



# Pursuer Assignment and Control Strategies in Multi-Agent Pursuit-Evasion Under Uncertainties

Leiming Zhang<sup>1</sup>, Amanda Prorok<sup>2</sup> and Subhrajit Bhattacharya<sup>1\*</sup>

<sup>1</sup>Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA, United States, <sup>2</sup>Department of Computer Science and Technology, Cambridge University, Cambridge, United Kingdom

## OPEN ACCESS

### Edited by:

Savvas Loizou,  
Cyprus University of Technology,  
Cyprus

### Reviewed by:

Nicholas Stiffler,  
University of South Carolina,  
United States  
Ning Wang,  
Harbin Engineering University, China

### \*Correspondence:

Subhrajit Bhattacharya  
sub216@lehigh.edu

### Specialty section:

This article was submitted to  
Multi-Robot Systems,  
a section of the journal  
Frontiers in Robotics and AI

**Received:** 06 April 2021

**Accepted:** 31 July 2021

**Published:** 17 August 2021

### Citation:

Zhang L, Prorok A and Bhattacharya S  
(2021) Pursuer Assignment and  
Control Strategies in Multi-Agent  
Pursuit-Evasion Under Uncertainties.  
*Front. Robot. AI* 8:691637.  
doi: 10.3389/frobt.2021.691637

We consider a pursuit-evasion problem with a heterogeneous team of multiple pursuers and multiple evaders. Although both the pursuers and the evaders are aware of each others' control and assignment strategies, they do not have exact information about the other type of agents' location or action. Using only noisy on-board sensors the pursuers (or evaders) make probabilistic estimation of positions of the evaders (or pursuers). Each type of agent use Markov localization to update the probability distribution of the other type. A search-based control strategy is developed for the pursuers that intrinsically takes the probability distribution of the evaders into account. Pursuers are assigned using an assignment algorithm that takes redundancy (i.e., an excess in the number of pursuers than the number of evaders) into account, such that the total or maximum estimated time to capture the evaders is minimized. In this respect we assume the pursuers to have clear advantage over the evaders. However, the objective of this work is to use assignment strategies that minimize the capture time. This assignment strategy is based on a modified Hungarian algorithm as well as a novel algorithm for determining assignment of redundant pursuers. The evaders, in order to effectively avoid the pursuers, predict the assignment based on their probabilistic knowledge of the pursuers and use a control strategy to actively move away from those pursues. Our experimental evaluation shows that the redundant assignment algorithm performs better than an alternative nearest-neighbor based assignment algorithm<sup>1</sup>.

**Keywords:** multi-robot systems, pursuit-evasion, probabilistic robotics, redundant robots, assignment

## 1 INTRODUCTION

### 1.1 Motivation

Pursuit-evasion is an important problem in robotics with a wide range of applications including environmental monitoring and surveillance. Very often evaders are adversarial agents whose exact locations or actions are not known and can at best be modeled stochastically. Even when the pursuers are more capable and more numerous than the evaders, capture time may be highly unpredictable in such probabilistic settings. Optimization of time-to-capture in presence of uncertainties is a challenging task, and an understanding of how best to make use of the excess resources/

<sup>1</sup>Some parts of this paper appeared as an extended abstract in the proceeding of the 2019 IEEE International Symposium on Multi-robot and Multi-agent Systems (MRS) (Zhang et al., 2019).

capabilities is key to achieving that. This paper address the problem of assignment of pursuers to evaders and control of pursuers under such stochastic settings in order to minimize the expected time to capture.

## 1.2 Problem Overview

We consider a multi-agent pursuit-evasion problem where, in a known environment, we have several surveillance robots (the pursuers) for monitoring a workspace for potential intruders (the evaders). Each evader emits a weak and noisy signal (for example, wifi signal used by the evaders for communication or infrared heat signature), that the pursuers can detect using noisy sensors to estimate their position and try to localize them. We assume that the signals emitted by each evader are distinct and is different from any type of signal that the pursuers might be emitting. Thus the pursuers can not only distinguish between the signals from the evaders and other pursuers, but also distinguish between the signals emitted by the different evaders. Likewise, each pursuer emits a distinct weak and noisy signal that the evaders can detect to localize the pursuers. Each agent is aware of its own location in the environment and the agents of the same type (pursuers or evaders) can communicate among themselves. The environment (obstacle map) is assumed to be known to either type of agents.

Each evader uses a control strategy that actively avoids the pursuers. The pursuers need to use an assignment strategy and a control strategy that allow them to follow the path with least expected capture time. The evaders and pursuers are aware of each others' strategies (this, for example, represents real-world scenario where every agent uses an open-source control algorithm), however, the exact locations and actions taken by one type of agent (evader/pursuer) at an instant of time is not known to the other type (pursuer/evader). Using the noisy signals and probabilistic sensor models, each type of agent maintains and updates (based on sensor measurements as well as the known control/motion strategy) a probability distribution that random variable for evader position type (pursuer/evader) see **Figure 1**. In this paper we use a first-order dynamics (velocity control) model for point agents (pursuers or evaders) as is typically done in many multi-agent problems such as coverage control (Cortes et al., 2004; Bhattacharya et al., 2014) and artificial potential function based navigation Rimon and Koditschek (1992).

## 1.3 Contributions

The main contributions of this paper are novel methods for pursuer-to-evader assignment in presence of uncertainties for total capture time minimization as well as for maximum capture time minimization. We also present a novel control algorithm for pursuers based on Theta\* search (Nash et al., 2007) that takes the evaders' probability distribution into account, and present a control strategy for evaders that try to actively avoid the pursuers trying to capture it. We assume that both groups of agents (pursuers and evaders) are aware of the control strategies employed by the other group, and can use that knowledge to predict and update the probability distributions that are used for internal representations of the competing group.

## 1.4 Overview of the Paper

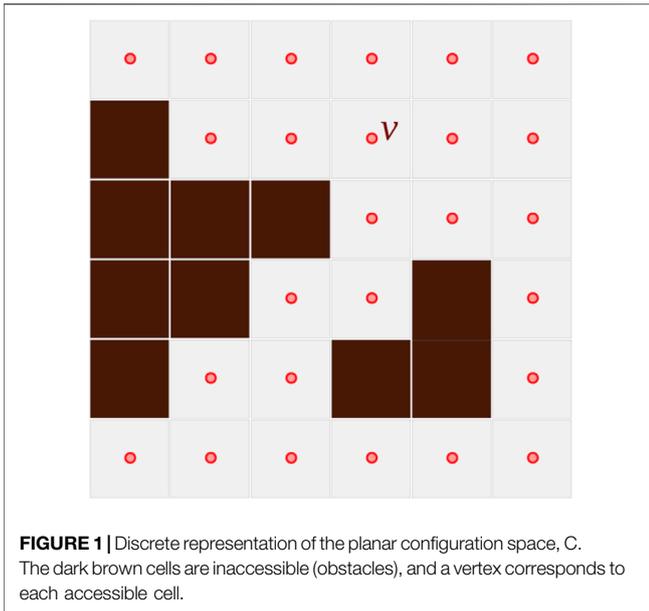
**Section 3** provides the technical tools and background for formally describing the problem. In **Section 4**, we introduce the control strategies for the evaders and pursuers. In presence of uncertainties this control strategy becomes a stochastic one. We also describe how each type of agent predict and update the probability distributions representing the other type using this known control strategy. In **Section 5**, we present algorithms for assigning pursuers to the probabilistic evaders so as to minimize the expected time to capture. In **Section 6** simulation and comparison results are presented.

## 2 RELATED WORK

The pursuit-evasion problem in a probabilistic setting requires localization of the evaders as well as development of a controller for the pursuer to enable it to capture the evader. Markov localization is an effective approach for tracking probabilistic agents in unstructured environments since it is capable of representing probability distributions more general than normal distributions [unlike Kalman filters (Barshan and Durrant-Whyte, 1995)]. Compared to Monte Carlo or particle filters (Fox et al., 1999a; Fox et al., 1999b), Markov localization is often computationally less intensive, more accurate and has stronger formal underpinnings.

Markov localization has been widely used for estimating an agent's position in known environments (Burgard et al., 1996) and in dynamic environments (Fox et al., 1999b) using on-board sensors, as well as for localization of evaders using noisy external sensors (Fox et al., 1998; Fox et al., 1999b; Zhang, 2007a). More recently, in conjunction with sensor fusion techniques, Markov localization has been used for target tracking using multiple sensors (Zhang, 2007b; Nagaty et al., 2015).

Detection and pursuit of an uncertain or unpredictable evader has also been studied extensively. (Chung et al., 2011) provides a taxonomy of search and pursuit problems in mobile robotics. Different methods are compared in both graphs and polygonal environments. Under that taxonomy, our work falls under the domain of probabilistic search problems with multiple heterogeneous searchers/pursuers and multiple targets on a finite graph representation of the environment. Importantly, this survey however notes that the minimization of distance and time to capture the evaders is less studied. (Khan et al., 2016) is another comprehensive review focused on cooperative multi-robot targets observation. (Hollinger et al., 2007) describes strategies for pursuit-evasion in an indoor environment which is discretized into different cells, with each cell representing a room. However, in our approach, the environment is discretized into finer grids that generalize to a wider variety of environments. In (Hespanha et al., 1999) a probabilistic framework for a pursuit-evasion game with one evader and multiple pursuers is described. A game-theoretic approach is used in (Hespanha et al., 2000) to describe a pursuit-evasion game in which evaders try to actively avoid the pursuers. (Makkapati and Tsiotras, 2019) describes an optimal strategy for evaders in multi-agent pursuit-evasion without uncertainties. Along similar lines, in (Oyler et al.,



2016) the authors describe a pursuit-evasion game in presence of obstacles in the environment. (Shkurti et al., 2018) describes a problem involving a robot that tries to follow a moving target using visual data. Patrolling is another approach to pursuit-evasion problems in which persistent surveillance is desired. Multi-robot patrolling with uncertainty have been studied extensively in (Agmon et al., 2009), (Agmon et al., 2012) and (Talmor and Agmon, 2017). More recently in (Shah and Schwager, 2019), Voronoi partitioning has been used to guide pursuers to maximally reduce the area of workspace reachable by a single evader. Voronoi partitioning along with area minimization has also been used for pursuer-to-evader assignments in problems involving multiple deterministic and localized evaders and pursuers (Pierson et al., 2017).

### 3 PROBLEM FORMULATION

#### 3.1 Representing the Pursuers, Evaders, and Environment

Since the evaders are represented by probability distributions by the pursuers, the time-to-capture an evader by a particular pursuer is a stochastic variable. We thus consider the problems of pursuer-to-evader assignment and computation of control velocities for the pursuers with a view of minimizing the total expected capture time (the sum of the times taken to capture each of the evaders) or the maximum expected capture time (the maximum out of the times taken to capture each of the evaders). We assume that the number of pursuers is greater than the number of evaders and that the pursuers constitute a heterogeneous team, with each having different maximum speeds and different capture capabilities. The speed of the pursuers are assumed to be higher than the evaders to enable capture in any environment (even obstacle-free or unbounded environment). The objective of this paper is to design strategies

for the pursuers to assign themselves to the evaders, and in particular, algorithms for assignment of the excess (redundant) pursuers, so as to minimize the total/maximum expected capture time.

While the evaders know the pursuers' assignment strategy, they don't know the pursuers' positions, the probability distributions that the pursuers use to represent the evaders, or the exact assignment that the evaders determine. Instead, the evaders rely on the probability distributions that they use to represent the pursuers to figure out the assignments that the pursuers are likely using. We use a Markov localization (Thrun et al., 2005) technique to update the probability distribution of each agent.

Throughout this paper we use the following notations to represent the agents and the environment:

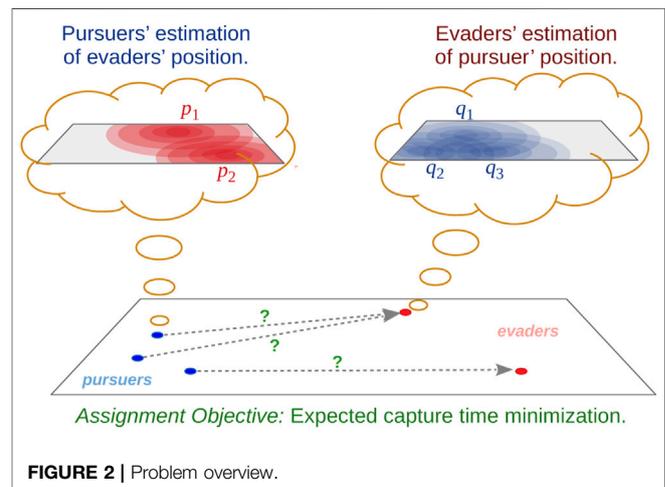
**Configuration Space Representation:** We consider a subset of the Euclidean plane,  $C \subset \mathbb{R}^2$ , as the configuration space for the pursuers as well as the evaders, which we discretize into a set of cells or vertices,  $V$ , where the agents can reside (Figure 1). A vertex in  $V$  will be represented with a lower-case letter  $v \in V$ , while its physical position (Euclidean coordinate vector) in  $C$  will be represented as  $\mathbf{X}(v)$ . For simplicity, we also use a discrete time representation.

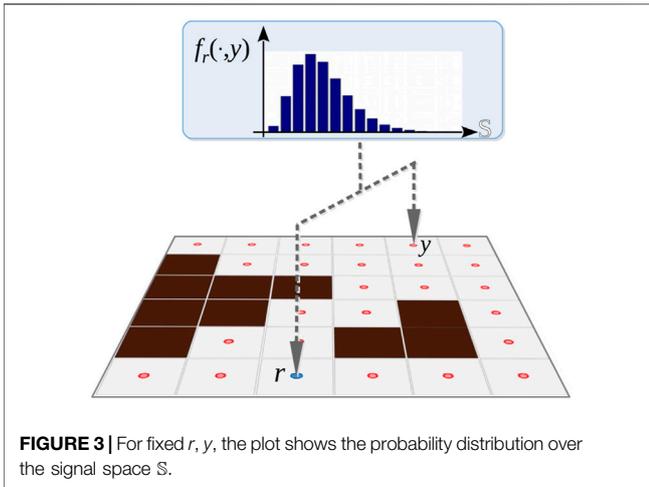
**Agents:** The  $i^{\text{th}}$  pursuer's location is represented by  $r_i \in V$ , and the  $j^{\text{th}}$  evader by  $y_j \in V$  (we will use the same notations to refer to the respective agents themselves). The set of the indices of all the pursuers is denoted by  $C_r$ , and the set of the indices of all the evaders by  $C_y$ .

**Heterogeneity:** Pursuer  $r_i$  is assumed to have a maximum speed of  $v_i$ , and the objective being time minimization, it always maintains that highest possible speed. It also has a capture radius (i.e., the radius of the disk within which it can capture an evader) of  $\rho_i$ .

#### 3.2 Probabilistic Representations

The pursuers represent the  $j^{\text{th}}$  evader by a probability distribution over  $V$  denoted by  $p_j^i: V \rightarrow \mathbb{R}_+$ . Likewise the evaders represent the  $i^{\text{th}}$  pursuer by a probability distribution over  $V$  denoted by  $q_i^j: V \rightarrow \mathbb{R}_+$ . The pursuers maintain the evader distributions,  $\{p_j^i\}_{j \in C_y}$ , which are unknown to the evaders. While the evaders





maintain the pursuer distributions,  $\{q_i^t\}_{i \in C_r}$ , which are unknown to the pursuers (see **Figure 2**). The superscript  $t$  emphasizes that the distributions are time-varying since they are updated by each type of agent (pursuer/evader) based on known control strategy of the other type of agent (evader/pursuer) and models for sensors on-board the agents.

### 3.2.1 Motion Model

At every time-step the known control strategy (hence, known transition probabilities) allows one type of agent to predict the probability distribution of the other type of agent in the next time-step:

$$\begin{aligned} \text{Pursuer's estimation of evader's position (prediction step): } \tilde{p}_j^t(y) &= \sum_{y' \in V} K_j(y, y') p_j^{t-1}(y') \\ \text{Evader's estimation of pursuer's position (prediction step): } \tilde{q}_i^t(r) &= \sum_{r' \in V} L_i(r, r') q_i^{t-1}(r') \end{aligned} \quad (1)$$

where using the first equation the pursuers predict the  $j^{\text{th}}$  evader's probability distribution at the next time-step using the transition probabilities  $K_j$  computed using the known control strategy of the evader. While the second equation is used by the evaders to predict the  $i^{\text{th}}$  pursuer's probability distribution using transition probabilities  $L_i$  computed from the known control strategy of the pursuers. These control strategies and the resulting transition probabilities will be discussed in more details in **Sections 4.1** and **4.2**.

### 3.2.2 Sensor Model

We assume that the probability that a pursuer at  $r \in V$  measures signal  $s$  (in some discrete signal space  $\mathbb{S}$ ) using its on-board sensors if the evader is at  $y \in V$  is given by the probability distribution  $f_r: \mathbb{S} \times V \rightarrow \mathbb{R}_+$ ,  $f_r(s, y) = \mathbb{P}(\mathcal{S} = s \mid \mathcal{Y} = y)$  where,  $\mathcal{S}$  is the random variable for signal measurement, and  $\mathcal{Y}$  is the random variable for evader position (see **Figure 3**). Likewise,  $h_y(s, r) = \mathbb{P}(\mathcal{S} = s \mid \mathcal{R} = r)$  is the sensor model used by the evaders giving the probability that an evader at  $y$  measures signal  $s$  when a pursuer is at  $r$ .

Using Bayes' rule, the updated probability distribution of the  $j^{\text{th}}$  evader as computed by a pursuer at  $r$ , based on sensor measurement,  $s^t$ , and the prior probability estimate,  $\tilde{p}_j^t$ , is

$$\begin{aligned} p_j^t(y) &= \mathbb{P}(\mathcal{Y}_j = y \mid \mathcal{S}_j = s^t) = \mathbb{P}(\mathcal{S}_j = s^t \mid \mathcal{Y}_j = y) \\ \mathbb{P}(\mathcal{Y}_j = y) &= \frac{f_r(s^t, y) \tilde{p}_j^t(y)}{\sum_{y' \in V} f_r(s^t, y') \tilde{p}_j^t(y')} \end{aligned}$$

If multiple signals,  $s_1^t, s_2^t, \dots$ , are received by pursuers  $r_1, r_2, \dots$  at a time step, they are incorporated in sequence:

Pursuer's estimation of evader's position (*updatestep*):

$$p_j^t(y) = \prod_l \frac{f_{r_l}(s_l^t, y)}{\sum_{y' \in V} f_{r_l}(s_l^t, y') \tilde{p}_j^t(y')} \tilde{p}_j^t(y) \quad (2)$$

Likewise, the evaders  $y_1, y_2, \dots$  measuring signals  $s_1^t, s_2^t, \dots$  update the probability distributions that they use to represent the  $i^{\text{th}}$  pursuer according to

Evader's estimation of pursuer's position (*updatestep*):

$$q_i^t(r) = \prod_l \frac{h_{y_l}(s_l^t, r)}{\sum_{r' \in V} h_{y_l}(s_l^t, r') \tilde{q}_i^t(r')} \tilde{q}_i^t(r) \quad (3)$$

The specific functional form for  $f$  and  $h$  depend not only on the distance between the pursuers and the evaders in the environment, but also on the obstacles that results on degradation of the signals emitted by the agents. The details of the specific sensor models appear in the "Results" section (**Section 6**).

## 3.3 Assignment Fundamentals

The goal for our assignment strategy is to try to find the assignment that minimizes either the total expected capture time (the sum of the times taken to capture each of the evaders in  $C_y$ ) or the maximum expected capture time (the maximum out of the times taken to capture each of the evaders in  $C_y$ ). We assume that there are more pursuers in the environment than the number of evaders. The following subsection provides some fundamental definitions and tools that are used to describe and solve the optimal assignment problem in **Section 5**.

### 3.3.1 Formal Description of Assignment

In order to formally describe the assignment problem, we use the following notations:

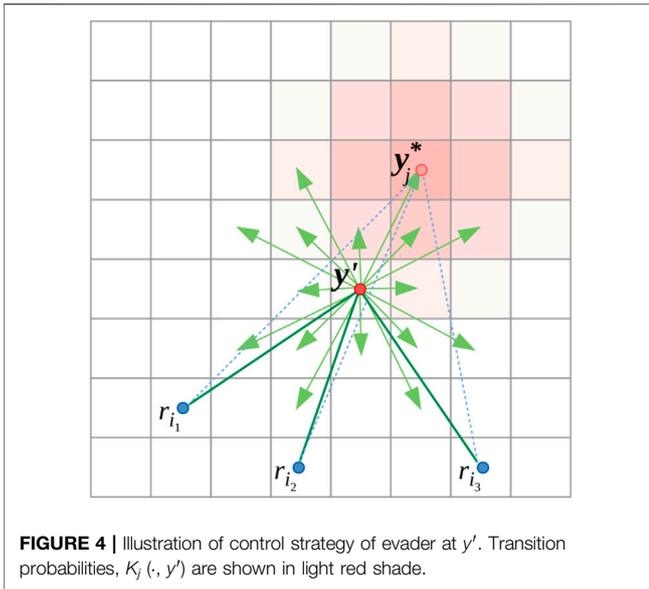
**Assignment:** The set of pursuers assigned to the  $j^{\text{th}}$  evader will be represented by the set  $I_j$ . The individual assignment of  $i^{\text{th}}$  pursuer to  $j^{\text{th}}$  evader will be denoted by the pair  $(i, j)$ .  $\mathcal{F} = \{(i, j) \mid i \in C_r, j \in C_y\}$  denotes the set of all possible such pursuer-to-evader pairings.

A (valid) assignment,  $\mathcal{A} \subseteq \mathcal{F}$ , is such that for every  $(i, j), (i', j') \in \mathcal{A}$ , we should have  $i = i' \Rightarrow j = j'$  (i.e., a pursuer cannot be assigned to two different evaders). This also implies  $|\{j \mid (i, j) \in \mathcal{A}\}| \leq 1, \forall i \in C_r$  (note that an assignment allows for unassigned pursuers).

The set of all possible valid assignments is denoted by  $\mathfrak{A} = \{\mathcal{A} \subseteq \mathcal{F} \mid \forall (i, j), (i', j') \in \mathcal{A}, i = i' \Rightarrow j = j'\}$ .

### 3.3.2 Probabilistic Assignment Costs

In this section we consider the time that pursuer  $i$  takes to capture evader  $j$ . We describe the computation from the perspective of the pursuers. Since the evader  $j$  is represented



by the probability distribution,  $p_j$ , over  $V$ , we denote  $T_{ij}$  as the random variable representing the uncertain travel time from pursuer  $i$  to evader  $j$ . The probability that  $T_{ij}$  falls within a certain interval is the sum of all the probabilities on the vertices of  $V$  such that the travel time from  $r_i$  to the vertex is within that interval. That is,

$$\mathbb{P}(T_{ij} \in [\tau, \tau + \Delta\tau]) = \sum_{\{y \in V \mid \frac{1}{v_i} d_g(r_i, y) \in [\tau, \tau + \Delta\tau]\}} p_j(y)$$

We first note that  $T_{ij}$  and  $T_{i'j'}$  are independent variables whenever  $j$  and  $j'$  are different (i.e., the time taken to reach evader  $j$  does not depend on time taken to reach evader  $j'$ ). However,  $T_{ij}$  and  $T_{i'j}$  are dependent random variables since, for a given travel time (and hence travel distance) from pursuer  $i$  to evader  $j$ , and knowing the distance between pursuers  $i$  and  $i'$ , the possible values of distances between pursuer  $i'$  and evader  $j$  are constrained by the triangle inequality. That is, for any given  $j$ , the random variables in the set  $\{T_{ij} \mid i \in \mathcal{I}\}$ , where  $\mathcal{I}$  is a set of pursuer indices, are dependent. This can be seen more clearly by considering a potential evader position  $y \in V$  which has an associated probability of  $p_j(y)$ . Given that position,  $\frac{1}{v_i} d_g(y, r_i)$  is the time taken by the pursuer  $i \in \mathcal{I}$  to reach the evader. In particular, the following holds:

$$\begin{aligned} & \mathbb{P}\left(\bigwedge_{i \in \mathcal{I}} T_{ij} \in [\tau_i, \tau_i + \Delta\tau_i]\right) \\ &= \sum_y \mathbb{P}\left(\bigwedge_{i \in \mathcal{I}} \frac{1}{v_i} d_g(r_i, y) \in [\tau_i, \tau_i + \Delta\tau_i]\right) \\ &= \sum_{\{y \in V \mid \frac{d_g(r_i, y)}{v_i} \in [\tau_i, \tau_i + \Delta\tau_i], \forall i \in \mathcal{I}\}} p_j(y) \end{aligned} \tag{4}$$

Thus, in order to compute the joint probability distributions of  $\{T_{ij} \mid i \in \mathcal{I}\}$ , we can sample a  $y$  from the probability distribution  $p_j$  and compute the travel times  $\tau_i = \frac{1}{v_i} d_g(r_i, y)$ ,  $i \in \mathcal{I}$ , and hence populate the distribution.

### 3.4 Problem Objectives

In the next sections we will describe the control strategy used by a pursuer that allows it to effectively capture the evader assigned to it, as well as the control strategy of an evader that allows it to move away from the pursuers assigned to it.

In **Section 5**, for designing the assignment strategy for the pursuers we will consider two metrics to minimize: 1) the total expected capture time, which is the sum of the times taken to capture each of the evaders, and, 2) the maximum expected capture time, which is the times taken to capture the last evader. While the actual assignment is computed by the pursuers and unavailable to the evaders, the evaders will estimate the likely assignment in order to determine their control strategy.

As mentioned earlier, we assume that both types of agents know all the strategies used by the other type of agents. That is, the pursuers know the evaders' control strategy and the evaders know the pursuers' control and assignment strategies. However the pursuers do not know the evaders' exact position and vice versa. Instead they reason about that by maintaining probability distributions representing the positions of the other type of agents and update those distributions using the known control strategies of the other type of agents and weak signals measured by onboard sensors.

## 4 CONTROL STRATEGIES

Assuming a known pursuer-to-evader assignment, in this section we describe the control strategies used by the evaders to avoid being captured and the control strategy used by the pursuers to capture the evaders.

### 4.1 Evader Control Strategy

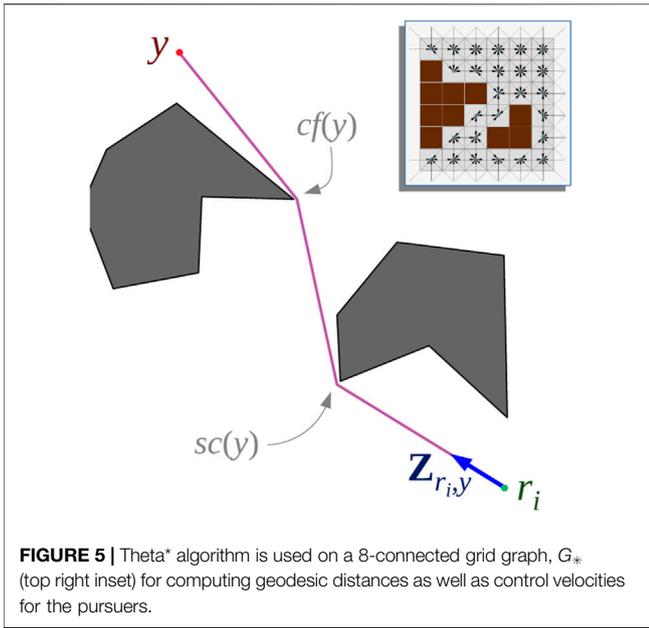
In presence of pursuers, an evader  $y_j$  actively tries to move away from the pursuers targeting it. With the evader at  $y \in V$  and deterministic pursuers,  $\{r_i\}_{i \in I_j}$ , trying to capture it, we define a mean capture time as the harmonic mean of the capture time for each of the pursuers:

$$\tau(y, \{r_i\}_{i \in I_j}) = \frac{1}{\sum_{i \in I_j} \frac{1}{\tilde{d}_g(r_i, y)/v_i}} \tag{5}$$

where  $\tilde{d}_g(r_i, y) = \max(0, d_g(r_i, y) - \rho_i)$  is the effective geodesic distance between  $r_i, y \in V$  (with  $d_g(r_i, y)$  being the geodesic distance or shortest path length between  $r_i$  and  $y$ ), which accounts for the fact that pursuer  $r_i$  has a capture radius of  $\rho_i$ . For a given set of pursuer positions,  $\tau$  thus a function that has higher value on the vertices in  $V$  that are farther away from the pursuers in  $I_j$ . The reason behind taking harmonic mean is that the harmonic mean gets lower contribution from distant pursuers and higher contribution from the nearby pursuers.

In order to determine the best action that the evader at  $y' \in V$  can take, it computes the marginal increase in  $\tau$  if it moves to  $y \in V$  (**Figure 4**):

$$\Delta\tau(y, y', \{r_i\}_{i \in I_j}) = \max(0, \tau(y, \{r_i\}_{i \in I_j}) - \tau(y', \{r_i\}_{i \in I_j}) + \epsilon) \tag{6}$$



where  $\epsilon$  is a small number that gives a small positive marginal increase for some neighboring vertices in scenarios when the evader gets cornered against an obstacle.

#### 4.1.1 Evader’s Control Strategy

In a deterministic setup the evader at  $y'$  will move to

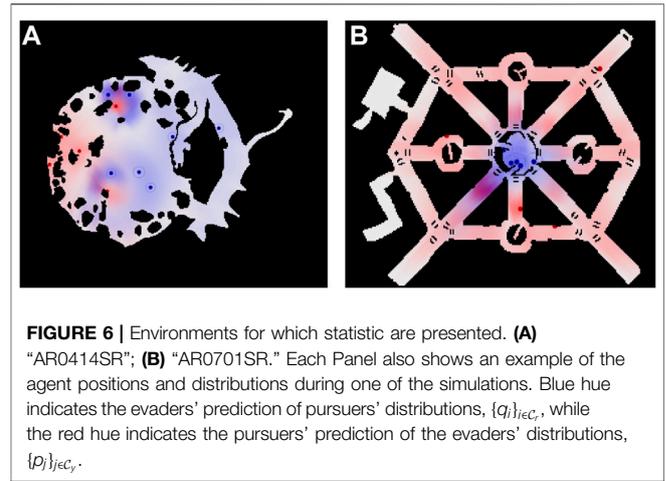
$$y_j^*(y', \{r_i\}_{i \in I_j}) := \arg \max_{y \in A_{y'}} \Delta \tau(y, y', \{r_i\}_{i \in I_j}) \quad (7)$$

where  $A_{y'}$  refers to the states/vertices in the vicinity of  $y'$  that the evader can transition to in the next time-step. But, in the probabilistic setup where the evaders represent the  $i^{\text{th}}$  pursuer by the distribution  $q_i$ , with every  $y \in A_{y'}$  an evader associates a probability that it is indeed the best transition to make. In practice, these probabilities are computed by sampling  $\{r_i\}_{i \in I_j}$  from the distributions  $\{q_i\}_{i \in I_j}$ , and counting the proportion of samples for which a  $y \in A_{y'}$  is the neighbor that maximizes the marginal increase in capture time. The evader then uses this probability distribution over its neighboring states to make a stochastic transition.

#### 4.1.2 Pursuer’s Prediction of Evader’s Distribution Based on Known Evader Control Strategy

The pursuers know the evader’s strategy of maximizing the marginal increase in capture time. However, they do not know the evaders’ exact position, nor do they know the distributions,  $q_i$ , that the evaders maintain of the pursuers. The uncertainty in the action of the evader due to that is modeled by a normal distribution centered at  $y_j^*(y', \{r_i\}_{i \in I_j})$ . If the evader is at  $y'$ , the transition probability  $K_j(y, y')$  is assumed to be

$$K_j(y, y') = \begin{cases} \kappa_j \exp\left(-\frac{d_f(y, y_j^*(y', \{r_i\}_{i \in I_j}))^2}{2\sigma_j^2}\right), & \text{if } y \in A_{y'} \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$



where, for simplicity,  $d_f$  is assumed to be the Euclidean distance between the neighboring vertices in the graph, and  $\kappa_j$  is a normalization factor so that  $\sum_{y \in V} K(y, y') = 1$ .

### 4.2 Pursuer Control Strategy

A pursuer,  $r_i \in I_j$ , pursuing the evader at  $y_j$  needs to compute a velocity for doing so.

In a deterministic setup, if the evader is at  $y_j \in V$ , the pursuer’s control strategy is to follow the shortest (geodesic) path in the environment connecting  $r_i$  to  $y_j$ . This controller, in practice, can be implemented as a gradient-descent of the square of the path metric (geodesic distance) and is given by  $\mathbf{v}_i = k \frac{\partial d_g(r_i, y_j)^2}{\partial \mathbf{x}(r_i)} = 2k d_g(r_i, y_j) \hat{z}_{r_i, y_j}$ , where  $k$  is a proportionality constant,  $d_g(r_i, y_j)$  is the shortest path (geodesic) distance between  $r_i$  and  $y_j$ , and  $z_{r_i, y_j}$  is the unit vector to the shortest path at  $r_i$  (see Figure 5). Such a controller does not suffer from local minimas due to presence of non-convex obstacles since the geodesic paths go around obstacles. A formal proof of that and the fact that  $\frac{\partial d_g(r, y)}{\partial \mathbf{x}(r)} = \hat{z}_{r, y}$ , appeared in (Bhattacharya et al., 2014). This gives a simple velocity controller for the pursuer.

#### 4.2.1 Pursuer’s Control Strategy

Since the pursuers describe the  $j^{\text{th}}$  evader’s position by the probability distribution  $p_j^i$  over  $V$ , we compute an expectation on the velocity vectors of the  $i^{\text{th}}$  pursuer (with  $i \in I_j$ ) as follows:

$$\hat{\mathbf{v}}_i = \sum_{y \in V} 2k d_g(r_i, y) \hat{z}_{r_i, y} p_j^i(y) \quad (9)$$

Since the pursuer has a maximum speed of  $v_i$ , and the exact location of the evader is unknown, we always choose the maximum as speed for the pursuer:  $\mathbf{v}_i = v_i \frac{\hat{\mathbf{v}}_i}{\|\hat{\mathbf{v}}_i\|}$ .

For computing  $d_g(r_i, y)$  we use the Theta\* search algorithm (Nash et al., 2007) on a uniform 8-connected square grid graph,  $G_*$ , representation of the environment (Figure 5 inset). While very similar to Dijkstra’s and A\*, Theta\* computes paths that are not necessary restricted to the graph and are closer to the true shortest path in the environment. While more advanced

variations of the algorithm exists [such as Lazy Theta\* (Nash et al., 2010) and Incremental Phi\* (Nash et al., 2009)], we choose to use the most basic variety for simplicity. Computation of the sum in **Equation 9** can also be performed during the Theta\* search. **Algorithm 1** describes the computation of  $d_g(r_i, y)$  (the shortest path (geodesic) distance between  $r_i$  and a point  $y$  in the environment) and the control velocity  $\mathbf{v}_i$ .

The algorithm is reminiscent of Dijkstra's search, maintaining an open list,  $Q$ , and expanding the least  $g$ -score vertex at every iteration, except that the came-from vertex ( $cf$ ) of a vertex can be a distant predecessor determined by line of sight (Lines 10–14) and the summation in (9) is computed on-the-fly during the execution of the search (Line 28).

We start the algorithm by initiating the open list with the single start vertex,  $r_i$ , set its  $g$ -score to zero, and its came-from vertex,  $cf$ , to reference to itself (line 4). Every time a vertex,  $y$  (one with the minimum  $g$ -score in the open list, maintained using a heap data structure), is expanded, Theta\* checks for the possibility of updating a neighbor,  $w$ , from the set of neighbors,  $\mathcal{N}_{G_*}(y)$ , of the vertex that are not in the closed list (line 9). Based on the existence of a direct line of sight from the came-from vertex of  $y$  and the vertex  $w$ , the potential new came-from vertex,  $cf$ , is set to  $cf(y)$  or  $y$ . The new potential  $g$ -score is

computed as the sum of the  $g$ -score of  $cf$  and the Euclidean distance,  $d_E(cf, w) = \|\mathbf{X}(cf) - \mathbf{X}(w)\|$ , between the two vertices. If lower,  $g(w)$  is updated, the came-from vertex of  $w$  is set to  $cf$ , and the vertex on the path second from the start,  $sc(w)$ , is copied from that of  $y$  unless  $w$  is itself second from start. We also compute the control velocity as part of the Theta\* search algorithm. Every time a vertex is expanded, we add the corresponding term in the summation of **Equation 9** to the vector  $\hat{\mathbf{v}}_i$  (line 28), which we scale to have magnitude of the maximum possible speed of the pursuer,  $v_i$ , at the end.

#### 4.2.2 Evader's Prediction of Pursuer's Distribution Based on Known Pursuer Control Strategy

Since the evaders represent the  $i^{\text{th}}$  pursuer using the probability distribution  $q_i$ , they need to predict the pursuer's probability distribution in the next time step knowing the pursuer's control strategy. This task is assigned to the  $j^{\text{th}}$  evader such that  $i \in I_j$  (we define  $j(i)$  to be the index of the evader assigned to pursuer  $i$ ). It executes Theta\* algorithm, similar to **Algorithm 1**, but the start vertex being  $y_j$ . Once executed, the line segment connecting any  $r' \in V$  and  $cf(r')$  gives the direction in which the  $i^{\text{th}}$  pursuer at  $r'$  would tentatively move in the next time-step based on the

---

#### Algorithm 1 | Theta\* Based Pursuer Control

---

**Inputs:** Graph  $G_* = (V, E)$ ; Pursuer location  $r_i \in V$ ; Evader probability distribution  $p_j$ .

**Outputs:** The lengths of shortest paths to all vertices  $g: V \rightarrow \mathbb{R}_+$ ; The control velocity  $\mathbf{v}_i$ .

```

1:  $g(y) \leftarrow \infty$ ,  $cf(y) = \emptyset$ ,  $sc(y) = \emptyset$  for all  $y \in V$ 
   //  $\hat{\hat{\hat{g}}}$   $g$ -score, came-from vertex and second-from-start vertex.
2:  $Q = \emptyset$ ,  $Z = \emptyset$  // open and closed lists
3:  $g(r_i) \leftarrow 0$ ,  $cf(r_i) \leftarrow r_i$ ,  $Q = Q \cup \{r_i\}$ 
4: Set  $\tilde{\mathbf{v}}_i \leftarrow \mathbf{0}$ 
5: while ( $Q \neq \emptyset$ ) do
6:   Set  $y \leftarrow \text{argmin}_{y' \in Q} g(y')$  // pop heap
7:   Set  $Q \leftarrow Q - \{y\}$ ,  $Z \leftarrow Z \cup \{y\}$ 
8:   for each ( $\{w \in \mathcal{N}_{G_*}(y) \mid w \notin Z\}$ ) do
9:     if ( $\text{LineOfSight}(cf(y), w)$ ) then
10:      Set  $\overline{cf} \leftarrow cf(y)$  // potential came-from vertex
11:     else
12:      Set  $\overline{cf} \leftarrow y$  // potential came-from vertex
13:     Set  $\overline{g} \leftarrow g(\overline{cf}) + d_E(\overline{cf}, w)$ 
14:     if ( $\overline{g} < g(w)$ ) then
15:       Set  $g(w) \leftarrow \overline{g}$ 
16:       Set  $cf(w) \leftarrow \overline{cf}$ 
17:       if ( $cf(w) \neq r_i$ ) then
18:         Set  $sc(w) \leftarrow sc(y)$ 
19:       else
20:         Set  $sc(w) \leftarrow w$ 
21:     if ( $y \neq r_i$ ) then // control computation
22:       Set  $\hat{\mathbf{z}}_{r_i, y} \leftarrow \frac{\mathbf{X}(sc(y)) - \mathbf{X}(r_i)}{\|\mathbf{X}(sc(y)) - \mathbf{X}(r_i)\|}$ 
23:       Set  $\tilde{\mathbf{v}}_i \leftarrow \tilde{\mathbf{v}}_i + 2k g(r_i, y) \hat{\mathbf{z}}_{r_i, y} p_j(y)$ 
24: Set  $\mathbf{v}_i \leftarrow v_i \frac{\tilde{\mathbf{v}}_i}{\|\tilde{\mathbf{v}}_i\|}$ 

```

---

aforesaid control strategy of the pursuer. Knowing the speed of a pursuer, the evader can thus compute the next position of the pursuer,  $r_j^*(r', y_j)$ , if it is currently at  $r'$ . However, in order to account for the fact that the pursuer does not precisely know the evader's position (and instead use the distribution  $p_j$  to represent it), analogous to (8), we use the following transition probability for the prediction step of updating  $q_i$

$$L_i(r, r') = \begin{cases} \kappa_i \exp\left(-\frac{d_f\left(r, r_j^*(r', y_{\bar{j}(i)})\right)^2}{2\sigma_i^2}\right), & \text{if } r \in A_{r'} \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

where  $\kappa_i$  is the normalization factor.

## 5 ASSIGNMENT STRATEGIES

We first consider the assignment problem from the perspective of the pursuers—with the evaders represented by probability distributions  $\{p_j\}_{j \in \mathcal{C}_e}$ , what's the best pursuer-to-evader assignment? In a probabilistic setup, where the costs (capture times) are stochastic variables (see Section 3.3.2), and there are excess pursuers, this needs to be solved in two stages (Prorok, 2020): First we need to determine an initial assignment of each evader to one pursuer. Following that we determine the assignment of the remaining (redundant) pursuers so as to minimize the (total or maximum) expected capture time.

### 5.1 Expected Capture Time Minimization for an Initial One-To-One Assignment

In order to determine an *initial* assignment  $\mathcal{A}_0 \subseteq \mathcal{F}$  such that exactly one pursuer is assigned to an evader (thus potentially allowing unassigned pursuers).

Since for every  $(i, j), (i', j') \in \mathcal{A}_0$ ,  $T_{ij}$  and  $T_{i'j'}$  are independent variables, the problem of finding the optimal initial assignment that minimizes the *total expected capture time* becomes<sup>2</sup>

$$\begin{aligned} \mathcal{A}_0 &= \arg \min_{\substack{\mathcal{A}' \subset \mathcal{F} \text{ s.t.} \\ (i,j), (i',j') \in \mathcal{A}' \Rightarrow i \neq i', j \neq j'}} \mathbb{E}\left(\sum_{(i,j) \in \mathcal{A}'} T_{ij}\right) \\ &= \arg \min_{\substack{\mathcal{A}' \subset \mathcal{F} \text{ s.t.} \\ (i,j), (i',j') \in \mathcal{A}' \Rightarrow i \neq i', j \neq j'}} \sum_{(i,j) \in \mathcal{A}'} \mathbb{E}(T_{ij}) \end{aligned} \quad (11)$$

Thus, for computing the initial assignment, it is sufficient to use the numerical costs of  $C_{ij} = \mathbb{E}(T_{ij})$  in the assignment of pursuer  $i$  to evader  $j$ , and thus find an assignment that minimizes the net cost. In practice we use a Hungarian algorithm to compute the assignment. While a Hungarian algorithm is an efficient method

for computing the assignment that minimizes the expected total time of capture, generalizing it to the problem of minimizing the expected maximum capture time is non-trivial, which we address next.

#### 5.1.1 Modified Hungarian Algorithm for Minimization of Maximum Capture Time

For finding the initial assignment that minimizes the *maximum expected capture time*, we develop a modified version of the Hungarian algorithm. To that end we observe that in a Hungarian algorithm, instead of using the expected capture times as the costs, we can use the  $p$ -th powers of the expected capture times,  $C_{ij} = (\mathbb{E}(T_{ij}))^p$ . Making  $p \rightarrow \infty$  results in the appropriate cost that makes the Hungarian algorithm compute an assignment that minimize the maximum expected capture time (the infinity norm). However, for computation we cannot practically raise a number to infinity, and thus need to modify the Hungarian algorithm at a more fundamental level.

In a simple implementation of the Hungarian algorithm (Munkres, 1957), one performs multiple row and column operations on the cost matrix wherein a specific element of the cost matrix,  $C_{i'j'}$ , is added or subtracted from all the elements of a selected subset of rows and columns. Thus, if we want to use the  $p$ th powers of the costs, but choose to maintain only the costs in the matrix (without explicitly raising them to the power of  $p$  during storage), for the row/column operations we can simply raise the elements of the matrix to the power of  $p$  right before the addition/subtraction operations, and then take the  $p$ th roots of the results before updating the matrix entries. That is, addition of  $C_{i'j'}$  to an element  $C_{ij}$  will be replaced by the operation  $C_{ij} \oplus_p C_{i'j'} = \sqrt[p]{C_{ij}^p + C_{i'j'}^p}$ , and subtraction will be replaced by the operation  $C_{ij} \ominus_p C_{i'j'} = \sqrt[p]{C_{ij}^p - C_{i'j'}^p}$ .

Thus, letting  $p \rightarrow \infty$ , we have  $C_{ij} \oplus_\infty C_{i'j'} = \max\{C_{ij}, C_{i'j'}\}$  and  $C_{ij} \ominus_\infty C_{i'j'} = \begin{cases} C_{ij}, & C_{ij} > C_{i'j'} \\ 0, & C_{ij} = C_{i'j'} \end{cases}$ . Thus, we can compute the assignment that achieves the minimization of the maximum expected capture time using this modified algorithm, but without actually needing to explicitly raise the costs to the power of a large  $p \rightarrow \infty$ .

### 5.2 Redundant Pursuer Assignment Approach

After computation of an initial assignment,  $\mathcal{A}_0$ , we determine the assignment of the remaining pursuers using the method proposed in (Prorok, 2020). Formally, we first consider the problem of selecting a set of redundant pursuer-evader matchings,  $\tilde{\mathcal{A}}$ , that minimizes the total expected travel time to evaders, under the constraint that any pursuer is only assigned once:

$$\tilde{\mathcal{A}} = \arg \min_{\substack{\mathcal{A}' \subset \mathcal{F} \text{ s.t.} \\ (i,j), (i',j') \in \mathcal{A}' \cup \mathcal{A}_0 \Rightarrow i \neq i'}} \sum_{(i,j) \in \mathcal{A}'} \mathbb{E}(T_{ij}). \quad (12)$$

Notably, the work in (Prorok, 2020) shows that a cost function such as (12), which considers redundant assignment under

<sup>2</sup>The expectation of the sum of two or more independent random variables is the sum of the expectations of the variables

**Algorithm 2** | Total Time minimization Redundant Pursuer Assignment

**Inputs:** Initial assignment,  $\mathcal{A}_0$ ;  $h$  samples,  $\tilde{T} = \{\{\tau_{ij}^z\}_{i \in \mathcal{C}_r, j \in \mathcal{C}_y}\}_{z=1,2,\dots,h}$ , from the joint probability distribution of the travel times,  $\{T_{ij}\}_{i \in \mathcal{C}_r, j \in \mathcal{C}_y}$ .

**Outputs:** Assignment of the redundant pursuers,  $\bar{\mathcal{A}}$ .

```

1:  $\bar{\mathcal{A}} \leftarrow \emptyset$ 
2: for  $z \in \{1, 2, \dots, h\}$  do
3:   for  $(i, j) \in \mathcal{A}_0$  do
4:      $S_j^z \leftarrow \tau_{ij}^z$ 
5:   for  $k \in \{1, \dots, |\mathcal{C}_r| - |\mathcal{C}_y|\}$  do
6:      $T_{\text{curr}}^* \leftarrow -\infty$ 
7:      $T_{\text{new}}^* \leftarrow \infty$ 
8:     for  $\{(i, j) \in \mathcal{F} - \mathcal{A}_0 - \bar{\mathcal{A}} \mid \mathcal{A}_0 \cup \bar{\mathcal{A}} \cup \{(i, j)\} \in \mathcal{A}\}$  do
9:        $T_{\text{curr}} \leftarrow \frac{1}{h} \sum_{z=1}^h S_j^z$ 
10:       $T_{\text{new}} \leftarrow \frac{1}{h} \sum_{z=1}^h \min(S_j^z, \tau_{ij}^z)$ 
11:      if  $T_{\text{curr}} - T_{\text{new}} > T_{\text{curr}}^* - T_{\text{new}}^*$  then
12:         $T_{\text{curr}}^* = T_{\text{curr}}$ 
13:         $T_{\text{new}}^* = T_{\text{new}}$ 
14:         $(i^*, j^*) \leftarrow (i, j)$ 
15:       $\bar{\mathcal{A}} \leftarrow \bar{\mathcal{A}} \cup \{(i^*, j^*)\}$ 
16:       $S_{j^*}^z \leftarrow \min(S_{j^*}^z, \tau_{i^*j^*}^z) \quad \forall z \in 1, 2, \dots, h$ 
17: return  $\bar{\mathcal{A}}$ 

```

uncertain travel time, is supermodular. It follows that the assignment procedure can be implemented with a greedy algorithm that selects redundant pursuers near-optimally<sup>3</sup>.

**Algorithm 2** summarizes our greedy redundant assignment algorithm. At the beginning of the algorithm, we sample  $h$   $|\mathcal{C}_r| \times |\mathcal{C}_y|$ -dimensional points from the joint probability distribution of  $\{T_{ij}\}_{i \in \mathcal{C}_r, j \in \mathcal{C}_y}$  and store them in the set  $T$ . In practice, the sampling is performed by sampling points,  $y_j \in V$ , from the evaders' probability distributions,  $p_j$ , for all  $j \in \mathcal{C}_y$ . The travel times,  $\tau_{ij} = \frac{1}{v_i} d_g(r_i, y_j)$ ,  $i \in \mathcal{C}_r, j \in \mathcal{C}_y$  then give the sample from the joint probability distributions of  $\{T_{ij}\}_{i \in \mathcal{C}_r, j \in \mathcal{C}_y}$  due to **Eq. 4**. The  $z^{\text{th}}$  sample is thus a set of travel times between every pursuer-evader pair, and will be referred to as  $T^z = \{\tau_{ij}^z\}_{i \in \mathcal{C}_r, j \in \mathcal{C}_y} \in T$ .

In this algorithm, we first consider the initial assignment,  $\mathcal{A}_0$ , and collect all the sampled costs of edges incident on to the  $j^{\text{th}}$  evader into the variable  $S$ . Note that a given  $j \in \mathcal{C}_y$  appears in exactly one element of  $\mathcal{A}_0$ , thus the assignment in Line 4 assigns a value to a  $S_j^z$  exactly once. The set  $\mathcal{A}$  contains the assignment of the remaining/redundant pursuers, that we initiate with the empty set.

In Line 10, we loop over all the possible pursuer-to-evader pairings,  $(i, j)$ , that are not already present in  $\mathcal{A}_0$  or  $\mathcal{A}$ , and which, along with  $\mathcal{A}_0$  or  $\mathcal{A}$ , constitute a valid assignment. We go through all such potential pairings,  $(i, j)$ , and pick the one

with the highest marginal gain,  $T_{\text{curr}}^* - T_{\text{new}}^*$ . The pair with the highest marginal gain, is thus added to  $\mathcal{A}$ . This process is carried out  $|\mathcal{C}_r| - |\mathcal{C}_y|$  times, thus ensuring that all pursuers get assigned.

### 5.3 Equality in Marginal Gain

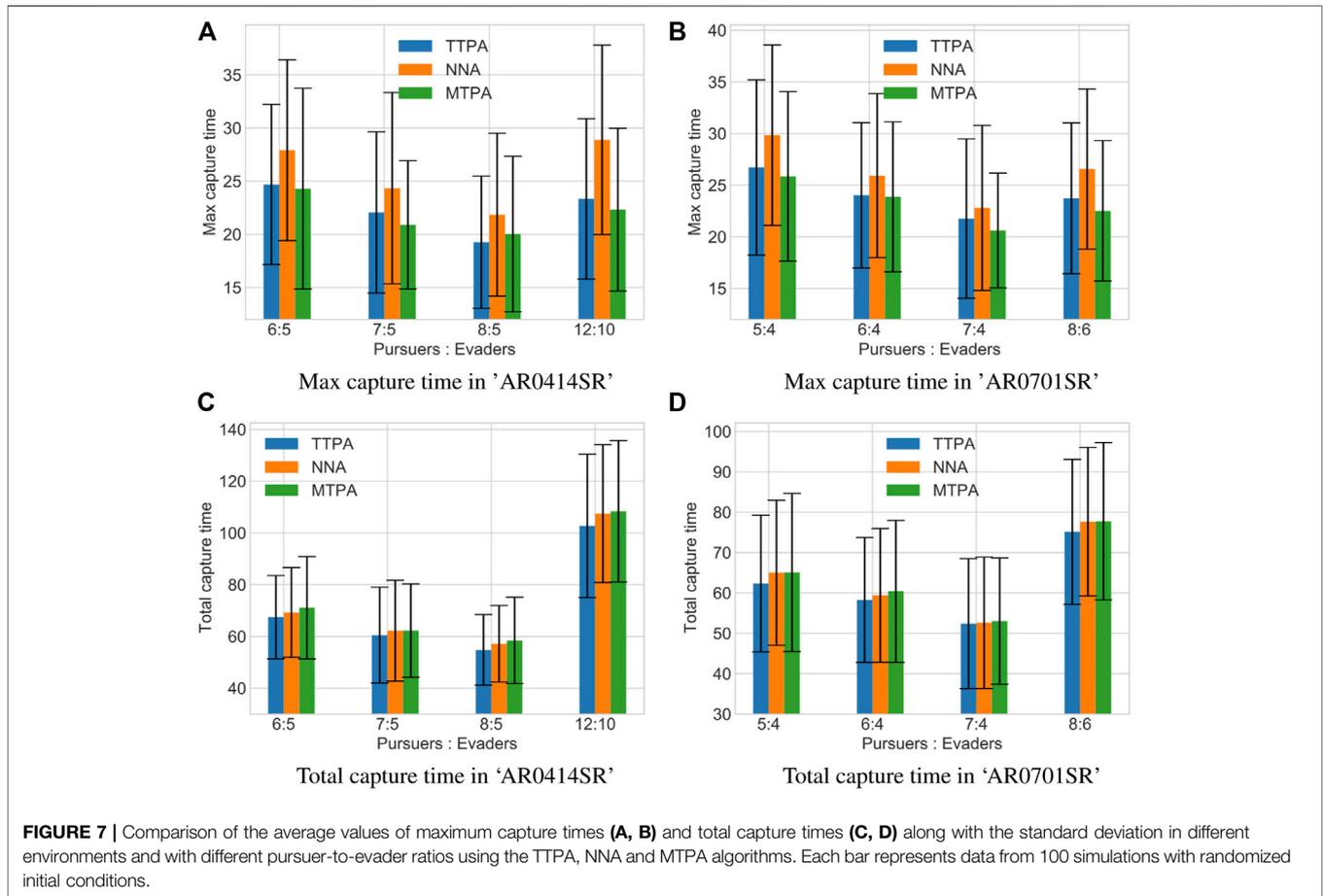
One way that the inequality condition in Line 13 gets violated is when the marginal gains  $T_{\text{curr}} - T_{\text{new}}$  and  $T_{\text{curr}}^* - T_{\text{new}}^*$  are equal. This can in fact happen quite often when one or more redundant pursuers are left to be assigned and all of them are far from all the evaders, rendering marginal gains for any of the assignments close to zero. In that case a pursuer  $i$  gets randomly assigned to an evader  $j$  based on the order in which the pairs  $(i, j) \in \mathcal{F} - \mathcal{A}_0 - \mathcal{A}$  are encountered in the for loop of Line 10.

In order to address this issue properly, we maintain a list of "potential assignments" that corresponds to  $(i, j)$  pairs (along with the corresponding  $T_{\text{new}}$  values maintained as an associative list,  $\mathcal{P} = \{\cup \Delta \cup \{=\} \mathcal{A}^*\}$  that produce the same highest marginal gains (i.e., in line 13 equality holds), and choose the one with the median  $T_{\text{new}}$  value for inserting into the assignment set in Line 19.

#### 5.3.1 Redundant Pursuer Assignment for Minimization of Maximum Capture Time

As for the minimization of the *maximum expected capture time* in the redundant assignment process, we take a similar approach as in **Section 5.1.1**. We first note that choosing  $(\mathbb{E}(T_{ij}))^p$  instead of simply the expected capture time in **(12)** still keeps the cost function supermodular. If we want to minimize the total (sum) expected  $p^{\text{th}}$  power of the capture time, the condition in the *if* statement in line 13 of the above algorithm needs to be simply changed to  $T_{\text{curr}}^p - T_{\text{new}}^p > (T_{\text{curr}}^*)^p - (T_{\text{new}}^*)^p$ . With  $p \rightarrow \infty$ , this

<sup>3</sup>We note that without an initial assignment  $\mathcal{A}_0$ , any solution that is smaller in size than  $|\mathcal{C}_y|$  would lead to an infinite capture time, and hence, the cost function loses its supermodular property. Hence, the assumption that we already have an initial assignment is necessary



**TABLE 1 |** Win rates of TTPA and MTPA algorithms over NNA. For a given set of initial conditions (initial position of pursuers and evaders), if TTPA takes less total time to capture all the evaders than NNA, it is considered a win for TTPA. While if MTPA takes less time to capture the last evader (maximum capture time) than NNA, it is considered as a win for MTPA.

Algorithm name	AR0414SR (%)	AR0701SR (%)
TTPA	69.3	58.2
MTPA	78.0	71.2

condition translates to  $\max(T_{curr}, T_{new}^{\star}) > \max(T_{curr}^{\star}, T_{new})$ . Furthermore, to deal with the equality situations in Line 13, instead of choosing the assignment with the median  $T_{new}$  from  $\mathcal{P} \{ \sqcup \Delta \langle \sqcup = \rangle \mathcal{A}^{\star} \}$ , we choose the one with the maximum  $T_{new}$ .

Thus assigning a redundant pursuer to an evader (out of the assignments that produce the same marginal gain) that has the maximum expected capture time, thus providing some extra help with catching the pursuer.

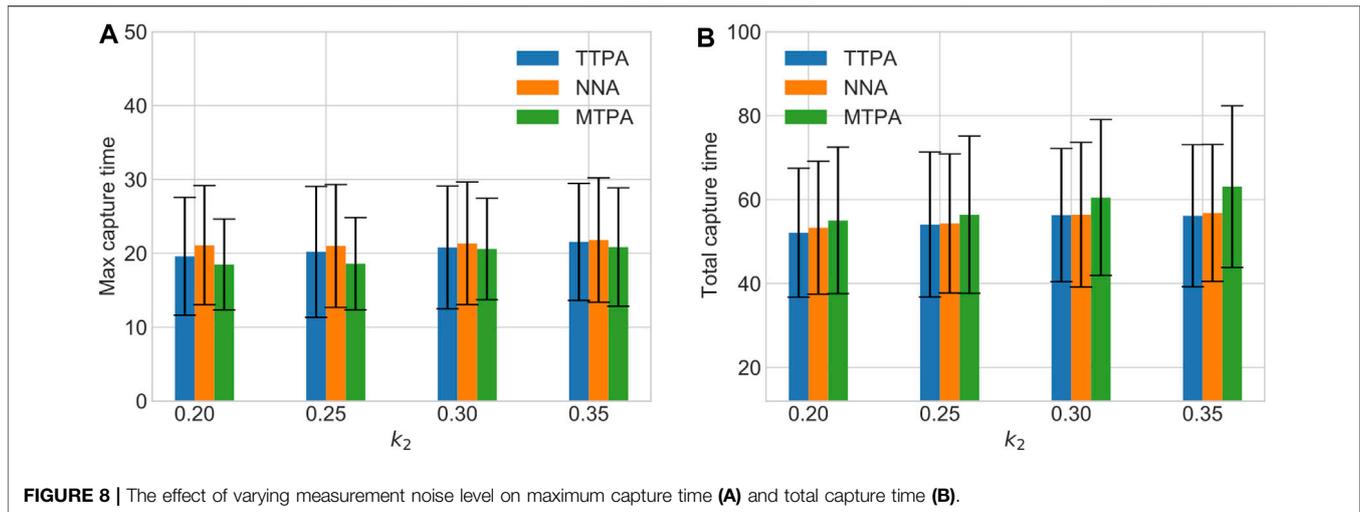
With these modifications, an assignment for the redundant pursuers can be found that minimizes the maximum expected capture time instead of total expected capture time. We call this redundant pursuer assignment algorithm “Maximum Time minimization Redundant Pursuer Assignment” (MTRPA).

### 5.4 Evader’s Estimation of Pursuer Assignment

Knowing the assignment strategy used by the pursuers, but the pursuers represented by the probability distributions  $\{q_i\}_{i \in C_r}$ , the evaders use the exact same assignment algorithm to estimate which pursuer is being assigned to it. The only difference is that in **Algorithm 2** the elements in the input,  $T$ , are sample travel times that are computed by sampling points,  $r_i$ , from the probability distribution,  $q_i$ , for all  $i \in C_r$ , and then computing  $\tau_{ij} = \frac{1}{v_i} d_g(r_i, y_j)$  as before. The assignment thus estimated is used by the evaders in computing their control as well as for updating the pursuers’ distributions,  $\{q_i\}_{i \in C_r}$ , as described in **Sections 4.1.1** and **4.2.2** respectively.

## 6 RESULTS

For the sensor models,  $f, h$ , we emulate sensing electromagnetic radiation in the infrared or radio spectrum emitted by the evaders/pursuers. Wi-fi signals and thermal signatures are such examples. For simplicity, we ignore reflection of the radiation from surfaces, and only consider a simplified model for transmitted radiation. If  $I_{r,y}$  is the line segment connecting the source,  $y$ , of the radiation to the location of a



sensor,  $r$ , and is parameterized by segment length,  $l$ , we define effective signal distance,  $d_{\text{eff}}(r, y) = \int_{L_y} \rho(l) dl$ , where  $\rho(l) = 1$  in obstacle-free space, and  $\rho_{\text{obs}} > 1$  inside obstacles to emulate higher absorption of the signal. The signal space,  $\mathbb{S} = \mathbb{R}_+$ , is the space of intensity of the measured radiation, and  $f_r$  and  $h_y$  are normal distributions over  $\mathbb{S}$  with mean  $\frac{k_1}{d_{\text{eff}}(r, y)}$  and standard deviation  $\sigma = k_2 d_{\text{eff}}(r, y)$  to emulate inverse decay of signal strength and higher noise/error for larger separation (we truncate the normal distribution at zero to eliminate negative signal values). In all our experiments we chose  $\rho_{\text{obs}} = 3$ ,  $k_1 = 10$ . We also fix  $k_2 = 0.3$ , except in the experiments in **Figure 8**, where we evaluate the performance with varying noise level (varying  $k_2$ ).

The motion models for predicting the probability distributions are chosen as described in Section 4.1.2 and 4.2.2.

For the parameter we choose  $\epsilon(y) \in (0, 0.3)$  (in **Equation 6**) depending on whether or not  $y$  is close to an obstacle. The pursuer (resp. evader) choose  $\sigma_j = 0.3$  (resp.  $\sigma_i = 0.3$ ) for modeling the uncertainties in the evaders' (resp. pursuers') estimate of the pursers' (resp. evaders') positions.

We Compared the Performance of the Following Algorithms

- **Total Time minimizing Pursuer Assignment (TTPA):** This assignment algorithm uses the basic Hungarian algorithm for computing the initial assignment  $\mathcal{A}_0$ , and uses the TTRPA algorithm (**Algorithm 2**) for the assignment of the redundant pursuers at every time step. Thus the algorithm seeks to minimize the *total expected capture time* (i.e., sum of the times to capture each evader).
- **Maximum Time minimizing Pursuer Assignment (MTPA):** This assignment algorithm uses the modified Hungarian algorithm described in **Section 5.1.1** for computing the initial assignment  $\mathcal{A}_0$ , and uses the MTRPA algorithm (**Section 5.3.1**) for the assignment of the redundant pursuers at every time step. Thus the algorithm seeks to minimize the *maximum expected capture time* (i.e. time to capture the last evader).

- **Nearest Neighbor Assignment (NNA):** In this algorithm we first construct a  $|C_r| \times |C_e|$  matrix of expected pursuer-to-evader capture times. An assignment is made corresponding to the smallest element of the matrix, and the corresponding row and column are deleted. This process is repeated until each evader gets a pursuer assigned to it. Then we start the process all over again with the unassigned pursuers and all the evaders, and the process continues until all the pursuers are assigned.

We evaluated the algorithms in two different environments: Game maps “AR0414SR” and “AR0701SR” from 2D Pathfinding Benchmarks (Sturtevant, 2012) see **Figure 6**. For different pursuer-to-evader ratios in these environments, we ran 100 simulations each. For each simulation, in environment “AR0414SR”, the initial positions of pursuers and evaders were randomly generated, while in environment :“AR0701SR” the initial position of the pursuers were randomly generated in the small central circular region and the initial position of the evaders were randomly generated in the rest of the environment. For each generated initial conditions we ran the three algorithms, TTPA, MTPA and NNA, to compare their performance.

- 1) Max capture time in “AR0414SR”
- 2) Max capture time in “AR0701SR”
- 3) Total capture time in “AR0414SR”
- 4) Total capture time in “AR0701SR”

**Figure 7** shows a comparison between the proposed pursuer assignment algorithms (TTPA and MTPA) and the NNA algorithm for the aforementioned environments. From the comparison it is clear that the MTPA algorithm consistently outperforms the other algorithms with respect to the maximum capture time (**Figure 7A**), while TTPA consistently outperforms the other algorithms with respect to the total capture time (**Figure 7C**). In addition, **Table 1** shows win rates of TTPA and MTPA over NNA (for TTPA this is the proportion of simulations in which the total capture time for TTPA was lower than NNA, while for MTPA this is the proportion of simulations in which the

total capture time for MTPA was lower than NNA). TTPA has a win rate of around 60%, and MTPA has a win rate of over 70%.

Clearly the advantage of the proposed greedy supermodular strategy for redundant pursuer assignment is statistically significant. Unsurprisingly, we also observe that increasing the number of pursuers tends to decrease the capture time.

- 1) Max capture time in “AR0414SR” with 7 pursuers and 5 evaders.
- 2) Total capture time in “AR0414SR” with 7 pursuers and 5 evaders.

**Figure 8** shows a comparison of the total and maximum capture times with varying measurement noise level (varying  $k_2$ ) in the environment “AR0414SR” with a fixed number of pursuers and evaders, and with 20 randomly generated initial conditions. As expected, higher noise leads to more capture time for all the algorithms. However MTPA still outperforms the other algorithms w.r.t. maximum capture time, while TTPA outperforms the other algorithms w.r.t. the total capture time.

## 7 CONCLUSION AND DISCUSSIONS

In this paper, we considered a pursuit-evasion problem with multiple pursuers, and multiple evaders under uncertainties. Each type of agent (pursuer or evader) represents the individuals of the other type using probability distributions that they update based on known control strategies and noisy sensor measurements. Markov localization is used to update a probability distributions. The evaders use a control strategy to actively evade the pursuers, while each pursuer use a control algorithm based on Theta\* search for reducing the expected distance to the probability distribution of the evader that it’s pursuing. We used a novel redundant pursuer assignment algorithm which utilizes an excess number of pursuers to minimize the total or maximum expected time to capture the evaders. Our simulation results have shown a consistent and statistically significant reduction of time to capture when compared against a nearest-neighbor algorithm.

## REFERENCES

- Agmon, N., Fok, C.-L., Emaliah, Y., Stone, P., Julien, C., and Vishwanath, S. (2012). “On Coordination in Practical Multi-Robot Patrol,” in 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, May 14–18, 2012, 650–656. doi:10.1109/ICRA.2012.6224708
- Agmon, N., Kraus, S., Kaminka, G. A., and Sadv, V. (2009). “Adversarial Uncertainty in Multi-Robot Patrol,” in Twenty-First International Joint Conference on Artificial Intelligence, Pasadena, CA, July 11–17, 2009.
- Barshan, B., and Durrant-Whyte, H. F. (1995). Inertial Navigation Systems for mobile Robots. *IEEE Trans. Robot. Automat.* 11, 328–342. doi:10.1109/70.388775
- Bhattacharya, S., Ghrist, R., and Kumar, V. (2014). Multi-robot Coverage and Exploration on Riemannian Manifolds with Boundaries. *Int. J. Robotics Res.* 33, 113–137. doi:10.1177/0278364913507324
- Burgard, W., Fox, D., Hennig, D., and Schmidt, T. (1996). “Estimating the Absolute Position of a mobile Robot Using Position Probability Grids,” in Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2, Portland, OR, August 4, 1996 (Palo Alto, CA: AAAI Press), 896–901. AAAI’96.
- Chung, T. H., Hollinger, G. A., and Isler, V. (2011). Search and Pursuit-Evasion in mobile Robotics. *Auton. Robot* 31, 299–316. doi:10.1007/s10514-011-9241-4

We considered a very complex problem setup that is not only stochastic in nature (each type of agent representing the other type of agents using probability distributions that are updated using a Markov localization model on a graph), but the environment is non-convex (due to presence of obstacles). While a general stability or convergence guarantee is extremely difficult, if not impossible, in such a complex problem setup, we can consider a simplified scenario for observing some of the stability and convergence properties of the control algorithm used by the pursuers. Such a simplified analysis has been provided in the **Appendix** below.

## DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## AUTHOR CONTRIBUTIONS

LZ was responsible for implementing the algorithms described in the paper, running simulations, generating numerical results as well as drafting majority of the results section. AP was responsible for developing the redundant pursuer assignment algorithm, overseeing its integration with the multi-agent control algorithms, and drafting the section on redundant pursuer assignment algorithm. SB was responsible for the development of pursuer and evader control algorithms, probabilistic representation and estimation algorithms, algorithms for initial pursuer-to-evader assignment, drafting of the corresponding technical sections in the paper, and overseeing the implementation and integration of all the different algorithmic components of the paper. All three authors contributed equally to the final writing and integration of the different sections of the paper, including the introductory and the conclusion sections.

- Cortes, J., Martinez, S., Karatas, T., and Bullo, F. (2004). Coverage Control for mobile Sensing Networks. *IEEE Trans. Robot. Automat.* 20, 243–255. doi:10.1109/tra.2004.824698
- Fox, D., Burgard, W., Dellaert, F., and Thrun, S. (1999a). “Monte Carlo Localization: Efficient Position Estimation for mobile Robots,” in Proceedings of the National Conference on Artificial Intelligence (Palo Alto, CA: AAAI), 343–349.
- Fox, D., Burgard, W., and Thrun, S. (1998). Active Markov Localization for mobile Robots. *Robotics Autonomous Syst.* 25, 195–207. doi:10.1016/s0921-8890(98)00049-9
- Fox, D., Burgard, W., and Thrun, S. (1999b). Markov Localization for mobile Robots in Dynamic Environments. *jair* 11, 391–427. doi:10.1613/jair.616
- Hespanha, J. P., Kim, H. J., and Sastry, S. (1999). “Multiple-agent Probabilistic Pursuit-Evasion Games,” in Decision and Control, 1999. Proceedings of the 38th IEEE Conference on, Phoenix, AZ, December 7–10, 1999 (IEEE) 3, 2432–2437.
- Hespanha, J. P., Prandini, M., and Sastry, S. (2000). “Probabilistic Pursuit-Evasion Games: A One-step Nash Approach,” in Decision and Control, 2000. Proceedings of the 39th IEEE Conference on, Sydney, Australia, December 12–15, 2000 (IEEE) 3, 2272–2277.
- Hollinger, G., Kehagias, A., and Singh, S. (2007). “Probabilistic Strategies for Pursuit in Cluttered Environments with Multiple Robots,” in Robotics and

- Automation, 2007 IEEE International Conference on, Rome, Italy, April 10–14, 2007 (IEEE), 3870–3876. doi:10.1109/robot.2007.364072
- Khan, A., Rinner, B., and Cavallaro, A. (2016). Cooperative Robots to Observe Moving Targets: Review. *IEEE Trans. Cybernetics* 48, 187–198. doi:10.1109/TCYB.2016.2628161
- Makkapati, V. R., and Tsiotras, P. (2019). Optimal Evading Strategies and Task Allocation in Multi-Player Pursuit-Evasion Problems. *Dyn. Games Appl.* 9, 1168–1187. doi:10.1007/s13235-019-0031910.1007/s13235-019-00319-x
- Munkres, J. (1957). Algorithms for the Assignment and Transportation Problems. *J. Soc. Ind. Appl. Math.* 5, 32–38. doi:10.1137/0105003
- Nagaty, A., Thibault, C., Trentini, M., and Li, H. (2015). Probabilistic Cooperative Target Localization. *IEEE Trans. Automat. Sci. Eng.* 12, 786–794. doi:10.1109/TASE.2015.2424865
- Nash, A., Daniel, K., Koenig, S., and Felner, A. (2007). “Theta\*: Any-Angle Path Planning on Grids,” in *AAAI* (Palo Alto, CA: AAAI Press), 1177–1183.
- Nash, A., Koenig, S., and Likhachev, M. (2009). “Incremental Phi\*: Incremental Any-Angle Path Planning on Grids,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, Pasadena, CA, July 11–17, 2009, 1824–1830.
- Nash, A., Koenig, S., and Tovey, C. (2010). “Lazy Theta\*: Any-Angle Path Planning and Path Length Analysis in 3d,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, Atlanta, Georgia, July 11–15, 2010. 24.
- Oyler, D. W., Kabamba, P. T., and Girard, A. R. (2016). Pursuit-evasion Games in the Presence of Obstacles. *Automatica* 65, 1–11. doi:10.1016/j.automatica.2015.11.018
- Pierson, A., Wang, Z., and Schwager, M. (2017). Intercepting Rogue Robots: An Algorithm for Capturing Multiple Evaders with Multiple Pursuers. *IEEE Robot. Autom. Lett.* 2, 530–537. doi:10.1109/LRA.2016.2645516
- Prorok, A. (2020). Robust Assignment Using Redundant Robots on Transport Networks with Uncertain Travel Time. *IEEE Trans. Automation Sci. Eng.* 17, 2025–2037. doi:10.1109/tase.2020.2986641
- Rimon, E., and Koditschek, D. E. (1992). Exact Robot Navigation Using Artificial Potential Functions. *IEEE Trans. Robot. Automat.* 8, 501–518. doi:10.1109/70.163777
- Shah, K., and Schwager, M. (2019). “Multi-agent Cooperative Pursuit-Evasion Strategies under Uncertainty,” in *Distributed Autonomous Robotic Systems* (Springer), 451–468. doi:10.1007/978-3-030-05816-6\_32
- Shkurti, F., Kakodkar, N., and Dudek, G. (2018). “Model-based Probabilistic Pursuit via Inverse Reinforcement Learning,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, May 21–25, 2018 (IEEE), 7804–7811. doi:10.1109/icra.2018.8463196
- Sturtevant, N. R. (2012). Benchmarks for Grid-Based Pathfinding. *IEEE Trans. Comput. Intell. AI Games* 4, 144–148. doi:10.1109/tciaig.2012.2197681
- Talmor, N., and Agmon, N. (2017). “On the Power and Limitations of Deception in Multi-Robot Adversarial Patrolling,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, Melbourne, Australia, August 19, 2017, 430–436. doi:10.24963/ijcai.2017/61
- Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.
- Zhang, L., Prorok, A., and Bhattacharya, S. (2019). “Multi-agent Pursuit-Evasion under Uncertainties with Redundant Robot Assignments: Extended Abstract,” in *IEEE International Symposium on Multi-Robot and Multi-Agent Systems*, New Brunswick, NJ, August 22–23, 2019. Extended Abstract. doi:10.1109/mrs.2019.8901055
- Zhang, W. (2007a). A Probabilistic Approach to Tracking Moving Targets with Distributed Sensors. *IEEE Trans. Syst. Man. Cybern. A.* 37, 721–731. doi:10.1109/tsmca.2007.902658
- Zhang, W. (2007b). A Probabilistic Approach to Tracking Moving Targets with Distributed Sensors. *IEEE Trans. Syst. Man. Cybern. A.* 37, 721–731. doi:10.1109/TSMCA.2007.902658

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher’s Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Zhang, Prorok and Bhattacharya. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

## APPENDIX: SIMPLIFIED THEORETICAL ANALYSIS

Suppose evader  $j$  is assigned to pursuer  $i$  and this assignment does not change. We consider the case when the evader's maximum speed is negligible compared to the pursuer's speed, as a consequence of which we make the simplifying assumption that the evader is stationary. The first observation that we can make is that with the stationary evader, the probability distribution for the evader's pose is updated according to  $\mathbf{p}_j^t = D^{t-1} \mathbf{p}_j^{t-1}$  (see Eq. 2), where  $p_j^t$  is a column vector containing the probability values over  $V$ , and  $D^{t-1}$  is a diagonal matrix that depends on the signal received as well as the probability distribution at the time-step such that the net probability always adds up to 1. It's easy to observe that a fixed point of this iteration is a distribution in which all the probability is concentrated on a single vertex, to which the iteration will converge. If the measurement model is unbiased, that vertex would be the vertex on which the actual evader resides.

Hence, after a sufficiently long period of time the evader is fully localized. The control law in (9), by construction, simply becomes following the negative of the gradient of the square of the geodesic distance to the evader (see first paragraph of Section 4.2). This ensures that the geodesic distance to the evader is decreased at every time-step (formally, the geodesic distance can be considered as a Lyapunov functional candidate the time derivative of which is always negative and zero when the pursuer and the evader are at the same location), hence ensuring the eventual capture of the evader. We summarize this simplified analysis under the following proposition:

**Proposition (informal):** For a fixed pursuer-to-evader assignment, if the evader's maximum speed is negligible compared to the pursuer's speed, and if the sensing model for the sensor onboard the pursuer is unbiased, after a sufficiently long period of time the control law in (9) will make the pursuer's position asymptotically converge to the position of the evader.