Evasion from a Group of Pursuers with a Prescribed Target Set for the Evader

Jhanani Selvakumar Efstathios Bakolas

***Abstract*— We consider the problem of characterizing an evasion strategy for an agent under pursuit by multiple independent agents that are distributed in a convex polygon. The proposed evasion strategy is centrally dependent on a generalized Voronoi partition of the polygon, whose proximity metric is the time of capture of the evader by the nearest pursuer. Specifically, the boundaries of the Voronoi cells determine a set of preferable paths for the evader that will safely take it to a prescribed target set. The motion of the pursuers is accounted for by sequential re-partitioning and re-planning. A novel method for the computation of the generalized Voronoi partition, which exploits the structure of the solution to a two-player differential game, is presented. It is shown that the proposed method is significantly faster (by at least one order of magnitude) than two known methods in the literature; this expedites computation of the proposed evasion strategy. Numerical simulations that illustrate the effectiveness of the proposed evasion strategy are presented.**

I. Introduction

Evasion of a single agent from a group of pursuers is a phenomenon found in a number of natural and man-made environments. The pursuers may be behaving cooperatively or engaged in individual/relay pursuit. In this paper, we are concerned with the successful evasion of a single agent from a non-cooperative group of agents, resulting in a multi-player non-zero sum game. Limiting the game to a convex polygonal domain, we propose a strategy for evasion where the evader stays close to the boundaries of the Voronoi cells of the pursuers. The proximity metric of the Voronoi paritioning problem is non-Euclidean and discontinuous. The evader seeks to reach the goal edge (target set) of the convex polygon unharmed, if possible (goal-oriented evasion). We assume perfect information for all players.

Literature survey: Rufus Isaacs in [1] analyzed Pursuit Evasion Games (PEGs) and focused on the characterization of a saddle point solution for zero-sum games with two players. Alternative techniques to solve for the optimal strategy in PEGs, including variational methods and methods to compute stroboscopic strategies are found in [2]–[5]. The existence of saddle points and Nash Equilibria for non-zero sum games with multiple players is examined in [6], [7]. Recent literature on cooperative multiple agent PEGs is rich. For instance, cooperative defense of one or more agents against a single pursuer is studied in [8]–[11]. A probabilistic framework is used to model a multiple pursuer - multiple evader PEG in [12]. A number of heuristic pursuit strategies are compared in [13] for a multi-player PEG with incomplete information. In [14], a group of pursuers aided by a sensor network coordinate and minimize time of capture of multiple evaders. Non-cooperative/decentralized pursuit of a single evader by multiple pursuers is discussed in [15], [16]. A switching strategy using Voronoi partitions for multiple agents pursuing a single evader is presented in [17]–[19]. In [20], the evader traverses a convex polygon while maximizing the time of capture by any of the pursuing agents. To achieve this, [20] proposes a roadmap evasion policy where the evader travels on the boundaries of the Voronoi cells generated by the pursuers' positions. However, the work in [20] assumes that only one pursuer goes after the evader whereas the other pursuers are static (this problem can essentially be addressed in one step). Further, [20] suggests methods in literature such as the Direct Diffusion algorithm [21] to quickly obtain the generalized Voronoi partition of a given space for any given proximity metric. However, Direct Diffusion does not correctly calculate Voronoi-like partitions composed of disconnected cells as in our case.

Main contributions: In this paper, we present an algorithm for effective evasion from moving pursuers under the assumption of complete capturability of the evader. The evader's path is determined by standard graph search algorithms using a cost function that is highly tuned for goal-oriented evasion. We propose an algorithm based on expanding isochrones to quickly and incrementally generate a non-Euclidean Voronoi partition of the convex domain. Our method is faster than the approach in [21], while also generating the correct disconnected Voronoi cells for our proximity metric. Our algorithm also facilitates easy extraction of the roadmap from the Voronoi partition, which is used to generate a feasible route through the domain. Finally, we contrast our evasion policy with a simple policy and show that our approach is highly effective in a large number of cases.

Structure of the paper: Section II introduces and formulates the evasion problem. In Section III, the algorithm of expanding isochrones is presented along with our evasion algorithm. Comparison with other algorithms via numerical simulations are provided in Section IV. Concluding remarks are presented in Section V.

II. Formulation of the evasion problem

In this section, we first state briefly the group pursuit-evasion problem, which has been described in detail in [20]. Let us consider a group of n pursuers, within a convex polygon $\mathcal{S}$ in $\mathbb{R}^2$. The boundary of $\mathcal{S}$ is denoted by $\partial\mathcal{S}$, and consists of m edges, $\mathcal{E}_j, j \in \{1, 2, \ldots, m\}$. At any instant

J.Selvakumar is a graduate student at the Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712-1221, USA, Email: jhanani@utexas.edu

E. Bakolas is an Assistant Professor at the Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712-1221, USA, Email: bakolas@austin.utexas.edu

of time t, the position of the i^{th} pursuer is denoted by $\boldsymbol{x}_i$ and its velocity by $\boldsymbol{v}_i$, and the index $i \in \mathcal{I} := \{1, 2, \ldots, n\}$. The motion of the i^{th} pursuer is governed by the following equations:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{v}_i, \quad \dot{\boldsymbol{v}}_i = F\boldsymbol{u}_i - k\boldsymbol{v}_i, \tag{1}$$

with initial conditions $\boldsymbol{x}_i(0) = \bar{\boldsymbol{x}}_i$ and $\boldsymbol{v}_i(0) = \bar{\boldsymbol{v}}_i$. The acceleration $F > 0$ is a constant, the coefficient of friction is $k > 0$, and $\boldsymbol{u}_i$ is the unit vector which acts as the input. It is to be noted here that the limiting velocity of each pursuer is given by F/k. The position of the single evader at time t is denoted by $\boldsymbol{x}_e$ and its velocity by $\boldsymbol{v}_e$. The evader follows simple dynamics:

$$\dot{\boldsymbol{x}}_e = w\boldsymbol{u}_e, \quad \boldsymbol{x}_e(0) = \bar{\boldsymbol{x}}_e, \tag{2}$$

where $\boldsymbol{u}_e$ is a unit vector that is the evader's directional input. The evader's speed w is a constant. Capture is defined by positional proximity to a user-specified tolerance $l > 0$.

Problem 1 (Goal-oriented evasion problem): There are n pursuers, at known initial locations inside a convex polygon $\mathcal{S} \subset \mathbb{R}^2$. The evader's initial position is on an edge $\mathcal{E}_s,\ s \in \{1, 2, \ldots, m\}$ of $\mathcal{S}$. Find a sequence of control inputs $\boldsymbol{u}_e$ that will steer the evader to the specified goal edge $\mathcal{E}_g, g \neq s$, while it is true that $\min_{i\in\mathcal{I}} \|\boldsymbol{x}_i - \boldsymbol{x}_e\| > l$ for all $t \in [0, T_c]$, where $T_c > 0$ denotes the time at which the evader reaches $\mathcal{E}_g$ for the first time.

Group of pursuers: The pursuers are in constant motion and engage in relay pursuit [18]. Each pursuer operates independently, and tries to minimize its own time of capture. Let us consider the i^{th} PEG, that is, the zero-sum game involving the i^{th} pursuer and the evader alone. It is noteworthy here that we assume the evader can be captured in finite time regardless of its initial position inside $\mathcal{S}$. This makes the problem of successful evasion non-trivial from the evader's perspective. A complete analysis of the two-player game within the capturability assumption is provided in [1], wherein it is referred to as the *isotropic rocket* game. When the condition for capture is satisfied, the value function $T_{\mathrm{sp}}(\boldsymbol{x}_i, \boldsymbol{v}_i, \boldsymbol{x}_e)$ of the i^{th} PEG is the least positive root of the equation

$$\boldsymbol{r}_i^2 - 2\langle \boldsymbol{r}_i, \boldsymbol{v}_i\rangle\eta + \boldsymbol{v}_i^2\eta^2 = \mathrm{Q}^2(T), \tag{3}$$

where $\boldsymbol{r}_i := \boldsymbol{x}_e - \boldsymbol{x}_i$, $\eta := (1 - e^{-kT})/k$, and $\mathrm{Q}(T) := l - wT + F\left(T - \eta\right)/k$. The set of positive real solutions to (3) is denoted by $T_{\mathrm{sol}}(\boldsymbol{x}_i, \boldsymbol{v}_i, \boldsymbol{x}_e)$, and $T_{\mathrm{sp}}(\boldsymbol{x}_i, \boldsymbol{v}_i, \boldsymbol{x}_e)$ is the minimum value in $T_{\mathrm{sol}}(\boldsymbol{x}_i, \boldsymbol{v}_i, \boldsymbol{x}_e)$. The value function is also the time of capture in the i^{th} two-player PEG. Now, we define the lower envelope function $T^*_{\mathrm{sp}}(\boldsymbol{x}_e)$ as follows:

$$T^*_{\mathrm{sp}}(\boldsymbol{x}_e) := \min_{\mathcal{I}} T_{\mathrm{sp}}(\boldsymbol{x}_i, \boldsymbol{v}_i, \boldsymbol{x}_e) = T_{\mathrm{sp}}(\boldsymbol{x}_{i_p}, \boldsymbol{v}_{i_p}, \boldsymbol{x}_e), \tag{4}$$

where i_p is the index of the nearest[1] pursuer (defined by T^*_{sp}), at any time t. This pursuer is the *active* pursuer. At each instant of time, only the active pursuer proceeds to capture the evader. The other pursuers passively maintain a localized periodic orbit. We assume here that the active pursuer will only attempt to capture the evader if $T^*_{\mathrm{sp}}(\boldsymbol{x}_e) < T_{\max}$ where $T_{\max}$ is a chosen threshold. Beyond this threshold, the pursuer will be indifferent to the presence of the evader. This assumption is relevant to situations where for each pursuer, there is a trade-off between capturing the evader and patrolling its local area in $\mathcal{S}$. Analysis of the multi-player game as a whole requires extensive computation, and is possibly intractable. By assuming the model of a relay pursuit as suggested in [18], at any instant of time t, we concern ourselves only with the computation of the lower envelope function, as defined in (4). Since the pursuers act independently, the PEG described in this paper can be approximated to n concurrent two-player *isotropic rocket* PEGs. It is to be noted that the $n + 1$ player PEG is not zero-sum as in the two-player case. However, at each time step, the active pursuer acts as if it were engaged in a two-player zero-sum game with the evader.

The evader: The evader has a two-fold mission: (1) evade from each pursuer successfully, (2) reach the goal edge $\mathcal{E}_g$ in minimum time. The control choice made by the evader at every instance reflects a trade-off between the two aspects of the mission. This means that the evader may engage in sub-optimal play with respect to the active pursuer, as the optimal evasive action might not steer it towards the goal. The evasion policy presented in this paper relies significantly on a Voronoi-like partition that is generated by the pursuers' positions using the lower envelope function defined in (4) as the proximity metric. The Generalized Voronoi Diagram (GVD) is denoted by $\mathfrak{V}$, which is defined as $\mathfrak{V} := \{\mathfrak{V}_i,\ i \in \mathcal{I}\}$, where

$$\mathfrak{V}_i := \{\boldsymbol{z} \in \mathcal{S} :\ T_{\mathrm{sp}}(\boldsymbol{x}_i, \boldsymbol{v}_i, \boldsymbol{z}) = T^*_{\mathrm{sp}}(\boldsymbol{z})\}. \tag{5}$$

The cell $\mathfrak{V}_i$ consists of all evader starting points in $\mathcal{S}$ that result in capture by the i^{th} pursuer before all others. It is the region of dominance of the i^{th} pursuer within $\mathcal{S}$.

III. Solution to evasion problem

In this section, we discuss the solution to the goal-oriented evasion problem. In particular, the solution comprises two steps: (i) obtaining the GVD with (4) as the metric, and (ii) finding a feasible path through the domain of interest using the GVD and graph search methods. The notion of a feasible path will be explained later. The computation of the Voronoi partition and the subsequent graph search are described in this section.

A. Partitioning algorithm

Straightforward partitioning algorithms construct approximations of the desired partitions by utilizing a discretization grid, say $\mathcal{G}$, over the space to be partitioned. Some characteristic examples are the exhaustive classification and *Direct Diffusion* algorithm [21]. We use an alternative approach to construct the GVD from level sets of the lower envelope function associated with each generator. We call this the "method of isochrones."

1) Exhaustive classification and Direct Diffusion: Using the lower envelope function $T^*_{\mathrm{sp}}(\cdot)$ as the proximity metric, each grid point of the discretized space is individually assigned to the nearest pursuer. Due to the fact that the proximity metric is discontinuous, we get a Voronoi partition

[1]If the argmin is not unique, we pick the lowest index

whose cell boundaries are intricate at certain locations. The grid employed must capture these geometric details.

2) Method of isochrones: In this paper, we propose a partitioning algorithm in which the GVD is composed as an image using overlapping isochrones. The isochrone corresponding to a pursuer for a particular time T is the spatial boundary of points the pursuer can "reach" in time $[0, T]$. For the time of capture metric discussed in this paper, the boundary is a circle with a moving center and time-varying radius, as we show next. The isochrone curves in $\mathbb{R}^2$ are obtained by substituting specific values for time in (3). For the i^{th} pursuer, initially given $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$, let the isochrone at time $T > 0$ be denoted by $\mathcal{C}_i^T$ and be defined as $\mathcal{C}_i^T := \{\boldsymbol{x}_e \in \mathbb{R}^2 : \ T \in T_{\text{sol}}(\boldsymbol{x}_i, \boldsymbol{v}_i, \boldsymbol{x}_e)\}$. Note that isochrones may intersect with each other. The level set at time T, denoted by $\mathcal{L}_i^T$, is defined as $\mathcal{L}_i^T := \{\boldsymbol{x}_e \in \mathbb{R}^2 : \ T_{\text{sp}}(\boldsymbol{x}_i, \boldsymbol{v}_i, \boldsymbol{x}_e) = T\}$. Note that $\mathcal{L}_i^T \subseteq \mathcal{C}_i^T$ and $\mathcal{L}_i^T \cap \{\cup_{0 \le t < T} \mathcal{C}_i^t\} = \varnothing$.

Proposition 1: The isochrone at time $t = T$ that is associated with the i^{th} agent, $\mathcal{C}_i^T$, is a circle with radius $R_i(T) = \mathrm{Q}(T)$ that is centered at the point $\boldsymbol{c}_i(T) := \boldsymbol{x}_i + \eta \boldsymbol{v}_i$.

Proof: Let $\boldsymbol{r}_i := \boldsymbol{x}_e - \boldsymbol{x}_i$. From (3), adding $\|\boldsymbol{x}_i + \eta \boldsymbol{v}_i\|^2$ on both sides, we have for any $\boldsymbol{x}_e \in \mathcal{C}_i^T$,

$$\|\boldsymbol{x}_e\|^2 - 2\langle \boldsymbol{x}_e, \boldsymbol{x}_i + \eta \boldsymbol{v}_i \rangle + \|\boldsymbol{c}_i(T)\|^2 = \mathrm{Q}^2(T) - \eta^2 \|\boldsymbol{v}_i\|^2 - 2\langle \boldsymbol{x}_i, \eta \boldsymbol{v}_i \rangle - \|\boldsymbol{x}_i\|^2 + \|\boldsymbol{c}_i(T)\|^2, \quad (6)$$

where $\boldsymbol{c}_i(T) := \boldsymbol{x}_i + \eta \boldsymbol{v}_i$. After simplifying (6) and collecting terms, we have that the equation for the isochrone at time T is

$$\|\boldsymbol{x}_e - \boldsymbol{c}_i(T)\| = \sqrt{\mathrm{Q}^2(T)}. \quad (7)$$

Because the condition for capturability in the isotropic rocket PEG is satisfied, we have that $\mathrm{Q}(T)$ is real and non-negative [1], which in turn implies that $R_i(T) = \mathrm{Q}(T)$. ■

The GVD is composed by expanding the overlapping isochrones backwards in time, starting from $T = T_{\max}$ until $T = 0$. For each pursuer, the boundary points of the current isochrone are calculated by explicitly substituting the value of T in (6) and (7). This guarantees the correctness of the isochrones. The intersections of the isochrones corresponding to different agents are the boundaries of the Voronoi cells. By expanding backward in time, at each step, we eliminate the need to keep track of points which were covered by previous isochrones. The composition of the image is done by superimposing (visual stacking [22]) new isochronic circles on top of the image of previously expanded isochrones. The absence of explicit traversal through a grid makes this method much faster than exhaustive classification or Direct Diffusion, as we will see in Section IV.

input : $T_{\max}$, n, δT, $\boldsymbol{x}_i$, $\boldsymbol{v}_i$ $\mathcal{S}$
output: A composed partition of $\mathcal{S}$

Preliminary steps
$T = T_{\max}$;
Create blank image M of $\mathcal{S}$;

while $T \geq 0$ **do**
  for $i \leftarrow 1$ **to** n **do**
    Render the isochrone $\mathcal{C}_i^T$ on the existing image M;
    Update new position of pursuer $\boldsymbol{c}_i$;
  end
  $T \rightarrow T - \delta T$;
end

Algorithm 1: Method of Isochrones

B. Feasible path to goal

The graph search for a feasible path for the evader through $\mathcal{S}$ is based on the GVD $\mathfrak{V}$. The precise cost metric used for the graph search dictates the behavior of the evader. In this context, we define a "pessimistic" time of capture based only on Euclidean distance from the active pursuer.

1) Pessimistic time of capture: The pessimistic time of capture, which is denoted by $T_p(\boldsymbol{x}_e)$, is defined as $T_p(\boldsymbol{x}_e) := \|\boldsymbol{x}_e - \boldsymbol{x}_{i_p}\|/(F/k)$, where i_p is the index of the active pursuer as defined in (4). From the perspective of achieving evasion, this is a more conservative estimate of the time of capture as $T_p(\boldsymbol{x}_e) \leq T^*_{\text{sp}}(\boldsymbol{x}_e)$, for all $\boldsymbol{x}_e \in \mathcal{S}$.
The *risk of capture* associated with a location on the map is the inverse of the least time of capture for the evader at that location. Hence, the risk of capture based on the pessimistic time of capture is defined as: $R_{cp}(\boldsymbol{x}_e) := 1/(T_p(\boldsymbol{x}_e))^2$.

2) Roadmap from GVD: Let $\partial\mathfrak{V}$ denote the set of all the boundary points of each Voronoi cell that lie in the interior of $\mathcal{S}$, that is, $\partial\mathfrak{V} := (\cup_{\mathcal{I}} \partial\mathfrak{V}_i) \backslash \partial\mathcal{S}$. The evading roadmap Γ associated with the generalized Voronoi diagram is the set of all continuous paths whose traces are in $\partial\mathfrak{V}$. On the output image from Algorithm 1, we use gradient-based edge extraction [22] to obtain the boundaries of the Voronoi cells and thus the set $\partial\mathfrak{V}$. The pixel grid is treated as an equivalent uniform Cartesian spatial grid. For a given origin and destination in $\partial\mathcal{S}$, the objective is to determine the optimal path γ_l among all the continuous paths in $\mathcal{S}$ that lead from the initial edge to the goal edge. Optimality is determined based on the cost function defined below.

3) Cost function and parameters for graph search: All the grid points including $\partial\mathfrak{V}$ are then represented as vertices of a directed graph $G := (V, E)$. V is the set of vertices (nodes) and E is the set of edges. Each element (u, v) in E indicates a transition from one node u to another node v on the graph. A cost function $\mathscr{C}(\cdot)$ is defined for each edge (u, v) in E. The cost incorporates two factors: (1) the risk of capture at node v given by $\mathscr{J}(v)$, and (2) the time to the goal node, τ_g. Let $\mathrm{node}(\cdot)$ be a function that takes in a pair of spatial coordinates and returns the corresponding node ID in V. The inverse of this function is denoted by $\mathrm{node}^{-1}(\cdot)$. Let u^0 and u^l be the start and goal nodes respectively. An edge cost from node $u \in V$ to node $v \in V$ is assigned as shown in Algorithm 2.

For nodes v that are not covered by the Voronoi regions of any generators, the risk of capture is zero. This is due to the upper limit $T_{\max}$ on the time of capture for the pursuers. For points that are in the set $\partial\mathfrak{V}$ or have neighboring grid points in $\partial\mathfrak{V}$, we scale the risk of capture by a factor α or β, where $0 < \alpha \leq \beta \leq 1$. The total cost associated with each edge also includes a penalty on the time to cover the Euclidean distance to the goal node. The weight ξ for each node u in the set $\mathfrak{V}$ is given by $\xi = \exp\left(-\|\mathrm{node}^{-1}(u) - \boldsymbol{x}_{i_p}\|\right)$. For other nodes, $\xi = 0$. The value of the weight ξ represents the priority given

input : u, v, w, $\partial\mathfrak{V}$, α, β
output: Cost of transition $\mathscr{C}$ from u to v

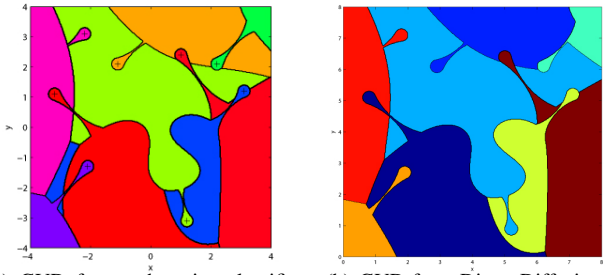
$\tau_g = \|\text{node}^{-1}(u^l) - \text{node}^{-1}(v)\|/w$;
if $v \in \partial\mathfrak{V}$ **then**
  $\mathscr{J}(v) = \alpha R_{cp}(\text{node}^{-1}(v))$;
  $\mathscr{C}(u,v) = \xi\mathscr{J}(v) + (1-\xi)\tau_g^2$;
else
  p:=any neighbor node of v other than u
  if $p \in \partial\mathfrak{V}$ **then**
    $\mathscr{J}(v) = \beta R_{cp}(\text{node}^{-1}(v))$;
    $\mathscr{C}(u,v) = \xi\mathscr{J}(v) + (1-\xi)\tau_g^2$;
  else
    $\mathscr{J}(v) = R_{cp}(\text{node}^{-1}(v))$;
    $\mathscr{C}(u,v) = \xi\mathscr{J}(v) + (1-\xi)\tau_g^2$;
  end
end

Algorithm 2: Cost assignment for graph search

to evading capture, while the value $(1-\xi)$ is the priority for reaching the goal. A graph search algorithm is used on the directed graph G to get the path from the start node to the goal node. In this paper, A* search has been used (the heuristic part of the cost is chosen so it is admissible). The path yielded by A* is given as $\mathscr{P} = \{u^0, u^1, ..., u^l\}$, where $l+1$ is the number of nodes in the path. The corresponding path in spatial coordinates is γ_l. The parameters α and β determine the adherence of the path to the generated roadmap.

4) Graph search: Cost assignment is done as described in Algorithm 2, with $\beta \ll 1$ and $\alpha \ll \beta$. While the roadmap is the preferred set of paths, A* calculates a route through the interior of the Voronoi cells if the goal node is not reachable through the roadmap. This guarantees completeness of the graph search, despite possibly incurring high cost. This has some important implications on the outcome of the evasion problem, as will be shown in Section IV.

5) Evasion based on updated GVD: The GVD changes due to the pursuers' movement. The cell boundaries move and reshape in a possibly discontinuous manner, owing to the proximity metric $T^*_{\text{sp}}(\cdot)$. A new GVD of $\mathcal{S}$ is generated at each stage, and the graph search algorithm is applied to determine a path from the current position of the evader to the goal edge. However, replanning the GVD at every time step could produce graphs which cause the evader to endlessly loop around in the position space. This in turn means that the global path search algorithm is not complete. Completeness is ensured by introducing a decision routine. Let the current position of the evader be $\boldsymbol{x}_e \in \mathcal{S}$, and $\text{dist}(\boldsymbol{z}, \mathcal{E}_g)$ be the shortest distance of a point $\boldsymbol{z}$ from the goal edge. In addition, let $\mathfrak{R}$ be the set of rejected path coordinates, defined as $\mathfrak{R} := \{\boldsymbol{z} \in \gamma_l : \text{dist}(\boldsymbol{z}, \mathcal{E}_g) \geq \text{dist}(\boldsymbol{x}_e, \mathcal{E}_g)\}$. The next "best" spatial coordinates for the evader are given by the first element of γ_l', where $\gamma_l' = \gamma_l \backslash \mathfrak{R}$. The argument for the completeness of the evasion algorithm is as follows. Suppose there exist graphs G and G' where the "best" position of the evader is at the nodes $u \in G$ and $u' \in G'$. Let G' be generated after replanning from G, and vice-versa. If either u or u' is on the goal edge, the A* algorithm will terminate, followed by the termination of the replanning routine. If neither u or u' is on the goal edge, the evader will alternate between positions $\text{node}^{-1}(u)$ and $\text{node}^{-1}(u')$ without termination. Let $\text{dist}(\text{node}^{-1}(u'), \mathcal{E}_g) \geq \text{dist}(\text{node}^{-1}(u), \mathcal{E}_g)$. Then, the decision routine applied to the optimal path of graph G will ensure that the node u' is removed from the path. Then, when the graph G' is generated subsequently, the evader will be in a new node $v \in G'$ that is different from u', such that $\text{dist}(\text{node}^{-1}(v), \mathcal{E}_g) < \text{dist}(\text{node}^{-1}(u), \mathcal{E}_g)$. Hence the alteration between u and u' is not possible when the decision routine is implemented. This routine ensures that the evader's distance to the goal monotonically decreases. The algorithm terminates automatically when the goal edge is reached.



(a) GVD from exhaustive classification, taken from [20]

(b) GVD from Direct Diffusion.

Fig. 1. GVD for a scenario with eight pursuers. It can been seen in (b) that a yellow region in the lower left, and a blue region in the upper right, are missing compared to (a).

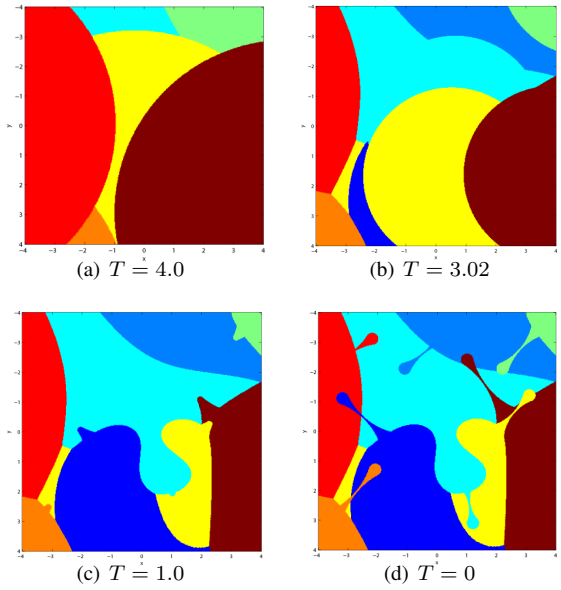
IV. Numerical Simulations and Comparison

We consider a scenario where the number of pursuers $n = 8$, and use the following data: $w = 0.6, F = 1, l = 0.2, k = 0$ and $\mathcal{S} = [-4, 4] \times [-4, 4]$. For the case of $k = 0$, the proximity metric can be obtained by solving a quartic equation [1], which is simpler than the transcendental equation in (3).

A. GVD generation

The partitioning of the convex polygon $\mathcal{S}$ using $T^*_{\text{sp}}(\cdot)$ as the proximity metric has been attempted using three methods: (i) Exhaustive classification of grid points, (ii) Classification of grid points by Direct Diffusion, and (iii) Composition of the GVD using expanding isochrones. The partitioning obtained from exhaustive classification for one particular set of initial conditions is shown in Fig. 1(a). The corresponding GVD from Direct Diffusion is illustrated in Fig. 1(b). The time complexity of Direct Diffusion when generators are sufficiently far apart is $\mathcal{O}(|\mathcal{G}|)$, where $|\mathcal{G}|$ is the number of grid nodes. Exhaustive classification has time complexity in $\mathcal{O}(n|\mathcal{G}|)$. Thus, Direct Diffusion, while faster [2] than exhaustive classification, is unable to properly identify and obtain disconnected partitions [21], should they exist. In our case, the disconnected partitions exist due to the discontinuous proximity metric. Fig. 2 shows stages of evolution of the GVD using the method of isochrones. The envelopes belonging to different agents are differentiated by color. All points inside the envelope of each pursuer are designated to it. Fig. 2(d) shows the composed image at $T_s = 0$. The method of isochrones is an order of magnitude faster

[2]Worst case complexity when generators are too close is same as exhaustive classification [21].



(a) $T = 4.0$ (b) $T = 3.02$ (c) $T = 1.0$ (d) $T = 0$

Fig. 2. Incremental computation of the GVD using method of isochrones, shown at different times T where $T \in [0, 4]$.

TABLE I

PARAMETERS OF DISTRIBUTION FOR INITIAL CONDITIONS

Variable	x	y	v	θ
μ	{1,6.5}	{7,1.1}	0.75	$\frac{5\pi}{6}$
σ^2	{0.5,1}	{1,0.5}	1	1

than Direct Diffusion and is independent of the placement of generators inside $\mathcal{S}$. In particular, the time complexity of this approach turns out to be in $\mathcal{O}(n)$. The simulations were carried out for sets of different number of pursuers, with different initial conditions. The initial conditions (position and velocity) for the pursuers were generated by random sampling from different normal distributions on the area of interest. A set of parameters used for the simulations in this paper are given in Table I. The graph in Fig. 3 illustrates a comparison of the time taken to generate the GVD in the case of the exhaustive classification, the Direct Diffusion method and the method of isochrones with $\delta T = 0.05$ and $T_{\max} = 10$. In all cases, $|\mathcal{G}| = 500$.

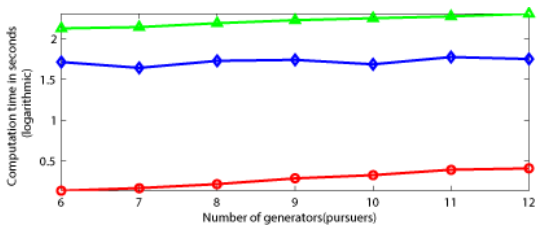


Fig. 3. Time-comparison for generation of GVD. Blue: Direct Diffusion, Green: exhaustive classification, Red: method of isochrones. The computations were performed on a platform with 8 GB RAM and 3.20 GHz processor.

The determination of the optimal path is a recurrent step in our evasion algorithm, as the GVD is updated. The method of isochrones generates the GVD faster than the other methods and consequently, the time to find the optimal path through the roadmap is the least using that method.

B. Goal-oriented evasion from a group of pursuers

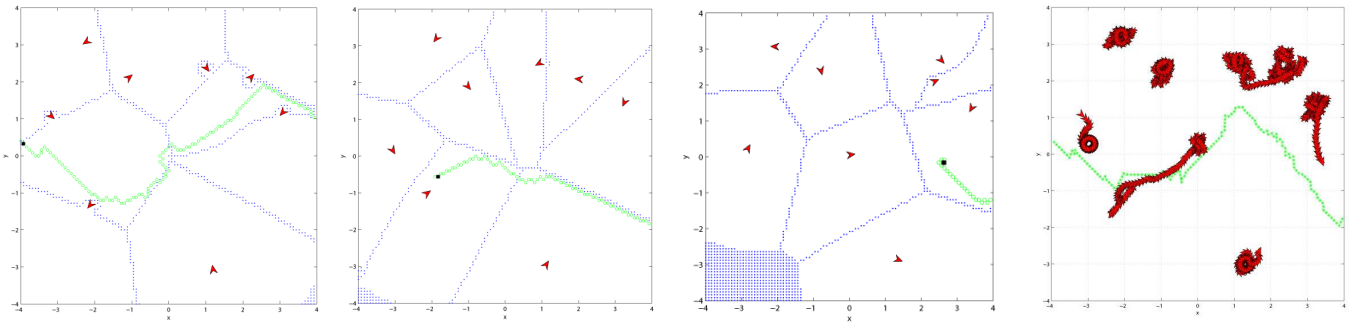
We view the PEG as a series of discrete moves, with the players moving simultaneously. For this reason, it is necessary to obtain a discretized model of the continuous system in (1) to update the pursuer positions. In this paper, evasion is accomplished by replanning at each time step using current positions of all players to generate a new GVD. In turn, this provides a new roadmap and a new optimal path to the goal edge $\mathcal{E}_g$ for the evader from its current position.

1) Roadmap evasion: The discontinuous value function in the isotropic rocket PEG reflects the "swerving" capability of the evader, which is a maneuvering advantage over the pursuers. The pursuers, which follow double integrator dynamics, cannot change direction instantaneously. For the following example, we set $\alpha = 0.01$ and $\beta = 0.25$. The path γ_l is optimal with respect to the current position of the pursuers. However, in continuous global replanning, optimality may not be preserved between two stages. Hence, the notion of an optimal path is replaced by that of a feasible path (globally) for the PEG. The globally feasible path γ_L is obtained by concatenation of segments of locally optimal paths γ_l generated by A* search at each update. Different stages of evasion using updated GVD for a scenario of eight pursuers are illustrated in Fig. 4. The set of grid points that do not belong to $\mathfrak{V}$ constitute the free regions within $\mathcal{S}$ as seen in Fig. 4(c). The free regions exist because there is a limit $T_{\max}$ on the capture time for the pursuers.

2) Comparison of replanning evasion with naive evasion: In this section, the proposed evasion policy (using the updated roadmap) is contrasted with a naive policy where the evader does not make use of the Voronoi partition.

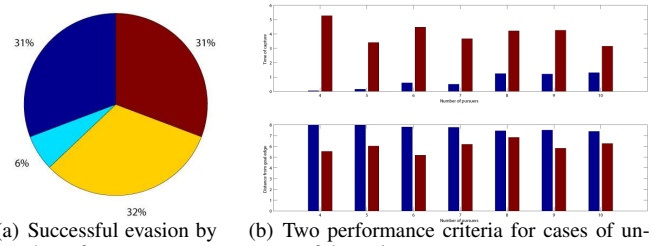
Naive evasion policy: In the naive policy, the evader progresses in a straight line trajectory towards the goal edge $\mathcal{E}_g$ as long as there is no pursuer in close proximity. The threshold of proximity is fixed as 5 times that of the capture radius l. While there is a pursuer of index i such that $\|\boldsymbol{x}_i - \boldsymbol{x}_e\| < 5l$, the evader will employ the optimal evasive action [1] for the two-person zero-sum PEG between the i^{th} pursuer and the evader. If there is more than one pursuer within the proximity threshold, the evader takes the optimal action with respect to the closest pursuer. Once no pursuer is within the threshold, the evader continues in its straight line path to the goal. For the simulations comparing evasion policies, we use the following parameter values: $F = 3$, $k = 1$, $w = 1$, $l = 0.2, T_{\max} = 3$. The target set (goal edge) is $\mathcal{E}_g = \{\boldsymbol{z} \in \mathcal{S},\ \boldsymbol{z} = [4\ z_2]^{\mathrm{T}}\}$. A large number of simulations (of order $\sim 10^3$) were performed, and the two policies were applied to the same initial conditions in each case. The pie graph in Fig. 5(a) shows a comparison between the two policies, in terms of number of successful evasions (where the evader reached the goal edge without being captured). It is clear that evasion based on the updated GVD is significantly more effective than the naive policy.

However, in the cases where evasion fails, our policy is comparable to the naive policy, in terms of two metrics of interest: (1) time of capture (2) distance from goal edge. It is notable at this point that the results obtained from the updated GVD evasion policy are significantly affected by



(a) $T = 0$ (b) $T = 3.5$ (c) $T = 11$ (d) Full path

Fig. 4. Evader movement using updated GVD at different times. The black dot represents the position of the evader, the red markers are the pursuers and the Voronoi boundaries and free regions within $\mathcal{S}$ are represented by blue dots. The path determined at each step is also shown in green. The complete path is shown in (d).



(a) Successful evasion by number of cases.

(b) Two performance criteria for cases of unsuccessful evasion.

Fig. 5. Comparison of updated GVD strategy with the naive strategy. In (a), Deep blue: Updated GVD evasion, Light blue: Naive evasion, Yellow: Both, Red: Neither. In (b), Blue: Updated GVD evasion, Red: Naive evasion,

the choice of parameters in the cost function, in particular, ξ. For unsuccessful cases, the updated GVD policy performs poorly on the time of capture and distance travelled. This is illustrated in Fig. 5(b).

V. Conclusion

We have presented an effective evasion strategy in a multi-player PEG involving an evader and a group of pursuers inside a convex polygon. The evader aims to reach a goal edge of the polygon while delaying or avoiding capture. Our evasion policy is based on a particular type of generalized Voronoi partition of the convex polygon, which is computed quickly and accurately by the method of isochrones presented herein. Also, the proposed solution method, which uses graph search and updated GVDs, offers an attractive alternative to directly solving the multi-player PEG, which is a complex process, if tractable at all. We will advance this work further by utilizing improved replanning algorithms like the Lifelong planning A*, or D*, instead of replanning from scratch at every step. We will also consider games in higher dimensions and in non-convex domains (to account for instance, for the presence of obstacles) or with higher fidelity kinematic models for the players.

References

[1] R. Isaacs, *Differential Games. A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. New York: Dover Publication, 1999.

[2] O. Hàjek, *Pursuit Games: An Introduction to the Theory and Applications of Differential Games of Pursuit and Evasion*. Mineola, New York: Dover Publications, second ed., 2008.

[3] S. Gutman, M. Esh, and M. Gefen, "Simple linear pursuit-evasion games," *Computers & Mathematics with Applications*, vol. 13, no. 1, pp. 83–95, 1987.

[4] P. Bernhard, "Linear-quadratic, two-person, zero-sum differential games: necessary and sufficient conditions," *J. Optim. Theory Appl.*, vol. 27, no. 1, pp. 51–69, 1979.

[5] L. D. Berkovitz, "A variational approach to differential games," *Advances in Game Theory*, no. 52, pp. 127–174, 1964.

[6] A. W. Starr and Y.-C. Ho, "Nonzero-sum differential games," *J. Optim. Theory Appl.*, vol. 3, no. 3, pp. 184–206, 1969.

[7] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. London: SIAM, 1995.

[8] W. Scott and N. Leonard, "Pursuit, herding and evasion: A three-agent model of caribou predation," in *ACC 2013*, pp. 2978–2983, June, 2013.

[9] A. Perelman, T. Shima, and I. Rusnak, "Cooperative differential games strategies for active aircraft protection from a homing missile," *J. Guid. Contr. Dynam.*, vol. 34, no. 3, pp. 761–773, 2011.

[10] Z. Fuchs, P. Khargonekar, and J. Evers, "Cooperative defense within a single-pursuer, two-evader pursuit evasion differential game," in *CDC 2010*, pp. 3091–3097, Dec., 2010.

[11] E. Garcia, D. W. Casbeer, and M. Pachter, "Cooperative strategies for optimal aircraft defense from an attacking missile," *J. Guid. Contr. Dynam.*, pp. 1–11, 2015.

[12] R. Vidal, O. Shakernia, H. Kim, D. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 662–669, 2002.

[13] A. Antoniades, H. Kim, and S. Sastry, "Pursuit-evasion strategies for teams of multiple agents with incomplete information," in *CDC 2003*, pp. 756–761, Dec., 2003.

[14] L. Schenato, S. Oh, S. Sastry, and P. Bose, "Swarm coordination for pursuit evasion games using sensor networks," in *ICRA 2005*, pp. 2493–2498, April, 2005.

[15] J. S. Jang and C. J. Tomlin, "Control strategies in multi-player pursuit and evasion game," *AIAA Journal*, vol. 6239, pp. 15–18, 2005.

[16] H. Huang, W. Zhang, J. Ding, D. Stipanovic, and C. Tomlin, "Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers," in *CDC-ECC 2011*, pp. 4835–4840, Dec., 2011.

[17] S. Pan, H. Huang, J. Ding, W. Zhang, D. Stipanovic, and C. Tomlin, "Pursuit, evasion and defense in the plane," in *ACC 2012*, pp. 4167–4173, June, 2012.

[18] E. Bakolas and P. Tsiotras, "Relay pursuit of a maneuvering target using dynamic Voronoi diagrams," *Automatica*, vol. 48, no. 9, pp. 2213–2220, 2012.

[19] W. Sun and P. Tsiotras, "A sequential pursuer-target assignment problem under external disturbances," in *CDC 2013*, pp. 3994–3999, Dec., 2013.

[20] E. Bakolas, "Evasion from a group of pursuers with double integrator kinematics," in *CDC 2013*, pp. 1472–1477, Dec., 2013.

[21] T. Nishida, S. Ono, and K. Sugihara, "Direct diffusion method for the construction of generalized Voronoi diagrams," in *ISVD'07*, pp. 145–151, July, 2007.

[22] O. Marques, *Practical Image and Video Processing using MATLAB*. New Jersey: John Wiley & Sons, 2011.