq 1, \quad \forall i, \quad (12d)$$

$$e_{sj} = 0, \quad \forall j, \forall s \in \mathcal{I}_j, \quad (12e)$$

$$e_{kj} + e_{lj} \leq 1, \quad \forall j, \forall (k, l) \in \mathcal{P}_j \quad (12f)$$

In this program, we denote the capture relationship between the attacker  $A_i$  and the defender  $D_j$  as  $e_{ij} \in \{0, 1\}$ . When  $e_{ij} = 1$ , it indicates that the defender  $D_j$  is assigned to capture the attacker  $A_i$  and  $e_{ij} = 0$  otherwise.

We will then construct the constraints for this optimization program. Firstly, the constraint (12c) limits the total number of attackers assigned to defender  $D_j$  to be at most two at any time. This upper bound is limited by the computational capability for HJ reachability analysis. Secondly, the constraint (12d) ensures that one attacker can only be captured by at most one defender to avoid the case where multiple defenders are assigned to only one attacker.

Let  $\mathcal{I}_j$  be the set of all attackers that defeat the defender  $D_j$  in a 1 vs. 1 game when using optimal control. The constraint (12e) makes sure that a defender  $D_j$  cannot be assigned to an attacker  $i$  that it cannot win in a 1 vs. 1 game by directly setting  $e_{ij} = 0$ . Using a pre-computed reach-avoid tube  $\mathcal{R}\mathcal{A}_\infty^{11}$ , the set  $\mathcal{I}_j$  can easily be constructed.

Similarly, using a pre-computed reach-avoid tube  $\mathcal{R}\mathcal{A}_\infty^{21}$ , we can construct the set  $\mathcal{P}_j = \{(k, l) \mid (\mathbf{x}_{A_k}, \mathbf{x}_{A_l}, \mathbf{x}_{D_j}) \in \mathcal{R}\mathcal{A}_\infty^{21}, 0 \leq k, l \leq M, k \neq l\}$  that contains all the **distinct** pairs of attackers that can win against defender  $D_j$  in a 2 vs. 1 game. The constraint (12f) ensures that defender  $D_j$  will not be assigned to defend against 2 attackers that  $D_j$  cannot capture both.

#### E. Iterative Capture Algorithm

The solution of our MIP in Section III-D determines the capturing assignment for each defender based on the current states of all agents. As the states of these agents will change over time, the constraints 12e and 12f of our MIP also need to be updated accordingly. Hence, we set up and solve the MIP iteratively after every time step to take into account these changes. Our iterative capture algorithm can be summarized as shown in the Algorithm 1.

---

#### Algorithm 1 Iterative Capture Algorithm

---

**Require:**  $\mathcal{R}\mathcal{A}_\infty^{11}, \mathcal{R}\mathcal{A}_\infty^{21}$

**Initialization:** states of all attackers and defenders

**while** not all attackers captured or arrive at target **do**

determine the constraint set  $\mathcal{I}_j$  for each defender  $D_j$

determine the constraint set  $\mathcal{P}_j$  for each defender  $D_j$

solve the MIP

apply optimal controls for defenders based on Eq.(8), Eq.(11)

and the MIP's assignment result

update states of all attackers and defenders after  $\Delta t$

**end while**

---

Due to the optimality of the HJ controller, the defender assigned to attackers is guaranteed to capture the attackers using the optimal control inputs regardless of what control the attackers use. Hence, the above capture algorithm has this property: the guaranteed maximum number of attackers that can reach the target never increases; instead, it can decrease depending on the trajectories of the players as the game plays out and new attacker assignments are made.

## IV. NUMERICAL SIMULATIONS

All HJ reach-avoid tubes are computed offline using the OptimizedDP library developed at Simon Fraser University [14]. First, we calculate the 1 vs. 1 and 2 vs. 1 reach-avoid tube  $\mathcal{R}\mathcal{A}_\infty^{11}$  and  $\mathcal{R}\mathcal{A}_\infty^{21}$  respectively. The computation of  $\mathcal{R}\mathcal{A}_\infty^{11}$  is done on a  $45 \times 45 \times 45 \times 45$  4-dimensional grid, which takes a few seconds to compute until convergence and consumes 100 megabytes of RAM. And the set  $\mathcal{R}\mathcal{A}_\infty^{21}$  is computed on a  $30 \times 30 \times 30 \times 30 \times 30 \times 30$  6-dimensional grid, which takes around 1 hour and 30 minutes to compute until convergence, and consumes around 20 gigabytes of RAM on a 16-thread Intel(R) Core(TM) i9-9900K CPU at 3.60GHz.

We verify the correctness of our reachability calculation and the effectiveness of our MIP method in the following simulations. In our experiment, the attackers use a suboptimal control strategy that takes the shortest path to the target  $\mathcal{T}$  while avoiding obstacles. For comparison, the method that uses maximum matching to obtain approximate solutions to the multi-agent reach-avoid game in [15] is chosen as the baseline. In this baseline method, the maximum number of attackers matched in every time step is at most the same as the number of defenders.

To show the advantages of our method, the number of defenders is chosen to be smaller than the number of attackers in the following simulations. In the simulations, the speed of defenders and attackers are set as  $v_D = 1.5$  unit per second and  $v_A = 1.0$  unit per second respectively. They are all confined to a square domain with two obstacles and a target set  $\mathcal{T}$  as shown in Fig. 2. The capture distance  $R_C$  is chosen as 0.1 units.

Once an attacker has been captured, it will be removed from the game and marked by a plus sign in the figures while the defender's past trajectory will be erased. To clearly visualize the trajectory of multiple players throughout the game, we plot the trajectories of defenders every eight-time step and remove the overlapping parts with other players. In each figure, the green edges connecting a defender to an attacker represent the capturing assignment results from our MIP and the maximum matching.

Attackers Captured			
Game config.	2 vs. 1	6 vs. 2	8 vs. 4
Baseline [15]	1	4	6
Our Method	<b>2</b>	<b>6</b>	<b>8</b>

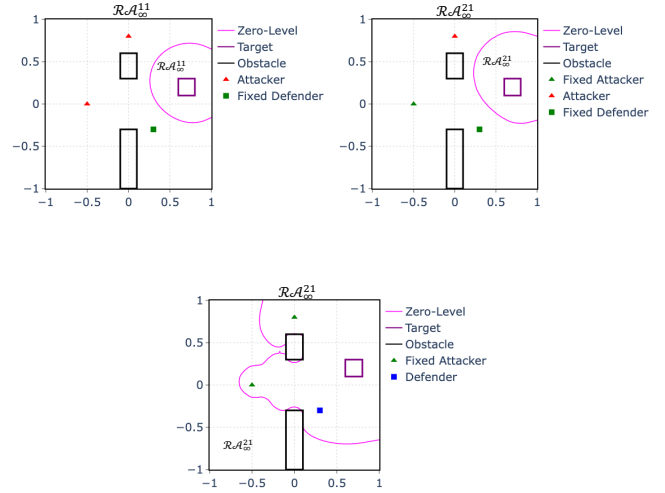
**TABLE I:** Number of attackers captured for different game scenarios. It can be observed that our method results in better capture rates compared to the baseline in all the games.

### A. Reach-Avoid Tube Visualization

First, we visualize the 4-dimensional 1 vs. 1 reach-avoid tube  $\mathcal{RA}_\infty^{11}$  in a 2-dimensional plane by fixing the state of the defender at  $(0.3, -0.3)$ . If an attacker is within the reach-avoid tube, then it can win the game using optimal control inputs; otherwise, the defender wins. As shown in the top left subplot of Fig. 2, both attackers at  $(-0.5, 0.0)$  and  $(0.0, 0.8)$  lie out of the region; hence, each of these attackers will lose in a 1 vs. 1 game to the defender. Similarly, to visualize the 6-dimensional reach-avoid tube  $\mathcal{RA}_\infty^{21}$ , we fix the positions of attacker 1 at  $(-0.5, 0.0)$  and the defender at  $(0.3, -0.3)$ , which are shown in the top right subplot of Fig. 2. The attacker at  $(0.0, 0.8)$  lies out of the reach-avoid tube boundary, hence the defender at  $(0.3, -0.3)$  could capture both attackers using the optimal control inputs.

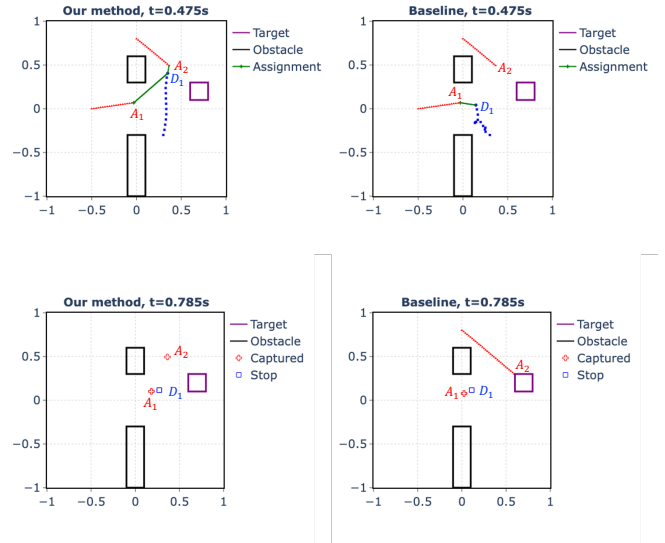
### B. The 2 vs. 1 Reach-Avoid Game Simulation

Fig. 3 shows the simulation results of two snapshots at  $t = 0.475s$  and  $t = 0.785s$ . With our method, the defender was assigned to capture both attackers until attacker  $A_2$  was



**Fig. 2:**  $\mathcal{RA}_\infty^{11}$  and  $\mathcal{RA}_\infty^{21}$  in a 2 vs. 1 reach-avoid game. **Top left figure:** We fix the position of a defender at  $x_D$  and visualize the set  $\mathcal{RA}_\infty^{11}$ . An attacker wins the game if its states are in the  $\mathcal{RA}_\infty^{11}$  region. **Top right figure:** The positions of a defender and an attacker are fixed to visualize the slice  $\mathcal{RA}_\infty^{21}$ . At least one of the attackers can reach the target set if  $x_{A_1}$  lies inside the region  $\mathcal{RA}_\infty^{21}$ . **Bottom figure:** We fix the positions of two attackers  $x_{A_1}, x_{A_2}$  to visualize the slice  $\mathcal{RA}_\infty^{21}$ . The defender can capture both attackers if it lies outside the region  $\mathcal{RA}_\infty^{21}$ .

captured at the time  $t = 0.475s$ , and eventually, the defender can capture the attacker  $A_1$ .



**Fig. 3:** A 2 vs. 1 reach-avoid game. In this figure, the plots on **left** column are simulations of the game using **our method** at different time steps. And the plots on the **right** are simulations using **baseline method** [15]. At the beginning of the game, the defender is assigned to capture both attackers and successfully captures them at the end using our method. In contrast, using the baseline method,  $A_1$  is captured first but then  $A_2$  is able to reach the target.

As the left column in Fig. 3 shows, at the beginning of the

game, the defender at  $(0.3, -0.3)$  is assigned to capture both attackers  $A_1$  and  $A_2$  by applying optimal control obtained by Eq. 11 using computed function  $\Phi^{21}$ . At the time  $t = 0.475s$ ,  $A_2$  was captured, and at  $t = 0.690s$   $A_1$  was captured by the defender. Using our method, both attackers were successfully captured.

In comparison, using the baseline method, the defender is only able to capture 1 attacker as shown in the right column of Fig. 3. Since any 1 vs. 1 assignment using the baseline is valid, initially the defender is assigned to capture attacker  $A_1$  and applies the optimal control computed by Eq. 8 using  $\Phi^{11}$ , which resulted in  $A_1$  being captured at the time  $t = 0.785s$ . However, the resulting relative position of the defender  $D_1$  allowed for attacker  $A_2$  to reach the target set at the end of the game.

### C. The 6 vs. 2 Reach-Avoid Game Simulation

Fig. 4 shows different snapshots of a 6 vs. 2 game. Using our proposed method, 6 attackers were able to be captured.

At the time  $t = 0.1s$ , each defender was assigned to capture 2 attackers. After attacker  $A_3$  was captured by defender  $D_2$ , the MIP assigned attacker  $A_4$  only to  $D_2$  as shown in the  $t = 0.3s$  snapshot while assigning  $D_1$  to both  $A_1$  and  $A_5$ . At the time  $t = 0.7s$ ,  $D_1$  was assigned to capture the rest two attackers  $A_1$  and  $A_4$ ; at this time,  $D_2$  did not get any more assignments and stopped. Finally, all attackers were captured at the time  $t = 1.0s$ . The reduction of the total number of attackers over time is consistent with the aforementioned property of Algorithm 1.

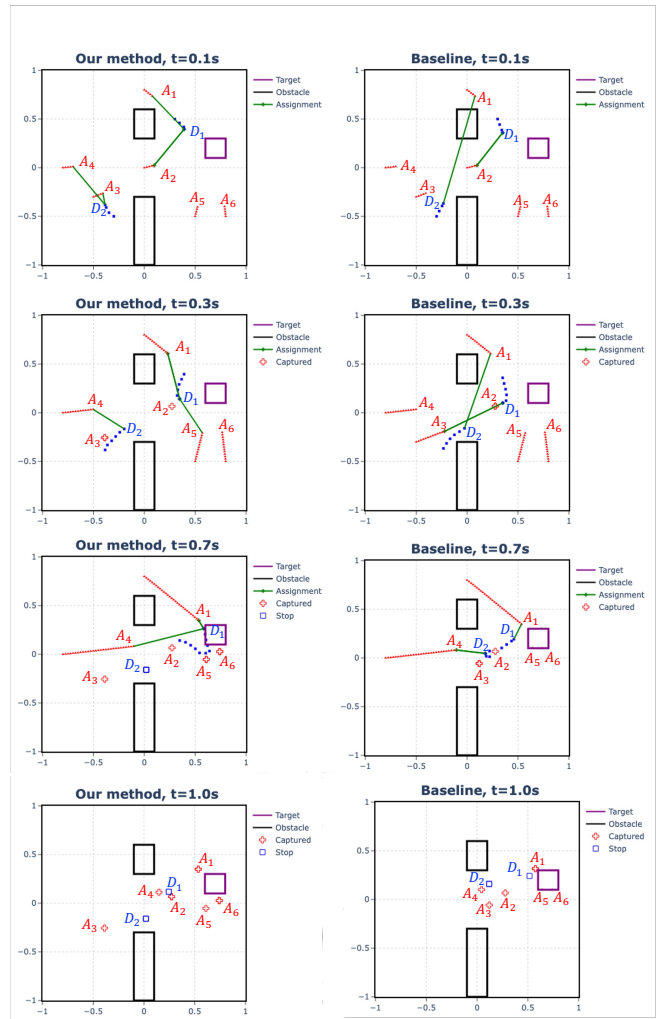
In comparison, when using the baseline method, only 4 attackers were successfully captured eventually. During the whole game, attackers  $A_5$  and  $A_6$  did not match to any defender and arrived at the target eventually. With the baseline method, one attacker could be assigned to at most one defender at a time, which would result in the defender acting more shortsightedly and missing the chances of stopping more attackers.

### D. The 8 vs. 4 Reach-Avoid Game Simulation

Fig. 5 shows the results of the 8 vs. 4 simulation. With our method, all 8 attackers were captured. Initially, at  $t = 0.1s$ , each defender was assigned to 2 attackers, as shown in the left column of Fig. 5. Because of that, over the course of 0.5 seconds, 4 attackers  $A_1$ ,  $A_3$ ,  $A_5$ , and  $A_8$  are already captured. At the time  $t = 1.0s$ , there was only one attacker  $A_2$  left, which is assigned to defender  $D_4$ . At the end of the game, all attackers fail to reach the target region without being captured.

In comparison, using the baseline method, 6 attackers were captured, while attackers  $A_7$  and  $A_8$  were never assigned to any defenders, and hence reached the target set as shown in the **right** column Fig. 5. The number of matching assignment is at most 4 during the whole simulation which is the same as the number of defenders.

It can be observed from our simulation results that the baseline method would require additional defenders in order to capture all the attackers .

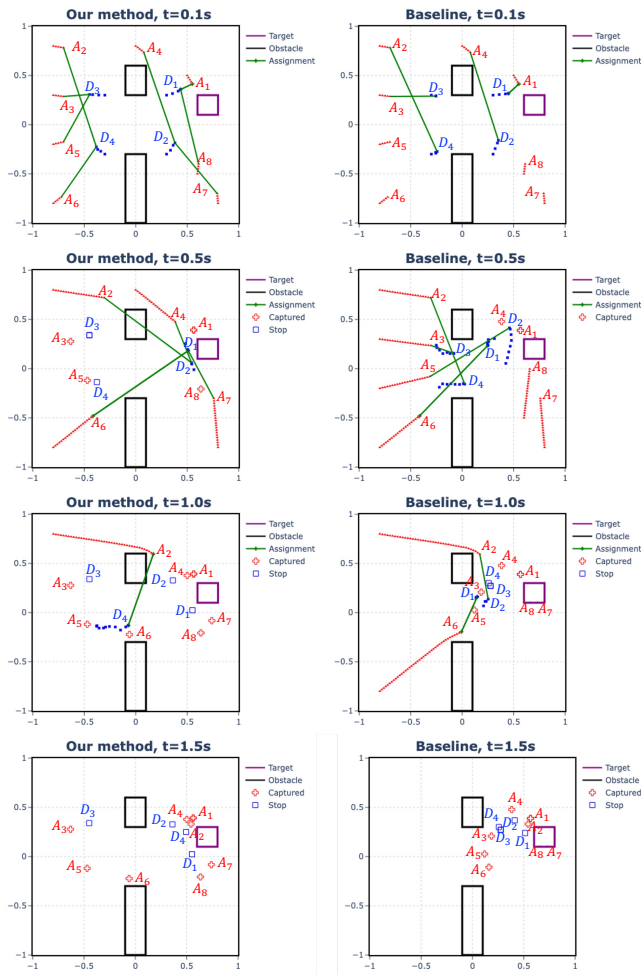


**Fig. 4:** 6 vs. 2 game. In this figure, the plots on the **left** column are simulations of the game using **our method** at different time steps. And the plots on the **right** are simulations using **baseline method** [15]. It can be observed at the end of the game, that our method can capture all attackers while the baseline can only capture 4 attackers with  $A_5, A_6$  arrived at the target set.

## V. CONCLUSION

In this paper, we have proposed an HJ reachability-based method to compute the winning-guaranteed set of states for 1 defender in a reach-avoid game against 2 attackers utilizing recent computational advances. Furthermore, we approximate the solution to multi-agent reach-avoid games by proposing a MIP method that maximizes the number of captured attackers. We proved the effectiveness of our method in simulations and showed that our method works very well in cases where defenders are fewer than attackers.

In the future, we will explore reach-avoid games with different configurations, for example, the 1 vs. 2 reach-avoid game with each agent 2-dimensional dynamics, and the 2 vs. 2 reach-avoid game where one of the agents has simple 1-dimensional dynamics. Besides, we will also use neural networks to approximate higher-dimensional games [26]. The new optimization scheme that can efficiently assign matching for these games is also in the future plan.



**Fig. 5:** 8 vs. 4 game. In this figure, the plots on the left column are simulations of the game using **our method** at different time steps. And the plots on the right are simulations using **baseline method** [15]. It can be observed at the end of the game, that our method can capture all attackers while the baseline can only capture 6 attackers with  $A_7, A_8$  arrived at the target set.

## REFERENCES

- [1] H. Christensen, N. Amato, H. Yanco, M. Mataric, H. Choset, A. Drobni, K. Goldberg, J. Grizzle, G. Hager, J. Hollerbach, *et al.*, “A roadmap for us robotics—from internet to robotics 2020 edition,” *Foundations and Trends® in Robotics*, vol. 8, no. 4, pp. 307–424, 2021.
- [2] K. Gao, J. Xin, H. Cheng, D. Liu, and J. Li, “Multi-mobile robot autonomous navigation system for intelligent logistics,” in *2018 Chinese Automation Congress (CAC)*, 2018.
- [3] S. Waydo and R. Murray, “Vehicle motion planning using stream functions,” in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2003.
- [4] R. Parasuraman, S. Galster, P. Squire, H. Furukawa, and C. Miller, “A flexible delegation-type interface enhances system performance in human supervision of multiple robots: empirical studies with roboflag,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 35, no. 4, pp. 481–493, 2005.
- [5] M. G. Earl and R. D’Andrea, “A decomposition approach to multi-vehicle cooperative control,” *Robotics and Autonomous Systems*, vol. 55, no. 4, pp. 276–291, 2007.
- [6] G. Ibragimov and N. Satimov, “A multiplayer pursuit differential game on a closed convex set with integral constraints,” in *Abstract and Applied Analysis*, vol. 2012, 2012.
- [7] M. Coon and D. Panagou, “Control strategies for multiplayer target-attacker-defender differential games with double integrator dynamics,”

- in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017.
- [8] X. Chen and J. Yu, “Reach-avoid games with two heterogeneous defenders and one attacker,” *IET Control Theory & Applications*, vol. 16, no. 3, pp. 301–317, 2022.
- [9] R. Yan, Z. Shi, and Y. Zhong, “Reach-avoid games with two defenders and one attacker: An analytical approach,” *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1035–1046, 2019.
- [10] R. Yan, Z. Shi, and Y. Zhong, “Cooperative strategies for two-evader-one-pursuer reach-avoid differential games,” *International Journal of Systems Science*, vol. 52, no. 9, pp. 1894–1912, 2021.
- [11] L. Liang, F. Deng, M. Lu, and J. Chen, “Analysis of role switch for cooperative target defense differential game,” *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 902–909, 2021.
- [12] M. Chen and C. J. Tomlin, “Hamilton-jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 333–358, 2018.
- [13] I. Mitchell, A. Bayen, and C. Tomlin, “A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games,” *IEEE Transactions on Automatic Control*, vol. 50, no. 7, pp. 947–957, 2005.
- [14] M. Bui, G. Giovanis, M. Chen, and A. Shriraman, “Optimizeddp: An efficient, user-friendly library for optimal control and dynamic programming,” *arXiv preprint arXiv:2204.05520*, 2022.
- [15] M. Chen, Z. Zhou, and C. J. Tomlin, “Multiplayer reach-avoid games via low dimensional solutions and maximum matching,” in *2014 American Control Conference*, 2014.
- [16] M. Chen, Z. Zhou, and C. J. Tomlin, “A path defense approach to the multiplayer reach-avoid game,” in *53rd IEEE Conference on Decision and Control*, 2014.
- [17] M. Chen, Z. Zhou, and C. J. Tomlin, “Multiplayer Reach-Avoid Games via Pairwise Outcomes,” *IEEE Transactions on Automatic Control*, vol. 62, pp. 1451–1457, Mar. 2017.
- [18] R. Yan, X. Duan, Z. Shi, Y. Zhong, and F. Bullo, “Matching-based capture strategies for 3d heterogeneous multiplayer reach-avoid differential games,” *Automatica*, vol. 140, p. 110207, 2022.
- [19] M. Chen, J. C. Shih, and C. J. Tomlin, “Multi-vehicle collision avoidance via hamilton-jacobi reachability and mixed integer programming,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016.
- [20] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, “Reach-avoid problems with time-varying dynamics, targets and constraints,” in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, 2015.
- [21] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, “A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag,” in *2011 IEEE International Conference on Robotics and Automation*, 2011.
- [22] I. M. Mitchell *et al.*, “A toolbox of level set methods,” *UBC Department of Computer Science Technical Report TR-2007-11*, p. 31, 2007.
- [23] S. Osher and R. Fedkiw, *The Level Set Methods and Dynamic Implicit Surfaces*, vol. 57, pp. xiv+273. 05 2004.
- [24] I. M. Mitchell, *Application of level set methods to control and reachability problems in continuous and hybrid systems*. stanford university, 2002.
- [25] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, “Hamilton-jacobi reachability: A brief overview and recent advances,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017.
- [26] S. Bansal and C. J. Tomlin, “Deepreach: A deep learning approach to high-dimensional reachability,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [27] J. Li, D. Lee, S. Sojoudi, and C. J. Tomlin, “Infinite-horizon reach-avoid zero-sum games via deep reinforcement learning,” *arXiv preprint arXiv:2203.10142*, 2022.
- [28] H. G. Santos and T. Toffolo, “Mixed integer linear programming with python,” *COINOR Computational Infrastructure for Operations Research*, 2020.
- [29] Gurobi Optimization, LLC, “Gurobi Optimizer Reference Manual,” 2023.
- [30] I. I. Cplex, “V12. 1: User’s manual for cplex,” *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.