2014 American Control Conference (ACC)
June 4-6, 2014. Portland, Oregon, USA

Multiplayer Reach-Avoid Games via Low Dimensional Solutions and Maximum Matching

Mo Chen, Zhengyuan Zhou, and Claire J. Tomlin

Abstract— **We consider a multiplayer reach-avoid game with an equal number of attackers and defenders moving with simple dynamics on a two-dimensional domain possibly with obstacles. The attacking team attempts to send as many attackers to a certain target location as possible quickly while the defenders aim to capture the attackers to prevent the attacking team from reaching its goal. The analysis of problems like this plays an important role in collision avoidance, motion planning, and aircraft control, among other applications. Computing optimal solutions for such multiplayer games is intractable due to numerical intractibility. This paper provides a first attempt to address such computational intractability by combining maximum matching in graph theory with the classical Hamilton-Jacobi-Isaacs approach. In addition, our solution provides an initial step to take cooperation into account by computing maximum matching in real time.**

I. INTRODUCTION

Differential games are powerful theoretical tools in robotics, aircraft control, security, and other domains [1], [2], [3]. The multiplayer reach-avoid game (to be defined precisely in Section II) is a differential game between two adversarial teams of cooperating players, where one team attempts to reach a certain target quickly while the other team aims to delay, or if possible, prevent the opposing team from reaching the target. One example of a reach-avoid game is the popular game Capture-the-Flag (CTF) [4], [5]. In robotics and automation, CTF has been explored most notably in the Cornell Roboflag competition, where two opposing teams of mobile robots are directed by human players to play the game [6]. A number of results related to motion planning and human-robot interactions have been reported from the competition [7], [8], [9], [10].

A multiplayer reach-avoid game is a complex game due to both the conflicting goals of the two teams and the cooperation among the players within each team, rendering the optimal solution for each team nontrivial to obtain and visualize. Previous work [4], [5] has shown that even in a 1 vs. 1 scenario, human agents are sometimes unable to find the optimal way to play, losing in situations in which an optimal winning strategy exists. For general multiplayer reach-avoid games, optimal solutions are extremely difficult to compute due to the intrinsic high dimensionality of the joint state space. Multiplayer differential games have been previously addressed using various techniques. In [7], where a team of defenders assumes that the attackers move towards their target in straight lines, a mixed-integer linear programming approach was used. In [11], optimal defender strategies are determined using a linear program, with the assumption that the attackers use a linear feedback control law. In complex pursuit-evasion games where players may change roles over time, nonlinear model-predictive control [12] and approximate dynamic programming [13] approaches have been investigated. In both cases, opponent strategies are estimated based on explicit prediction models.

The above-mentioned techniques tend to work well in the situations in which accurate models of the opponent team can be obtained. While those techniques have proven to be effective in their corresponding scenarios, they cannot be easily adapted to solve a general multiplayer reach-avoid game when no prior information on each side is known. The ideal framework to use for such a general multiplayer reach-avoid game is the Hamilton-Jacobi-Isaacs (HJI) approach [14], in which an HJI partial differential equation (PDE) is solved to obtain optimal strategies for both teams. In this case, if both teams play optimally, the result of the game is determined by the joint initial condition of the players. In addition, many numerical tools [15], [16], [17] have been developed to carry out the computations and leverage the power of the HJI framework when the dimensionality of the problem is low. These tools have been employed to successfully solve a variety of differential games, path planning problems, and optimal control problems, including aircraft collision avoidance [15], automated in-flight refueling [18], and two-player reach-avoid games [5]. These tools offer tremendous flexibility in terms of the player dynamics and terrain, and do not explicitly assume any specific control strategy or prediction models for the players.

Despite the power that the HJI framework and the numerical tools have to offer, solving a general multiplayer reach-avoid game is computationally intractable due to the curse of dimensionality: the joint state space of two N-player teams in a two-dimensional (2D) domain is $4N$-dimensional ($4N$D). Numerically, when the state space is discretized, the number of nodes scales exponentially with the number of dimensions. Therefore, computing optimal solutions for a general multiplayer reach-avoid game in the HJI framework is out of reach. As a result, the inherent trade-off between optimality of the solutions and computational complexity

This work has been supported in part by NSF under CPS:ActionWebs (CNS-931843), by ONR under the HUNT (N0014-08-0696) and SMARTS (N00014-09-1-1051) MURIs and by grant N00014-12-1-0609, by AFOSR under the CHASE MURI (FA9550-10-1-0567).

M. Chen, and C. J. Tomlin are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA {mochen72,tomlin}@eecs.berkeley.edu

Z. Zhou is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA zyzhou@stanford.edu

We thank Haomiao Huang for sharing MatLab code for 4D HJ reachability calculations.

must be considered and made.

In the multiplayer reach-avoid game, we consider each defender-attacker pair and compute the optimal solutions for both players using the HJI framework. We then invoke the graph-theoretical maximum matching algorithm [19], [20] to determine optimal pairings. This procedure approximates the solution to the multiplayer game by combining the solutions from the N^2 two-player games between each attacker-defender pair. This way, the computation complexity is reduced from solving a $4N$D HJI PDE to solving N^2 4D HJI PDEs. If we also assume that players on each team have the same dynamics, then only *one* 4D HJI PDE needs to be solved. This is because an HJI calculation partitions the joint state space of the two players into a winning region for the attacker and a winning region for the defender.

Our contributions can hence mainly be stated as follows. First, it is an extension of the work in [5] on two-player reach-avoid games to multiplayer reach-avoid games. Our extension is easily implemented at almost no additional computational cost compared to the two-player games. Hence, our approach provides an appealing solution, especially when the number of players becomes large. In addition, some cooperation is taken into account by the maximum matching process. Second, as there are other ways to trade off optimality of the solutions and computational complexity, our extension provides an initial, in fact, the only baseline benchmark in the HJI framework to which other approaches can be compared. This is again due to the numerical intractibility of HJI computations, which are practically limited to at most five-dimensional systems. Hence, any sub-game one hopes to solve using the HJI framework can involve at most two players. Such a benchmark can also be viewed as an invitation for other approaches that do not require solving HJI PDEs to address the complexity-optimality trade-off.

II. Problem Formulation

We consider a multiplayer reach-avoid game between a team of N attackers, $\{P_{A_i}\}_{i=1}^N = \{P_{A_1}, \dots, P_{A_N}\}$ and a team of N defenders, $\{P_{D_i}\}_{i=1}^N = \{P_{D_1}, \dots, P_{D_N}\}$. Each player is confined in a bounded, open domain $\Omega \subset \mathbb{R}^2$, which can be partitioned as follows: $\Omega = \Omega_{free} \cup \Omega_{obs}$. Ω_{free} is a compact set representing the free space in which the players can move, while $\Omega_{obs} = \Omega \setminus \Omega_{free}$ represents the obstacles that obstruct movement in the domain. Let $x_{A_i}, x_{D_j} \in \mathbb{R}^2$ denote the states of the players P_{A_i} and P_{D_j} respectively. Initial conditions of the players are denoted by $x_{A_i}^0, x_{D_i}^0 \in \Omega_{free}, i = 1, 2, \dots, N$. We assume that the dynamics of the players are defined by the following decoupled system for $t \geq 0$:

$$\begin{aligned} \dot{x}_{A_i}(t) &= v_A a_i(t), \qquad x_{A_i}(0) = x_{A_i}^0, \\ \dot{x}_{D_i}(t) &= v_D d_i(t), \qquad x_{D_i}(0) = x_{D_i}^0, \\ &i = 1, 2, \dots, N \end{aligned} \tag{1}$$

where $a_i(\cdot), d_i(\cdot)$ represent the control functions of P_{A_i} and $P_{D_i}, i = 1, 2, \dots, N$ respectively. The attackers $\{P_{A_i}\}_{i=1}^N$ have the same maximum speed v_A and the defenders $\{P_{D_i}\}_{i=1}^N$ have the same maximum speed v_D. We assume that the control functions $a_i(\cdot), d_i(\cdot)$ are drawn from the set $\Sigma = \{\sigma \colon [0, \infty) \to \overline{B}_n \mid \sigma \text{ is measurable}\}$, where $\overline{B}_n$ denotes the closed unit ball in $\mathbb{R}^2$. As a clarification on the notation and terminology, the control functions (with a dot notation, e.g. $a_i(\cdot), d_i(\cdot), u(\cdot)$ etc.) which are the entire control trajectories, are distinguished from the control inputs (such as $a_i, a_i(t), d_i, d_i(t)$ etc.) which are the instantaneous control inputs. Furthermore, given $x_{A_i}^0 \in \Omega_{free}$, we define the admissible control function set for P_{A_i} to be the set of all control functions such that $x_{A_i}(t) \in \Omega_{free}, \forall i$, $\forall t \geq 0$. The admissible control function set for defenders $P_{D_i}, i = 1, 2, \dots, N$ is defined similarly, given that $x_{D_i}^0 \in \Omega_{free}$. The joint state of all the players is denoted by $\mathbf{x} = (x_{A_1}, \dots x_{A_N}, x_{D_1}, \dots, x_{D_N})$. The joint initial condition is denoted by $\mathbf{x^0} = (x_{A_1}^0, \dots, x_{A_N}^0, x_{D_1}^0, \dots, x_{D_N}^0)$.

In this reach-avoid game, the attacking team aims to reach the target $\mathcal{T} \subset \Omega_{free}$, a compact subset of the domain, without getting captured by the defenders. The capture conditions are formally described by the capture sets $\mathcal{C}_{ij} \subset \Omega^{2N}$ for the pairs of the players $\left(P_{A_i}, P_{D_j}\right), i, j = 1, \dots, N$. In general, $\mathcal{C}_{ij}$ can be an arbitrary compact subset of Ω^{2N}, which represents the set of the joint player states $\mathbf{x}$ at which P_{A_i} is captured by P_{D_j}. Hence, in the general case, the interpretation of capture is given by the set $\mathcal{C}_{ij}$, which in turn depends on the specific situation one wishes to model. In this paper, we define the capture sets to be $\mathcal{C}_{ij} = \left\{\mathbf{x} \in \Omega^{2N} \mid \|x_{A_i} - x_{D_j}\|_2 \leq R_C\right\}$, the interpretation of which is that P_{A_i} is captured by P_{D_j} if P_{A_i}'s position is within R_C of P_{D_j}'s position.

In our multiplayer reach-avoid game, we are interested in determining how many attackers are able to reach the target $\mathcal{T}$ without being captured. An illustration of the game setup is shown in Figure 1.

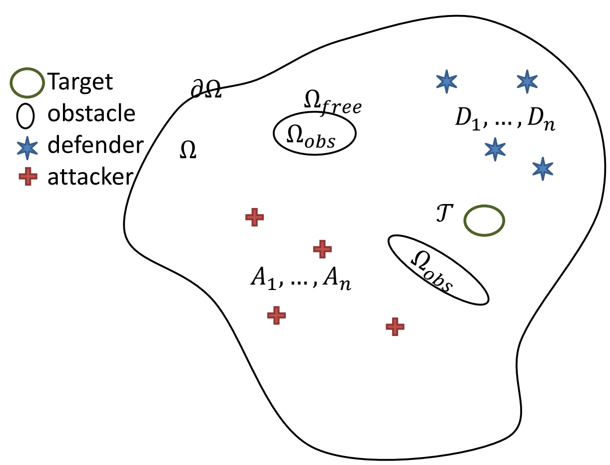


Fig. 1: The components of a multiplayer reach-avoid game.

Consider the special case in which each team only has one player. We denote the attacker as P_A, the defender as P_D, their states and initial conditions as x_A, x_D, x_A^0, x_D^0. Their dynamics are

$$\begin{aligned} \dot{x}_A(t) &= v_A a(t), \quad x_A(0) = x_A^0, \\ \dot{x}_D(t) &= v_D d(t), \quad x_D(0) = x_D^0. \end{aligned} \tag{2}$$

The players' joint state and joint initial condition become $\mathbf{x} = (x_A, x_D), \mathbf{x^0} = (x_A^0, x_D^0)$ respectively. The capture set for P_A is then simply

$$\mathcal{C} = \left\{(x_A, x_D) \in \Omega^2 \mid \|x_A - x_D\|_2 \le R_C\right\}. \tag{3}$$

The attacker wins when P_A reaches the target $\mathcal{T}$ without being captured by the defender P_D. If the defender P_D can delay P_A from reaching $\mathcal{T}$ indefinitely, the defender wins.

This two-player reach-avoid game and its variant were first studied in [5] and [21]. In this paper, we extend the HJI framework to deal with the multiplayer reach-avoid games.

III. Solution

We first describe the HJ reachability framework for solving differential games with arbitrary terrain, domain, obstacles, target set, and player velocities based on [15], [22], [5]. The results of HJ computation assume a closed-loop strategy for both players given previous information of the other players.

Solving the $4N$D PDE corresponding to the full multiplayer game is numerically intractable, so we solve a 4D HJI PDE instead, and construct an approximation to the $4N$D solution using maximum matching. The approximation provides an upper bound on the number of attackers who are able to reach the target.

A. Hamilton-Jacobi Reachability

The setup for using HJ reachability to solve differential games can be found in [15], [22], [5]. In summary, we are given the continuous dynamics of the system state:

$$\dot{\mathbf{x}} = f(\mathbf{x}, u, d), \mathbf{x}(0) = \mathbf{x^0}, \tag{4}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the system state, $u \in \mathbb{U}$ is the joint control input of the attacking team, and $d \in \mathbb{D}$ is the joint control input of the defending team. The sets $\mathbb{U}$ and $\mathbb{D}$ represent the sets of the joint admissible control inputs of the attacking team and the defending team, respectively.

We specify the terminal set R (described in detail in Section III-B) as the attackers' winning condition, and propagate backwards this set subject to the constraint imposing that the attackers be outside the capture regions and the obstacles. This constraint is described by the avoid set A.

More precisely, the HJ reachability calculation is as follows. First, given a set G, the level set representation of G is a function $\phi_G : \mathbb{R}^n \to \mathbb{R}$ such that $G = \{\mathbf{x} \in \mathbb{R}^n \mid \phi_G \le 0\}$. In particular, the terminal set R and the avoid set A will be represented by the functions ϕ_R and ϕ_A respectively.

Let $\Phi : \mathbb{R}^n \times [-T, 0] \to \mathbb{R}$ be the viscosity solution [23] to the constrained terminal value HJI PDE:

$$\frac{\partial \Phi}{\partial t} + \min\left[0, H\left(\mathbf{x}, \frac{\partial \Phi}{\partial \mathbf{x}}\right)\right] = 0,\ \Phi(\mathbf{x}, 0) = \phi_R(\mathbf{x}) \tag{5}$$

subject to

$$\Phi(\mathbf{x}, t) \ge -\phi_A(\mathbf{x}),$$

where the optimal Hamiltonian is given by

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

By the argument presented in [15] and [24], the set of initial conditions from which the attackers are guaranteed to win within time T is given by

$$\mathcal{RA}_T(R, A) := \{\mathbf{x} \in \mathbb{R}^n \mid \Phi(\mathbf{x}, -T) \le 0\}. \tag{6}$$

Hence, $\Phi(\mathbf{x}, -T)$ is the level set representation of $\mathcal{RA}_T(R, A)$. The optimal control input for the attacking team is given by [25], [26], [5]:

$$u^*(\mathbf{x}, t) = \arg\min_{u \in \mathbb{U}} \max_{d \in \mathbb{D}} p(\mathbf{x}, -t)^T f(\mathbf{x}, u, d), t \in [0, T] \tag{7}$$

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

Similarly, the optimal control input for the defending team is given by

$$d^*(\mathbf{x}, t) = \arg\max_{d \in \mathbb{D}} p(\mathbf{x}, -t)^T f(\mathbf{x}, u^*, d),\ t \in [0, T]. \tag{8}$$

Taking $T \to \infty$, we obtain the set of initial conditions from which the attackers are guaranteed to win. We denote this set $\mathcal{RA}_\infty(R, A)$. The set of initial conditions from which the defenders are guaranteed to win is given by all points not in $\mathcal{RA}_\infty(R, A)$. For an N vs. N game on a two-dimensional domain $\Omega \subset \mathbb{R}^2$, the reachable set $\mathcal{RA}_\infty(R, A)$ is $4N$D.

A highly accurate numerical solution to Equation (5) can be computed using the Level Set Toolbox for MATLAB [22].

B. Hamilton-Jacobi Reachability for the Two Player Game

In general, an HJI PDE of dimensions higher than five cannot be solved practically, so we are limited to only being able to solve the HJI PDE corresponding to a two-player game in which each player's state space is 2D. We will solve the multiplayer game by combining the solution to the two-player game and maximum matching from graph theory.

In the two-player game, the goal of the attacker is to reach the target set $\mathcal{T}$ while avoiding capture by the defender. This terminal set R is represented by the attacker being inside $\mathcal{T}$. En route to $\mathcal{T}$, the attacker must avoid capture by the defender. This is represented by the set $\mathcal{C}$.

Both players also need to avoid the obstacles Ω_{obs}, which can be considered as the locations in Ω where the players have zero velocity. In particular, the defender wins if the attacker is in Ω_{obs}, and vice versa. Therefore, we define the terminal set and avoid set as

$$\begin{aligned} R = &\left\{\mathbf{x} \in \Omega^2 \mid x_A \in \mathcal{T} \wedge \|x_A - x_D\|_2 > R_C\right\} \\ &\cup \left\{\mathbf{x} \in \Omega^2 \mid x_D \in \Omega_{obs}\right\} \end{aligned} \tag{9}$$

$$\begin{aligned} A = &\left\{\mathbf{x} \in \Omega^2 \mid \|x_A - x_D\|_2 \le R_C\right\} \\ &\cup \left\{\mathbf{x} \in \Omega^2 \mid x_A \in \Omega_{obs}\right\} \end{aligned} \tag{10}$$

Given these sets, we can define the corresponding level set representations ϕ_R, ϕ_A, and solve (5). If $\Omega \subset \mathbb{R}^2$, the result is $\mathcal{RA}_\infty(R, A) \in \mathbb{R}^4$, a 4D reach-avoid set with the level set representation $\Phi(\mathbf{x}, -\infty)$. The attacker wins if and only if $(x_A^0, x_D^0) = \mathbf{x^0} \in \mathcal{RA}_\infty(R, A)$.

If $\mathbf{x^0} \in \mathcal{RA}_\infty(R, A)$, then the attacker is guaranteed to win the game. Applying Equation (7) to the two-player game, we obtain the explicit winning strategy is given in [5]:

$$a^*(x_A, x_D, t) = -v_A \frac{p_u(x_A, x_D, -t)}{\|p_u(x_A, x_D, -t)\|_2}. \quad (11)$$

where $p = (p_u, p_d) = \frac{\partial \Phi}{\partial (x_A, x_D)}$.

Similarly, if $\mathbf{x^0} \notin \mathcal{RA}_\infty(R, A)$, then the defender is guaranteed to win the game. Applying (8) to the two-player game, we obtain the explicit winning strategy given in [5]:

$$d^*(x_A, x_D, t) = v_D \frac{p_d(x_A, x_D, -t)}{\|p_d(x_A, x_D, -t)\|_2}. \quad (12)$$

C. Maximum Matching

We can determine whether the attacker can win the multiplayer reach-avoid game by combining the solution to the two-player game, characterized by $\mathcal{RA}_\infty(R, A)$, and maximum matching [19], [20] from graph theory as follows:

1) Compute $\mathcal{RA}_\infty(R, A)$
2) Construct a bipartite graph with two sets of nodes $\{P_{A_i}\}_{i=1}^N, \{P_{D_i}\}_{i=1}^N$, where each node represents a player.
3) For each P_{D_i}, determine whether P_{D_i} can win against P_{A_j}, for all j. Given $\mathcal{RA}_\infty(R, A)$, we can determine the winner of the two player game for any given pair $(x_{A_i}, x_{D_j}) \forall (i, j)$.
4) Form a bipartite graph: Draw an edge between P_{D_i} and P_{A_j} if P_{D_i} wins against P_{A_j}
5) Run any matching algorithm to find a maximum matching in the graph. This can be done using, for example, a linear program [19], or the Hopcroft-Karp algorithm [20].

After constructing the bipartite graph, if the maximum matching is of size m, then the defending team would be able to prevent *at least* m attackers from reaching the target. Alternatively, $N - m$ is an *upper bound* on the number of attackers that can reach the target.

For intuition, consider the following specific cases of m. If $m = N$, then no attacker will be able to reach the target. If $m = 0$, then there is no initial pairing that will prevent any attacker from reaching the target; however, the attackers are not guaranteed to all reach the target, as $N - m = N$ is only an upper bound on the number of attackers who can reach the target. Finally, $m = N - M + 1$, then the attacking team would only be able to send at most $N - m = M - 1$ attackers to the target.

The optimal strategy for the defenders can be obtained from (12). If the i^{th} defender P_{D_i} is assigned to defend against the j^{th} attacker P_{A_j} by the maximum matching, then the strategy that guarantees that P_{A_j} never reaches the target is given by

$$d_i^*(x_{A_j}, x_{D_i}, t) = v_D \frac{p_d(x_{A_j}, x_{D_i}, -t)}{\|p_d(x_{A_j}, x_{D_i}, -t)\|_2} \quad (13)$$

The entire procedure of applying maximum matching to the 4D HJ reachability calculation is illustrated in Figure 2.

Our solution to the multiplayer reach-avoid game is an approximation to the optimal solution that would be obtained by directly solving the $4N$D HJI PDE; it is conservative for the defending team because by creating defender-attacker pairs, each defender restricts its attention to only one opposing player. For example, if no suitable matching is found, the defending team is not guaranteed to allow all attackers to reach the target, as the defending team could potentially capture some attackers without using a strategy that creates defender-attacker pairs. Nevertheless, our solution is able to overcome the numerical intractibility to approximate a reachability calculation, and is useful in many game configurations.

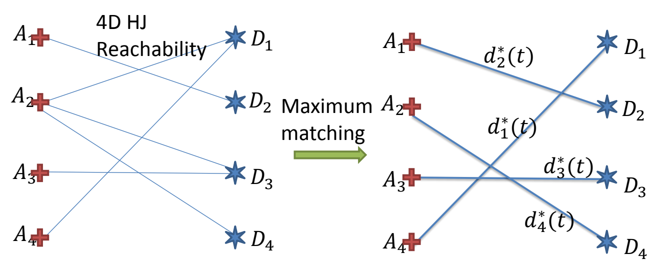


Fig. 2: The 4D HJ reachability and maximum matching approximation to the multiplayer reach-avoid game. A maximum matching of size m indicates that at most $N - m$ attackers will be able to reach the target.

D. Time-Varying Defender-Attacker Pairings

The procedure in Section III-C assigns an attacker to each defender that is part of a maximum pairing in the beginning of the game, and the assignment does not change during the course of the game. However, the bipartite graph and its corresponding maximum matching can be updated as the players change positions during the game. Because $\mathcal{RA}_\infty(R, A)$ captures the winning conditions for every joint defender-attacker configuration given $\Omega, \Omega_{obs}, \mathcal{T}$, this update can be performed in real time by the following procedure:

1) Given the position of each player, determine whether each defender can win against each attacker.
2) Construct the bipartite graph and find its maximum matching to assign an attacker to each defender that is part of the maximum matching.
3) For a chosen duration Δ, compute the optimal control input and trajectory for each defender that is part of the maximum matching via Equation (13). For the rest of the defenders and for all attackers, compute the trajectories assuming any control function.
4) Repeat the procedure with the new player positions.

As $\Delta \to 0$, the above procedure computes a bipartite graph and its maximum matching as a function of time. Whenever the maximum matching is not unique, the defenders can choose a different maximum matching and still be guaranteed to prevent the same number of attackers from reaching the target. As long as each defender uses the optimal control input given in Equation (13), the size of the maximum matching can never decrease as a function of time.

On the other hand, it is possible for the size of the maximum matching to increase as a function of time. This occurs if the joint configuration of the players becomes such that the resulting bipartite graph has a bigger maximum matching than before, which may happen since the size of the maximum matching only gives an upper bound on the number of attackers that are able to reach the target. Furthermore, there is no numerically tractable way to compute the joint optimal control input for the attacking team, so a suboptimal strategy from the attacking team can be expected, making an increase of maximum matching size likely. Determining defender control strategies that optimally promote an increase in the size of the maximum matching would be an important step towards the investigation of cooperation, and will be part of our future work.

IV. Computation Results

We illustrate our reachability and maximum matching approach in the example below. The HJ reach-avoid sets are calculated using the Level Set Toolbox [22] developed at the University of British Columbia. We calculated $\mathcal{RA}_\infty(R, A)$ by incrementing T until $\mathcal{RA}_T(R, A)$ converges. The computation of $\mathcal{RA}_\infty(R, A)$, done on a $45 \times 45 \times 45 \times 45$ grid, took approximately 30 minutes on a Lenovo T420s laptop with a Core i7-2640M processor.

The example is shown in Figures 3. There are four attackers and four defenders playing on a square domain with obstacles; the defenders have a capture radius of 0.1 units. All players have equal speeds ($v_D = v_A$).

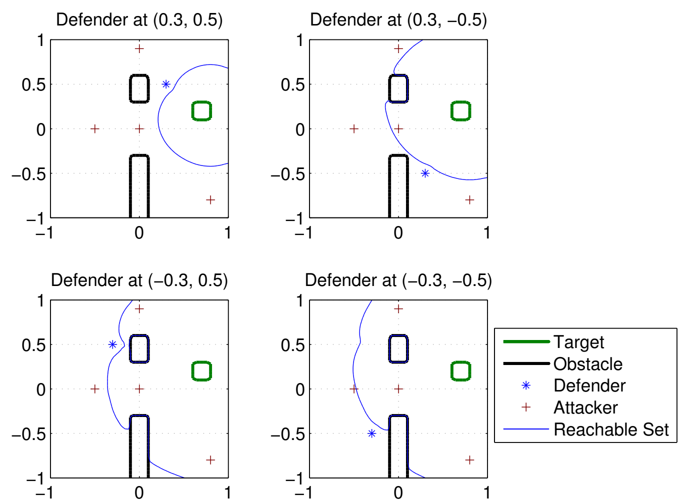
$\mathcal{RA}_\infty(R, A)$ is a 4D set that represents the joint configurations in which the attacker wins the game. To visualize the 4D set in 2D, we view the reach-avoid set at the slices representing the positions of particular players. Figure 3a shows boundaries of $\mathcal{RA}_\infty(R, A)$ with fixed defender positions. In each subplot, attackers which are closer to the target set than the reach-avoid set boundary win against the particular defender. For example, in the right top subplot of Figure 3a, the defender at $(0.3, -0.5)$ loses to the attacker at $(0, 0)$, but wins against the other three attackers.

Similarly, Figures 3b shows boundaries of $\mathcal{RA}_\infty(R, A)$ with fixed attacker positions. Defenders which are closer to the target set than the reach-avoid set boundary win against the particular attacker. For example, in the bottom left subplot, the attacker at $(0, 0)$ wins against the defender at $(0.3, 0.5)$ but loses against the other three defenders.

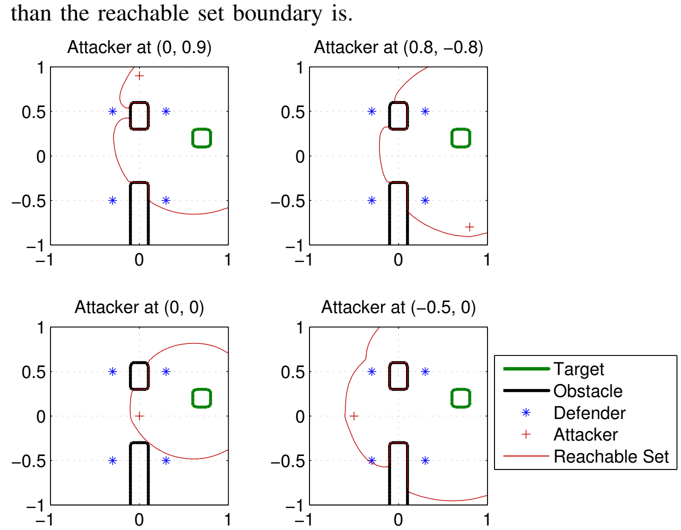
Figure 4 shows the resulting bipartite graph (thin solid blue lines) and the maximum matching (thick dashed blue lines) after applying the algorithm described in Section III-C. The maximum matching is of size 3, which means that at most 1 attacker will be able to reach the target.

Figure 5 shows the result of a 4 vs. 4 reach-avoid game simulation being played out in a course of 0.6 time units to illustrate the potential usefulness of time-dependent defender-attacker pairings. Every $\Delta = 0.005$ time units, a bipartite graph and its maximum matching are computed according to the algorithm in Section III-D. The defender that is not part of the maximum matching plays optimally according to (13) against the closest attacker. The attackers use the suboptimal strategy of taking the shortest path to the target while steering 0.125 units clear of the obstacles and disregarding the control inputs of other players in the game.

The initial size of the maximum matching is 3. At $t = 0.4$, because the attackers have not been playing optimally, the attacker at $(-0.50, 0.16)$ becomes in a losing position against the defender at $(-0.26, -0.34)$. Thus, the maximum matching now assigns the attacker at $(-0.50, 0.16)$ to the defender at $(-0.26, -0.34)$. At the same time, the defender at $(-0.09, -0.22)$ switches from defending the attacker at $(-0.50, 0.16)$ to defending the attacker at $(-0.27, 0.14)$, creating a perfect matching and preventing any attacker from reaching the target.



(a) Slices of reach-avoid set at the four defender positions. The defender wins against any attacker who is farther away from the target than the reachable set boundary is.



(b) Slices of reach-avoid set at the the four attacker positions. Each attacker is wins against any defender who is farther away from the target than the reachable set boundary is.

Fig. 3: A 4 vs. 4 reach-avoid game in which all players have equal maximum speeds.

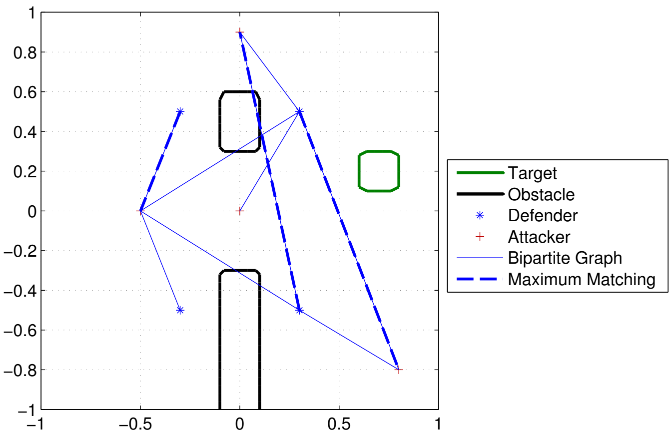


Fig. 4: Bipartite graph and maximum matching results. Each edge (solid blue line) connects a defender to an attacker against whom the defender is guaranteed to win, creating a bipartite graph. A maximum matching (dashed blue line) of size 3 indicates at most one attacker can reach the target.

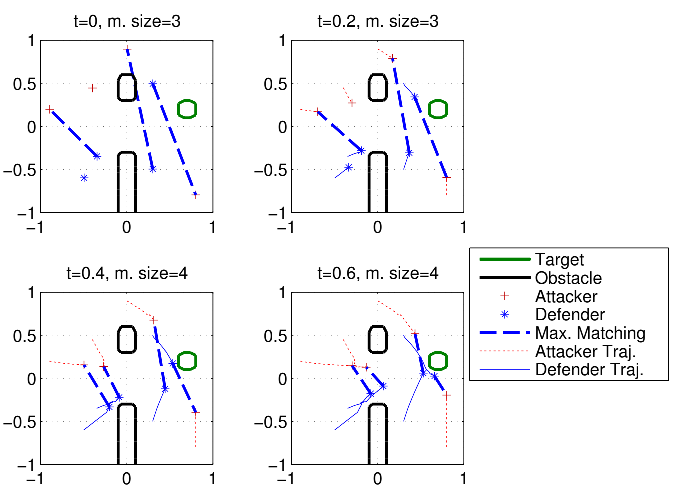


Fig. 5: Real-time maximum matching updates. The defenders can update the bipartite graph and maximum matching via the procedure described in Section III-D in real time. Because the attacking team is not playing optimally, the defending team finds a perfect matching after $t = 0.4$.

V. Conclusion

By solving a single 4D HJI PDE representing the two-player reach-avoid game, we obtained the winners in all attacker-defender pairs. Then, a maximum matching algorithm determines the pairing that prevents the maximum number of attackers from reaching the target. Calculating time-varying defender-attacker pairings allows the defending team to potentially increase the size of the maximum matching over time.

References

[1] Department of the Air Force, "United states air force unmanned aircraft systems flight plan 2009-2047," *oai.dtic.mil*, Jan 2009.

[2] H. Erzberger, "Automated conflict resolution for air traffic control," *25 th International Congress of the Aeronautical Sciences*, Jan 2006.

[3] A. Madrigal, "Autonomous robots invade retail warehouses," http://www.wired.com/wiredscience/2009/01/retailrobots/.

[4] H. Huang, "Reachability-based control for human and autonomous agents in adversarial games," Ph.D. dissertation, Stanford University, 2012.

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

[6] R. D'Andrea and M. Babish, "The roboflag testbed," *Proceedings of the American Control Conference*, vol. 1, pp. 656–660, 2003.

[7] M. 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.

[8] M. Campbell, R. D'Andrea, D. Schneider, A. Chaudhry, S. Waydo, J. Sullivan, J. Veverka, and A. Klochko, "Roboflag games using systems based, hierarchical control," *Proceedings of the American Control Conference*, vol. 1, pp. 661–666, 2003.

[9] S. Waydo and R. Murray, "Vehicle motion planning using stream functions," *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03*, pp. 2484–2491, 2003.

[10] 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, p. 481, 2005.

[11] G. Chasparis and J. Shamma, "Linear-programming-based multi-vehicle path planning with adversaries," *American Control Conference, 2005. Proceedings of the 2005*, pp. 1072–1077, 2005.

[12] J. Sprinkle, J. Eklund, H. Kim, and S. Sastry, "Encoding aerial pursuit/evasion games with fixed wing aircraft into a nonlinear model predictive tracking controller," *Decision and Control, 43rd IEEE Conference on*, 2004.

[13] J. McGrew, J. How, L. Bush, B. Williams, and N. Roy, "Air combat strategy using approximate dynamic programming," *AIAA Guidance, Navigation, and Control Conference*, Aug 2008.

[14] R. Isaacs, *Differential Games*. New York: Wiley, 1967.

[15] 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, July 2005.

[16] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996. [Online]. Available: http://www.pnas.org/content/93/4/1591.abstract

[17] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002, ISBN: 978-0-387-95482-0.

[18] J. Ding, J. Sprinkle, S. S. Sastry, and C. J. Tomlin, "Reachability calculations for automated aerial refueling," in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.

[19] A. Schrijver, *Combinatorial Optimization : Polyhedra and Efficiency (Algorithms and Combinatorics)*. Springer, Jul. 2004. [Online]. Available: http://www.worldcat.org/isbn/3540204563

[20] M. Karpinski and W. Rytter, *Fast parallel algorithms for graph matching problems*. New York, NY, USA: Oxford University Press, Inc., 1998.

[21] Z. Zhou, R. Takei, H. Huang, and C. Tomlin, "A general, open-loop formulation for reach-avoid games," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 2012, pp. 6501–6506.

[22] I. Mitchell, *A Toolbox of Level Set Methods*, 2009, http://people.cs.ubc.ca/ mitchell/ToolboxLS/index.html.

[23] M. G. Crandall and P.-L. Lions, "Viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society*, vol. 277, no. 1, pp. 1–42, 1983.

[24] I. Mitchell, "Application of level set methods to control and reachability problems in continuous and hybrid systems," Ph.D. dissertation, Stanford University, 2002.

[25] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, pp. 349 – 370, 1999.

[26] C. Tomlin, J. Lygeros, and S. Shankar Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949 –970, Jul 2000.