Weighted Matching-Based Capture Strategies for 3D Heterogeneous Multi-Player Reach-Avoid Differential Games

Prajwal Vijay, Abinash Agasti, Bharath Bhikkaji[1]

[1]Electrical Engineering Department, Indian Institute of Technology, Madras

Introduction

We analyze a 3D multiplayer Pursuer-Evader differential game with Play and Goal regions. Multiple pursuers defend the goal region by consecutively capturing multiple evaders in the play region. The players can have heterogeneous moving speeds and the pursuers have heterogeneous capture radii. For the analysis, we decompose the game into smaller subgames, which involve a coalition of pursuers chasing one evader. We compute the value function of the Hamilton-Jacobi-Isaacs (HJI) equation for these subgames. We show that under these conditions the value functions actually refer to the closest distance an evader can reach to the goal, before being caught by the pursuers. Assuming all players are playing optimally, we wish to compute that matching between pursuer coalition and evader, that i) maximizes the number of evaders caught and ii) maximizes the sum of weights (value functions) among all the possible coalitions of maximum cardinality. We also showcase a simulation framework developed for such 3D multiplayer pursuer evader games, using Python.

Problem Description

Multiplayer reach avoid differential games. Consider a reach-avoid differential game with $N_p + N_e$ players, where there are N_p pursuers $P = \{P_1, ..., P_{N_p}\}$ and N_e evaders $\varepsilon = \{E_1, ..., E_{N_e}\}$. The players are assumed to be mass points and they can move in any direction (holonomic). The game is played in the 3D Euclidean space R^3, where a plane τ divides the game space R^3 into two disjoint subregions Ω_{goal} and Ω_{play}. The mathematical descriptions of τ, Ω_{goal} and Ω_{play} are given by $\{\mathbf{x} \in R^3 | z = 0\}, \{\mathbf{x} \in R^3 | z \leq 0\}$ and $\{\mathbf{x} \in R^3 | z > 0\}$, respectively. Let $\mathbf{x}_{P_i} = \begin{bmatrix} x_{P_i}(t) & y_{P_i}(t) & z_{P_i}(t) \end{bmatrix}^T \in R^3$ be the positions of P_i and E_j at time t, respectively. The dynamics of the players are described by the system for $t \geq 0$:

$$\dot{\mathbf{x}}_{P_i}(t) = v_{P_i}\mathbf{u}_{P_i}(t), \quad \mathbf{x}_{P_i}(0) = \mathbf{x}^0_{P_i}, \quad P_i \in P$$
$$\dot{\mathbf{x}}_{E_j}(t) = v_{E_j}\mathbf{u}_{E_j}(t), \quad \mathbf{x}_{E_j}(0) = \mathbf{x}^0_{E_j}, \quad E_j \in \varepsilon$$

where $x^0_{P_i}$ and $x^0_{E_j}$ are the initial positions of P_i and E_j and the control inputs at time t for their headings is $u_{P_i}(t)$ and $u_{E_j}(t)$. So all players are allowed to change their orientations instantaneously. Suppose that the pursuer P_i has capture radius $r_i \geq 0$. The evader E_j is captured as soon as his distance from at least one of pursuers becomes equal to the corresponding capture radius. The capture set of the pursuit team is defined by $C := \cup_{i=1}^{Np} C_i$, where C_i is the capture set of pursuer P_i and is given by $\{x \in R^3 | \|x - x_{Pi}\|^2 \leq ri\}$. Assume that the number of pursuers remains constant, and the pursuers chase the evaders until all evaders in the play region Ω_{play} are captured.

The evasion team tries to send as many evaders to the goal region, or come as close to the goal region as possible, while the pursuers try to capture as many evaders as possible in the play region as far away from the goal.

Information structure and assumptions. Due to the adversarial nature of their goals, the pursuers and the evaders do not have access to the opponent's current input. Each player chooses its current input u_{P_i} or u_{E_j} based on the current value of the information set $\{x_{P_i}, x_{E_j}\}_{P_i \in P, E_j \in \varepsilon}$.

Assumption 2.1 (*Initial deployment*): The initial positions of all players satisfy the following four conditions:
(1) $\|x^0_{P_i} - x^0_{P_j}\|_2 > 0$ for all $P_i, P_j \in P, P_i \neq P_j$;
(2) $\|x^0_{E_i} - x^0_{E_j}\|_2 > 0$ for all $E_i, E_j \in \varepsilon, E_i \neq E_j$;
(3) $\|x^0_{E_j} - x^0_{P_i}\|_2 > r_i$ for all $P_i \in P$ and $E_j \in \varepsilon$;
(4) $x^0_{P_i} \in R^3$ for all $P_i \in P$ and $x^0_{E_j} \in \Omega_{play}$ for all $E_j \in \varepsilon$
We focus on heterogeneous players (even our simulation was developed for heterogeneous case), where different players have different speeds and capture radii.

Assumption 2.2 (*Speed Ratio*). Suppose the speed ratio $\alpha_{ij} = v_{P_i}/v_{E_j} > 1$ for all $P_i \in P$ and $E_j \in \varepsilon$.

Subgames of coalitions. It is hard to analyze the whole game directly, so we will decompose the whole game as many subgames which involve multiple pursuers and one evader. (1)

For any $s \in [\{1, 2, 3...N_p\}]^+$, let $P_s = \{P_i \in P | i \in s\}$ be an element of $[P]^+$, we refer to P_s as a pursuit coalition containing pursuer P_i if the subscript satisfies $i \in s$. States of all pursuers in s is stacked into x_s and u_s respectively. For the subgame between a pursuit coalition P_s and an evader E_j the winning condition is that P_s can capture E_j before the latter reaches Ω_{goal}; otherwise, the evader E_j wins. The capture means $x_{E_j} \in C_s$, where $C_s = \cup_{i \in s} C_i$.

Important Theory*

Definition (Evasion Space) Given any $P_s \in [P]^+$ and $E_j \in \varepsilon$, the evasion space (ES) E(s,j) is the set of positions in R^3, that E_j can reach without being captured by P_s, regardless of P_s control input, and let the surface $\partial E(s,j)$ which bounds $E(s,j)$ denote the boundary of evasion space(BES).

Definition (Potential Function) Given $\mathbf{x_{P_i}}$ and $\mathbf{x_{E_j}}$ satisfying $|\mathbf{x_{E_j}} - \mathbf{x_{P_i}}|_2 \geq r_i$, define the potential function $f_{ij}(x) : R^3 \to R$ with speed ratio α_{ij} as
$f_{ij}(x) = \|x - x_{P_i}\|_2 - \alpha_{ij}\|x - x_{E_j}\|_2 - r_i$
$E(i,j) = \{x \in R^3 | f_{ij} > 0\}$
$\partial E(i,j) = \{x \in R^3 | f_{ij} = 0\}$

Theorem 1: (2) Closure of ES(i,j) with respect to $P_i \in P$ and $E_j \in \varepsilon$ its closure is bounded and strictly convex.

Definition (Interception Point) if $\bar{E}(s,j) \cap \Omega_{goal}$ is empty, the interception point I(s,j) = $[x_{l(s,j)} \ y_{l(s,j)} \ z_{l(s,j)}]^T \in \bar{E}(s,j)$, be the closest point that is closest to Ω_{goal}

Theorem 2: (2) (Properties of Interception point). The interception point $I(s,j)$ has the following properties:
(i) $I(s,j)$ lies on $\partial E(s,j)$
(ii) for any s with |s| = 3, if E_j and the pursuers in P_s are not coplanar and $I(s,j) \in \cap_{i \in s} \partial E(i,j)$, then there exists a plane such that $I(s,j)$ is an intersection point of two strictly convex closed curves in the plane.
(iii) for any s with |s|=3, if E_j and the pursuers in P_s are not coplanar, then $\cap_{i \in s} \partial E(i,j)$ contains at most 4 intersection points.

Theorem 3: (2) (Degeneration of the Interception Point) For any $P_s \in [P]^+$ and $E_j \in \varepsilon$, suppose that $\bar{E}(s,j) \cap \Omega_{goal}$ is empty. Then, there must exist a pursuit sub coalition $s_1 \subset s$ such that $|s_1| \leq 3$ and $I(s_1,j) = I(s,j)$.

Theorem 4: (2) (ES-Based Strategies for Multiple Pursuers). For any $P_s \in [P]^+$ and $E_j \in \varepsilon$, suppose that $\bar{E}(s,j) \cap \Omega_{goal}$ is empty, and let s_1 be a subset of s such that $I(s_1,j) = I(s,j)$ and $|s_1| \leq 3$. If every pursuer P_i in P_{s_1} adopts the feedback strategy $\mathbf{u}_{P_i} = \frac{I(s_1,j) - \mathbf{x}_{P_i}}{\|I(s_1,j) - x_{E_j}\|_2}$, then the pursuit subcoalition P_{s_1} guarantees that $\bar{E}(s,j)$ does not approach Ω_{goal}, i.e, $\dot{z}_{I(s,j)} = 0$ if and only if E_j adopts the feedback strategy $\mathbf{u}_{E_j} = \frac{I(s_1,j) - \mathbf{x}_{E_j}}{\|I(s_1,j) - \mathbf{x}_{E_j}\|_2}$.

Theorem 5: (2) (Game of Kind). The game winner between a pursuit coalition P_s and an evader E_j can be determined as follows: If $\bar{E}(s,j) \cap \Omega_{goal}$ is empty, then the pursuit team P_s wins; if $\bar{E}(s,j) \cap \Omega_{goal}$ has more than one element, then E_j wins; if $\bar{E}(s,j) \cap \Omega_{goal}$ has a unique element then the two teams are tied. The maximum number of pursuers required to capture an evader before the evader reaches the goal region, is three.

Value functions and solving the HJI PDE

Theorem 6: (2) (Value Function). Consider the differential game, where $\bar{E}(s,j) \cap \Omega_{goal}$ is empty. For the states $(\mathbf{x}_s, \mathbf{x}_{E_j})$ such that the value function V$(\mathbf{x}_s, \mathbf{x}_{E_j})$ is differentiable, then V$(\mathbf{x}_s, \mathbf{x}_{E_j})$ can be computed by the convex optimization problem.
V$(\mathbf{x}_s, \mathbf{x}_{E_j})$ = minimize$_{x \in R^3}$ z
subject to $f_{ij} \geq 0, \forall i \in s$.

Matching strategies

Due to the theorem 3, our problem is simplified greatly, as we only need to consider all pursuit coalitions of size less than or equal to three. Number of possible vertices for P in the bipartite graph is $N_o = N_p * (N_p^2 + 5)/6$. Let $G = (U \cup V, E)$, be an undirected bipartite graph consisting of two independent vertex sets U and V, where E is the set of edges. The edge connecting vertex $P_s \in U$ and vertex $E_j \in V$ by e_{sj}. These edges e_{sj} also have a weight, which is $w_{sj} = V(\mathbf{x}_s, \mathbf{x}_{E_j})$. The vertex set U contains all pursuit coalitions of size no more than three, and V represents the set of evaders. The bipartite graph G is formally defined as:
$U = [P]^3, V = \varepsilon$
$E = \{e_{sj} | P_s \in U, E_j \in V, |\bar{E}(s,j) \cap \Omega_{goal}| \leq 1, \forall s_1 \nsubseteq s, |\bar{E}(s_1,j) \cap \Omega_{goal}| > 1\}$
This problem is a constrained maximum bipartite matching problem. The conflicts among the pursuit coalitions can be represented by a conflict graph C = $(E, \bar{E})$. Each vertex in C corresponds uniquely to an edge $e \in E$ of G. An edge $\bar{e} \in \bar{E}$ implies that the two vertices connected by $\bar{e}$ cannot coexist in the maximum matching of G. The conflict graph C may contain isolated vertices, which means that the corresponding edges in G do not conflict with others.

Formulation of the MBMC. Given the bipartite graph and the conflict graph, we define the binary integer programming (BIP) formulation for the MBMC as follows:
maximize $\Sigma_{e_{sj} \in E} \ a_{sj}$, first priority, then
maximize $\Sigma_{e_{sj} \in E} \ w_{sj}$
subject to $\Sigma_{s \in U} \ a_{sj} \leq 1, \ \forall E_j \in V,$
$\Sigma_{j \in V} \ a_{sj} \leq 1, \ \forall P_s \in U,$
$a_{sj} + a_{pq} \leq 1, \ \forall (e_{sj}, e_{pq}) \in \bar{E},$
$a_{sj} \in \{0,1\}, \ \forall e_{sj} \in E,$
$a_{sj} = 0, \ \forall e_{sj} \notin E,$
where $a_{sj} = 1$ indicates the assignment of pursuit coalition P_s to capture E_j and $a_{sj} = 0$ means no assignment, and w_{sj} refers to the weight given to each edge.
This MBMC is a NP-Hard problem (3). So, to solve it, we use an approximation, which is the sequential matching algorithm.

The Flow Algorithm (4) (5) . In order to maximize the cardinality of the matching while also maximizing the weight of the edges, we use the MaxCostMaxFlow algorithm at each step of the sequential matching.
We divide the matching into Coalitions of size 1 vs Evaders, size 2 vs Evaders and size 3 vs Evaders. After matching the

coalitions of size 1, we make sure to remove all the coalitions of size 2 and 3 which have pursuers that were matched before. Similarly after matching the coalitions of size 2, we remove those coalitions of size 3 which have pursuers that were matched before. The final match is obtained by finding the union of the individual matches, obtained in each step.

Algorithm 1: Sequential Matching Algorithm

Input: A bipartite graph $G = (U \cup V, E)$ with $U = [P]^3$ and $V = \varepsilon$, where $e_{sj} \in E$ if the pursuit coalition P_s in U can defeat the evader E_j in V.
Output: An approximation matching M in G

$U_1 \leftarrow P, V_1 \leftarrow \varepsilon, E_1 \leftarrow \{e_{sj} \in E | P_s \in U_1, E_j \in V_1\}$;
Compute the maximum matching M_1 in the subgraph $G_1 = (U_1 \cup V_1, E_1)$ by maximum cost maximum flow;
$A_1 \leftarrow \{P_i | i \in s, e_{sj} \in M_1\}, B_1 \leftarrow \{E_j | e_{sj} \in M_1\}$;
$U_2 \leftarrow [P \backslash A_1]^2, V_2 \leftarrow \varepsilon \backslash B_1$;
$E_2 \leftarrow \{e_{sj} \in E | P_s \in U_2, E_j \in V_2\}$;
Compute an approximation matching M_2 in the subgraph $G_2 = (U_2 \cup V_2, E_2)$ by maximum cost maximum flow again;
$A_2 \leftarrow \{P_i | i \in s, e_{sj} \in M_2\}, B_2 \leftarrow \{E_j | e_{sj} \in M_2\}$;
$U_3 \leftarrow [P \backslash (A_1 \cup A_2)]^3, V_3 \leftarrow \varepsilon \backslash (B_1 \cup B_2)$;
$E_3 \leftarrow \{e_{sj} \in E | P_s \in U_3, E_j \in V_3\}$;
Compute an approximation matching M_3 in the subgraph $G_3 = (U_3 \cup V_3, E_3)$ by maximum cost maximum flow once more;
Return $M = M_1 \cup M_2 \cup M_3$;

The Successive Shortest Path algorithm (SSP) works by finding the shortest path for a unit of "flow" to go from the source to the sink. Then it creates an augmented graph, where the remaining capacity is present(The initial flow is subtracted out). Potentials are assigned to the nodes, so that weights are non negative, and Dijkstra's algorithm can run without any issues.

The simulation framework

The simulation framework for the Multiplayer Pursuer-Evader 3D Differential game was created using Python and Matplotlib. You can find the github repository here. A video demonstration of the simulation is available here.
Some features of the simulation are as follows:
3D Visualization

- Real-time trajectory plotting for all agents
- Target region visualization (z=0 plane)
- Dynamic camera positioning.

Configurable Parameters

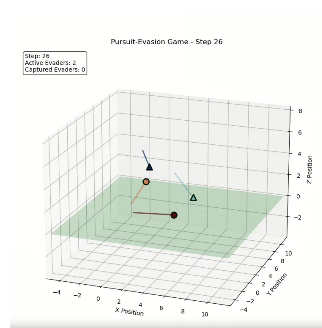
- Number of pursuers and evaders
- Agent speeds and capabilities
- Target region definition

Algorithm 2: Successive Shortest Path Algorithm for Maximum Cost Maximum Flow

Input: A directed graph $G = (V, E)$ where each edge $(u, v) \in E$ has capacity c_{uv} and cost w_{uv};
Source node $s \in V$, sink node $t \in V$.
Output: A feasible flow f achieving maximum cost maximum flow.

Initialize $f(u, v) \leftarrow 0$ for all $(u, v) \in E$;
Initialize node potentials $\pi(v) \leftarrow 0$ for all $v \in V$;
Construct the residual graph $G_f \leftarrow G$;
while *there exists a shortest s-t path in G_f under reduced costs* **do**
  Compute distances $d(v)$ and predecessors prev(v) for all $v \in V$ using Dijkstra's algorithm with reduced costs $c'_{uv} = w_{uv} - \pi(u) + \pi(v)$;
  if *no path from s to t exists* **then**
    break;
  end
  Update node potentials $\pi(v) \leftarrow \pi(v) + d(v)$ for all $v \in V$;
  Augment flow by 1 unit along the path defined by prev(v) from s to t;
  Update residual graph G_f capacities and reverse edges accordingly;
end
return *final flow* f;

- Capture radius setting
- Time step resolution
- Strategies followed by the different agents



The Simulation of a 2 pursuers vs 2 evaders game

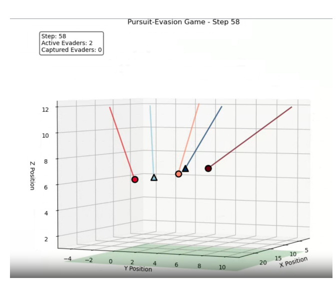


Fig. 1. The Simulation of a 3 pursuers vs 2 evaders game

References

1. Mo Chen, Zhengyuan Zhou, and Claire J. Tomlin. Multiplayer reach-avoid games via pairwise outcomes. *IEEE Transactions on Automatic Control*, 62:1451–1457, 2016.
2. Rui Yan, Xiaoming Duan, Zongying Shi, Yisheng Zhong, and Francesco Bullo. Matching-based capture strategies for 3d heterogeneous multiplayer reach-avoid differential games. *Automatica*, 140:110207, 2022. ISSN 0005-1098. doi: https://doi.org/10.1016/j.automatica.2022.110207.
3. Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972. ISBN 978-1-4684-2001-2. doi: 10.1007/978-1-4684-2001-2_9.
4. Lester Randolph Ford and Delbert Ray Fulkerson. Flows in networks. 2015.
5. Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. Network flows: Theory, algorithms, and applications. 1993.