2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Valuing Attrition in a Fleet of Robots Used as Path-Based Sensors for Gathering Information in a Communications Restricted Environment

Loy McGuire[1], Michael Otte[2], and Donald Sofge[3]

***Abstract*— In this paper we propose a new algorithm for robots searching a hazardous, communications-denied area to gather information using a robot fleet that has a limited number of agents. The centralized algorithm uses robot survival along search paths as a sensor event for a distributed sensor network. As agents are lost to hazards, the search behavior adjusts to prioritize agent longevity in order to maximize information gain. In the past, related work solving this problem has assumed an infinite number of agents. In contrast, we assume that the number of agents is finite. We use Bayesian inference to update target and hazard belief maps of an area using data from the probability of survival of prior agents' paths as well as sensor readings from the agents along those paths. Using those belief maps, the algorithm can construct paths that maximize information gain, in expectation, while taking into account the predicted decrease in future information collected when losing an agent. This behavior increases the likelihood that agents survive longer, allowing them to collect more data.**

Using simulations with various fleet sizes and probabilities for hazards disabling agents, we compare our algorithm to work that does not account for attrition. The results show an increase in the longevity of the fleet when hazards are more effective at disabling agents. In nearly all cases, this contributes to an increased rate in information gain when the fleet size is small. Small sized fleets, in our case 10 or less agents, do not meet a threshold of collected information necessary to direct agents away from hazards. Large fleets, over 200 agents in our scenario, collect most of the information before Our algorithm causes a noticeable change in agent behavior (as compared to existing techniques). We find that the proposed method provides the greatest advantage for mid-sized fleets, between 20 and 100 agents, and when hazards have an increased probability of immobilizing agents.

I. Introduction

Searching for hazards in an environment has the effect of exposing agents to those hazards. A robot exposed to a hazard might be disabled by the hazard before the robot is able to communicate information about the hazard's location. For example, when continuous communication from a robot abruptly stops during a search, we can infer a hazard might be near the location from which the robot last communicated. However, if the agent is not able to communicate during the search, such as in a cave or underwater, then this strategy does not work.

This work is supported by ONR grants N0001420WX01827 and N00014-20-1-2712. The views, positions, and conclusions contained in this document are solely those of the authors and do not explicitly represent those of ONR.

[1]Loy McGuire is with the Distributed Autonomous Systems Section, U.S. Naval Research Laboratory, Washington, DC 20375 USA loy.j.mcguire.civ@us.navy.mil

[2]Michael Otte is with the University of Maryland otte@umd.edu

[3]Donald Sofge is with the Distributed Autonomous Systems Section, U.S. Naval Research Laboratory, Washington, DC 20375 USA donald.a.sofge.civ@us.navy.mil

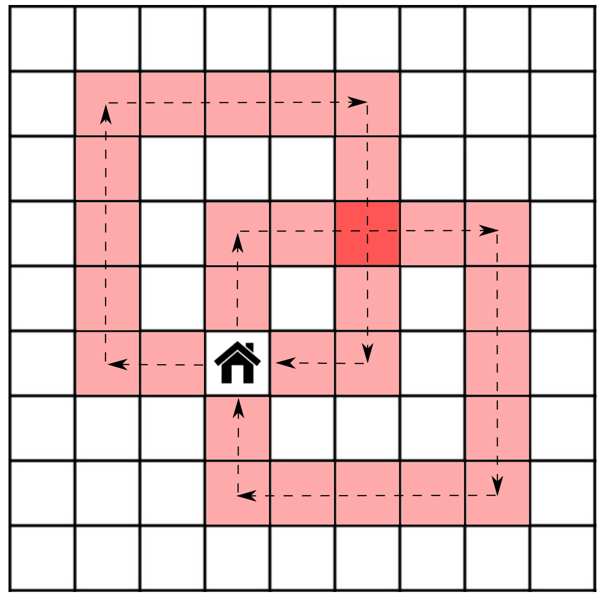

Fig. 1. Two paths where robots didn't return. Their intersection points show a higher probability of a hazard being in those cells.

Previous works [1], [2] use path-sensors to find hazards in communications-denied spaces. Path-sensors construct paths *a priori* and a sensor reading is either the agent returning or the agent not returning. As shown in Figure 1, the intersection of paths that agents did not return from indicate a higher probability of a hazard being in the intersection. The previous work assumes the availability of infinitely many agents.

Our work assumes a fleet consisting of a finite number of agents. Once all agents have been disabled, we are no longer able to collect information. Thus, losing an agent results in an expected decrease in the fleet's capacity to collect information. The change in the fleet's capacity to collect information is the value a single agent provides the fleet. In this paper we explore a method for constructing new search paths that considers this value (ability to perform future search) when determining the risk of loosing an agent along a particular path.

Our method works by sending out agents one at a time from a location where the agents are able to relay information, an uplink point. The central system creates a search path *a priori* and sends one agent to search along that path. The system updates beliefs about the environment based on information gained by the agent returning or not returning. The system then constructs a new search path and the process repeats.

Search paths are constructed by attempting to maximize the information gained while moving along a path. We quantify the amount of information in a path using information theory, and attempt to maximize the mutual information gained between our sensor readings and prior beliefs. The expected information gained in a cell is compared with the risk to the fleet's capabilities to collect information if an agent is lost. The comparison is used as a basis for a risk-reward function to create paths that maximize the amount of information we expect to collect.

The reward components of the function are composed of information about benign targets and hazards. An agent will take a target sensor reading in each cell along the search path. If the agent returns to the uplink point, the agent uploads sensor readings to the system. The sensor readings are lost if the agent does not return. Hazard information is gathered through the path-sensors mentioned previously. Hazards are never sensed, only when an agent is lost.

We use separate belief maps consisting of cells associated with locations in the environment to represent target and hazard information. Each cell in a target or hazard belief map contains a probability a target or hazard is in the cell's location, respectively. The system updates the target belief map using a standard Bayesian update for each cell the agent visits according to the sensor readings received in each respective cell. For the hazard belief map, the hazard probability in each cell in the search path updates according to whether the agent returns to the uplink point. This is similar to a Bayesian update that is applied along an entire path based on a path-sensor event.

The risk in the risk-reward function is the fleet's capacity of information collection that is lost when an agent dies. To find an agent's value, the current information available for collection in the environment is divided by the number of agents at the time the search path is constructed.

To construct the search path, the algorithm examines every reachable cell at the next time step. The algorithm accounts for every type of sensor event that may occur: if the target is or is not found and if the agent survives or dies, and calculates how the information of the cell changes for each event. The expected information-gain is found by summing the product of the probability and the respective changes in information of each event. The product of the probability an agent dies in the cell and the value of that agent is subtracted from the expected information gain to find the total expected change in information for searching a cell.

The algorithm then assigns the cell a parent cell at the current time step by examining which potential parent cell is part of the path with the most total expected information gained. This process repeats until all reachable cells across all time steps have been examined and a list of cells and their parent cells creates a search path from the start to the goal.

Our work contributes the inclusion of the valuation of an agent's capabilities over its lifespan when determining the path a robot will take. The formulation of this value aids probability-based searches with finite agents in hazardous environments. Also, the authors note that this method is a special case of a multi-agent system. Although agents do not communicate directly with other agents and only one agent is searching the environment at any time, the reliance on attritable agents to sense hazards requires multiple agents over separate time spans to collect information in a collaborative manner in order to create and update a shared belief map.

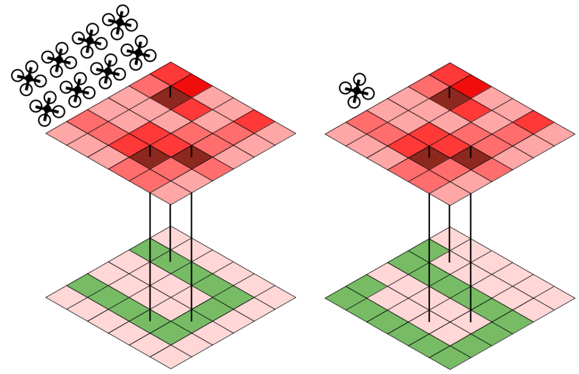

Fig. 2. In this figure the upper grids use darker red cells to represent a greater belief a cell contains a hazard. The lower grids use green cells to represent a path an agent might take to maximize information gain. The vertical lines show the locations of cells with the highest probabilities for hazards in both grids. A larger fleet searching an environment may choose to search a hazardous path when there are many agents available. This changes as the number of agents approaches zero. When agents are limited they are more likely to avoid cells they believe are hazardous.

II. Related Work

Our method falls into the broader category of target detection or search problems. Surveys about target search are analyzed in [3], [4]. Two noteworthy distinctions between our work and the majority of other works discussed in the surveys are the consideration of hazards in the environment that can disable agents and the denial of communications. Some works have considered one or the other, but the only related papers that consider both aspects are previous works that our method augments [1], [2].

In [1], [2], a framework is developed for searching a space that contains targets of interest and hazards that can disable agents conducting the search. Their method uses a recursive Bayesian update that informs a belief map of target positions using sensor readings and updates a separate belief map for hazards using the event of agents not returning as a sensor event. A path is built on the basis of following the gradient of mutual information but there are no communications outside of the designated information transfer location where agents are deployed from and return to. For this work, the paths are constructed *a priori* so the central system can narrow the position of failure to some cell along the path if an agent does not return. Our work goes further by incorporating a measure of risk to an agent within a fleet containing a finite number of agents.

Our work is one of several that uses information theory [5] as a basis for creating paths for the purpose of target search. A formal derivation of mutual information gradient ascent is introduced in [6], where information surfing is used to allow a multiagent system to share information during the search. In [7], [8], a similar method is used that considers a scenario where dangerous hazards are in the environment and agent attrition is expected. Their work uses the last known positions of lost agents to inform other agents of potential hazard positions, helping inform their search to minimize information loss. A decentralized version of this is used in [9], which is expanded in [10] to a three step look-ahead receding horizon search method.

Just as our method uses a Bayesian approach to better understand where our data indicates points of interest are, many other works use Bayesian methods for target search. Both [11] and [12] develop methods for fusing information from multiple Bayesian belief maps gathered using distributed algorithms that improves search efficiency. In [13], a Monte Carlo Tree Search (MCTS) is used for planning actions to try to maximize information gain as well as a Bayesian network for processing the data collected and building sensor belief knowledge of their robots. In [14], a cooperative search is considered where local occupancy grids are built to try to localize targets in the environment. In that work Bayesian updates are used to inform those grids. Our work follows [1], [2] in using a more standard Bayesian update for targets and a modified path-based Bayesian update for hazards.

A method that incorporates risk in their planning is found in [15], which uses a distributed system in order to search a space with multiple agents working at the same time. Their risk is assessed according to the distance between an agent and a hazard, whereas our method models risk as a factor of the information in the environment and the amount of agents that are available.

Using information theoretic approaches also has advantages when it concerns working with communication constrained problems. In [16], an information-theoretic co-evolutionary rolling horizon cooperative search algorithm is proposed that allows agents to adapt their behavior based on information currently available. They do so under constrained-communications. A single agent method for replanning with limited communications to a central system was developed in [17]. They plan by optimizing the amount of information gathered then continuously replan as new information is analyzed. In [18], a method is created to limit search time in micro aerial vehicle teams by considering the sensing and communication limitations of their agents and formulating the problem as a travelling salesman problem. They formulated their searches as either information gathering based or by having their algorithms be centralized or distributed. In [19], a decentralized algorithm is developed for multiagent path planning using submodular set optimization for gathering information to spread the work among the agents with limited communication. These methods limit the amount of communications but allow them to some degree. In contrast, our method assumes a complete lack of communications while the agents are deployed.

Algorithm 1 Iterative Information Path Planning for Targets and Hazards

Input: Prior Target Beliefs $\mathbf{X}$ and Hazard Beliefs $\mathbf{Z}$
Output: Iterative Sequence of Paths and Updates to $\mathbf{X}$ and $\mathbf{Z}$

```
for r = 1, 2, ... do
    while isAlive(r) do
        ζ = calculatePath(X, Z).
        Robot r attempts path ζ
        if θ_{ζ,alive} then
            X ← BayesianCellUpdates(X, Y_ζ)
            Z ← BayesianCellUpdates(Z, [0, ..., 0])
        else
            Z ← KilledOnPathUpdate(Z, ζ)
```

III. Notation

Consider the set of all targets $\mathbf{X}$ and the set of all hazards $\mathbf{Z}$ within the search space $\mathbf{S}$. Each s is a vector describing a point within $\mathbf{S}$. The set of uplink points $\mathbf{W}$ denotes every s that a robot $r \in \mathbf{R}$ may depart from, return to, and transfer data within, referred to as an uplink point w. Each path ζ, where $\zeta : [0,1] \rightarrow \mathbf{S}$, is a set of adjacent s such that $\zeta(0) = s_{start} = w_a$ and $\zeta(1) = s_{goal} = w_b$ where $w_a, w_b \in \mathbf{W}$. Each s can be described by a set of 2-D coordinates as well as a time integer $\tau \in Z^+$ contained within the set of space-time nodes $\mathbf{V}_{\mathbf{S}\times\mathbb{T}}$. Nodes are connected by edges $(v_i, v_j) \in \mathbf{E}_{\mathbf{S}\times\mathbf{T}}$ as part of a graph $\mathbf{G}_{\mathbf{S}\times\mathbf{T}}$ such that $\mathbf{G}_{\mathbf{S}\times\mathbf{T}} = (\mathbf{V}_{\mathbf{S}\times\mathbf{T}}, \mathbf{E}_{\mathbf{S}\times\mathbf{T}})$. $\zeta_{\hat{v}}$ contains the best current subpath from $\hat{v}$ to a goal. A path can be at most ℓ nodes in length, where ℓ represents the fuel a robot is able to use in a single deployment.

Due to sensor error, there are false positive and false negative readings. We model the errors using the probability of a false positive $p_{falsepos}$ or false negative $p_{falseneg}$ from our path based sensor. A false positive happens when the agent does not return but is not killed by a hazard either, $p_{falsepos} = p_{malfunc}$, and a false negative is when an agent encounters a cell with a hazard but does not die, $p_{falseneg} = 1 - p_{kill}$.

The set of sensor observations is denoted $\mathbf{Y}$. Each sensor observation is taken at a discrete time such that $t = 1, 2, 3, ...$ The events $\theta_{\zeta,dead}$ and $\theta_{\zeta,alive}$ represent whether a path-sensor is triggered or not triggered along path ζ, respectively. Θ_ζ is the random variable for probability of survival along path ζ.

IV. Methodology

Our algorithm seeks to maximize the amount of information that has been collected in the environment by minimizing the amount of information that is still available to collect. The information that has not been collected is a representation of uncertainty and is referred to as information entropy. The algorithm constructs paths attempting to search locations with the most information entropy, reducing the uncertainty of information that is gathered from the environment. The method in this paper differs from previous methods by accounting for the number of agents available to search when determining a path.

Due to the assumption that the number of agents are finite, they now have a potential value that can be associated with each one. By finding how much information is in the environment, we can divide that by the number of agents to find the average amount of information each agent can gain from the environment. Because we value finding information within the environment, each agent is valued by how much information they could potentially find. This is formulated by the following equation:

$$a_t = \frac{\Sigma_{j=1}^{m} h_j}{n_{agents} n_{\mathbf{V_S}}} \tag{1}$$

where a_t is the value of an agent at time step t, $n_{\mathbf{V_S}}$ is the number of cells within the physical environment, h_j is the entropy at cell j, and n_{agents} is the number of agents within the fleet at the time the algorithm is calculating the path.

In Algorithm 1, the system creates paths for robots as long as there are robots available to search the environment (lines 1-3). A robot returning from a search path is an event that can be sensed. A robot's return is sensed after the maximum amount of time has elapsed that the robot can be deployed and still return, determined by its fuel cost ℓ. The system records if the robot has returned or been deemed as disabled along the path (line 5). The system then updates beliefs based on sensor measurements and repeats the previous steps (lines 6-9).

The construction of a path is found in Algorithm 2. The algorithm assumes one or more uplink points exist which robots are able to depart from and traverse the environment on a predetermined path (lines 3-5). The paths are determined by a centralized system considering every potential movement a robot is able to make from every reachable cell in a specific time step (line 7). For each potential movement, the system calculates the expected information gained if it were to search the cell (lines 9-12). If the potential future cell has already been considered by another cell that could reach it, the potential future cell sets itself as a child node of whichever path has the most expected information gained from searching it (lines 13-15). Once all available cells have been searched, the starting uplink point with the path that minimizes the most entropy is chosen as the starting cell of the path (line 19). That cell and all of the child cells become the path that is returned (line 20).

The expected information gained from each move is assessed by the change in information from every potential event that could happen when searching a cell $\hat{v}_j$. The algorithm would look at the potential events occurring from taking a sensor reading of the cell, such as true and false positives as well as true and false negatives. The system would consider a potential event and calculate the hypothetical new amount of information entropy gained in that cell and subtract it from the current information entropy of that cell. This gives the expected information gain, $\Delta\hat{h}_{event}$, if that event happens. The information gained for each event is then multiplied by the probability of that event occurring

Algorithm 2 calculatePath($\boldsymbol{X}, \boldsymbol{Z}$)

Input: Target Beliefs $\boldsymbol{X}$ and Hazard Beliefs $\boldsymbol{Z}$
Output: Path ζ

1: **for all** $\hat{v}_i \in V_{\boldsymbol{S}\times\mathbb{T}\times\mathbb{H}}$ **do**
2: $\quad h_{\hat{v}_i} = -\infty$
3: **for all** uplink points $w \in W_{goal}$ **do**
4: $\quad \zeta_w \leftarrow \emptyset$.
5: $\quad$ InsertFIFOQueue(w)
6: **while** $\hat{v}_i \leftarrow$ PopFIFOQueue **do**
7: $\quad$ **for all** $(\hat{v}_i, \hat{v}_j) \in E_{\boldsymbol{S}\times\mathbb{T}}$ **do**
8: $\quad\quad \zeta \leftarrow (\hat{v}_i, \hat{v}_j) + \zeta_{\hat{v}_j}$
9: $\quad\quad \hat{h}^{\chi}_{live} \leftarrow \int_{x\in\chi} H(\boldsymbol{X}_{live})dx$
10: $\quad\quad (\boldsymbol{Z}_{live}, p^{alive}_{\zeta}) \leftarrow$ KilledOnPathUpdate($\boldsymbol{Z}, \zeta$)
11: $\quad\quad \boldsymbol{Z}_{killed} \leftarrow$ BayesianCellUpdates($\boldsymbol{Z}, [0, \ldots, 0]$)
12: $\quad\quad \hat{h}_{this} \leftarrow (c_Z(p^{alive}_{\zeta} H(\boldsymbol{Z}_{live}) + (1 - p^{alive}_{\zeta})H(\boldsymbol{Z}_{killed}) + c_X p^{alive}_{\zeta} \hat{h}^{\chi}_{live}) - (1 - p^{alive}_{\zeta})a_t$
$\quad\quad$ ▷ *Note: The subtraction of a_t is the major addition to the algorithm from [1], [2]*
13: $\quad\quad$ **if** $\hat{h}_{this} > h_{\hat{v}_i}$ **then**
14: $\quad\quad\quad \zeta_{\hat{v}_i} \leftarrow \zeta$
15: $\quad\quad\quad h_{\hat{v}_i} \leftarrow h_{this}$
16: $\quad\quad$ **for all** $(\hat{v}_k, \hat{v}_i) \in E_{\boldsymbol{S}\times\mathbb{T}}$ **do**
17: $\quad\quad\quad$ **if not** InQueue($\hat{v}_k$) **then**
18: $\quad\quad\quad\quad InsertFIFOQueue(\hat{v}_k)$
19: $w_{start} \leftarrow \arg\min_{w\in W_{start}} \hat{v}_j$
20: $\zeta \leftarrow \zeta_{w_{start}}$
21: **return** ζ

based off of prior sensor readings P_{event}. The values from each event being considered is summed together to find the total expected information $\hat{h}_{\hat{v}_j}$ gained by searching that cell. This is shown in the following equation:

$$\begin{aligned} \hat{h}_{\hat{v}_j} =& \Delta\hat{h}_{falsePos} * P_{falsePos} \\ &+ \Delta\hat{h}_{truePos} * P_{truePos} \\ &+ \Delta\hat{h}_{falseNeg} * P_{falseNeg} \\ &+ \Delta\hat{h}_{trueNeg} * P_{trueNeg} \end{aligned} \tag{2}$$

where $\hat{h}_{\hat{v}_j}$ is the expected information gained by searching cell j. Equation 2 is the basis for $\hat{h}_{this}$ (Algorithm 2 line 12), where $\hat{h}_{this}$ analyzes a path and $\hat{h}_{\hat{v}_j}$ examines a singluar cell.

We constrain the length of the path for each robot. In the real world this would translate to having a limited amount of fuel. Our scenario allows an agent to move 1 cell in any cardinal direction or diagonally from the cell it is currently in from one time step to the next. It may also choose to stay in the same cell it is in and measure it again in the next time step.

Each node is associated with a cell in the belief map for targets and a cell in the belief map for hazards which contain the probability of a target or hazard being in that cell, respectively. The sensor readings for targets are stored

Algorithm 3 KilledOnPathUpdate($\boldsymbol{Z}, \zeta$)

Input: Beliefs $\boldsymbol{Z}$ and Path ζ That Triggered Path-Based Sensor

Output: Posterior Beliefs $\mathbf{Z}$

$p_1^{survivedTo} \leftarrow 1$
for $k \leftarrow 1, \ldots, \ell$ **do**
  $i \leftarrow$ cell index in which k-th observation was made
  $p_k^{killedInGivenAt} \leftarrow (p_{kill} + p_{malfunc}(1 - p_{kill}))\boldsymbol{Z}[i] + p_{malfunc}(1 - \boldsymbol{Z}[i])$
  $p_{k+1}^{survivedTo} \leftarrow p_k^{survivedTo}(1 - p_k^{killedInGivenAt})$
  $\boldsymbol{Z}_k \leftarrow \boldsymbol{Z}$
  $\boldsymbol{Z}_k \leftarrow$ BayesianCellUpdates($\boldsymbol{Z}_k, [\mathbf{0}_{1:k-1}, 1]$)
$p_\zeta^{dead} \leftarrow \Sigma_{k=1}^{\ell} p_k^{survivedTo}$
$\boldsymbol{Z} \leftarrow \Sigma_{k=1}^{\ell} \frac{p_k^{survivedTo}}{p_\zeta^{dead}} \boldsymbol{Z}_k$
return $(\boldsymbol{Z}, (1 - p_\zeta^{dead}))$

Algorithm 4 BayesianCellUpdates($\mathbf{B}$,$\bar{\beta}$)

Input: Beliefs $\mathbf{B}$ and Standard Sensor Observation Sequence $\bar{\beta}$

Output: Posterior Beliefs $\mathbf{B}$

for $k = 0, \ldots, \ell$ **do**
  $i \leftarrow$ cell index in which k-th observation was made
  $\boldsymbol{B}[i] \leftarrow \mathbb{P}(B_i | \boldsymbol{B}[i], \bar{\beta}[k])$
return $\boldsymbol{B}$

onboard the robot until it is able to return to an uplink point. When the information is delivered, target belief maps are updated with the new information. If the robot does not return to an uplink point then the information about targets that it found while it was travelling is lost. The belief map for hazards in the environment is updated each time a robot is expected to return. If a robot does not return, all cells in the path the robot attempted to take are updated to increase the belief that a hazard is located in at least one of those cells. When robots return to the uplink point, all cells along the path are updated to reflect that hazards are less likely to be encountered in those cells. Either of these scenarios will change how much information is left in the cell after it has been searched.

Additionally, we can weight the sum of the target and hazard information. By weighting them using c_X and c_Z, we can prioritize how much we value information gained about one versus the other.

V. Experiments

We run a set of repeated trials to evaluate how differing the initial number of robots and the effectiveness of hazards for disabling agents affect the algorithm's performance. We then compare the performance of our algorithm to previous methods. Three algorithms are examined during the simulations, the first is the new algorithm which takes into account a fleet with a finite number of agents. We also run simulations using a previous algorithm that had assumed there are an infinite number of agents, enabling us to compare how accounting for

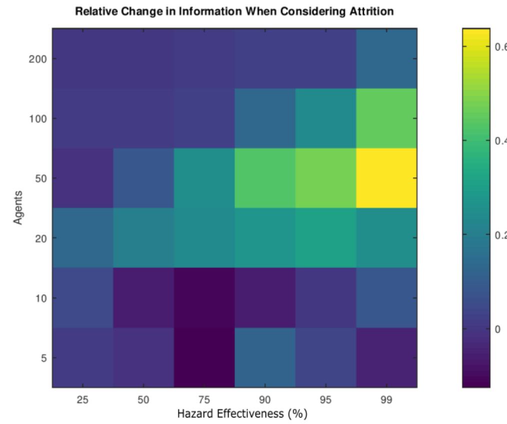

Fig. 3. The plot shows the ratio of improvement between the average of a set of experiments using the new algorithm compared to the old algorithm.

attrition changes the behavior of the fleet. The last behavior is a random walk to compare the algorithms with directed behavior against a baseline.

Between the sets of experiments, the variables for the number of agents in the fleet and the degree of effectiveness of hazards were changed.

The simulations were set up so that the agents were leaving from and returning to one stationary uplink point. The environment was a 10 by 10 grid with 4 hazards and 6 targets arbitrarily scattered throughout. Targets and hazards were allowed to be located within a single cell. Each agent was allowed to move between cells a maximum of 20 times before running out of fuel.

We used several metrics for examining how the behavior changed. We looked at how quickly agents were being lost and how quickly they were finding new information. We also collected data on how many time steps they took before the fleet had been completely exhausted. To ensure the consistency of the data collected, each set of experiments were run 20 times.

VI. Discussion of Results

In Fig. 3 the results of all the simulations appear showing the relative improvement in each set of experiments between our method which considers attrition as compared to the original method used in [1], [2]. For our figures we refer to our method as the new method and the method from [1], [2] as the old method. When the number of agents is too low, the results for 5 and 10 agents, there does not seem to be a noticeable pattern across the different degrees of hazard effectiveness in how much relative information the new algorithm collected. The relative difference at these low numbers of agents also seems to stay close to 0. For a fleet of 20 agents, the results show noticeable improvement at all levels of hazard effectiveness. From there the results show a strong correlation between an increase in the level of hazard effectiveness with an increase in information gathering performance.

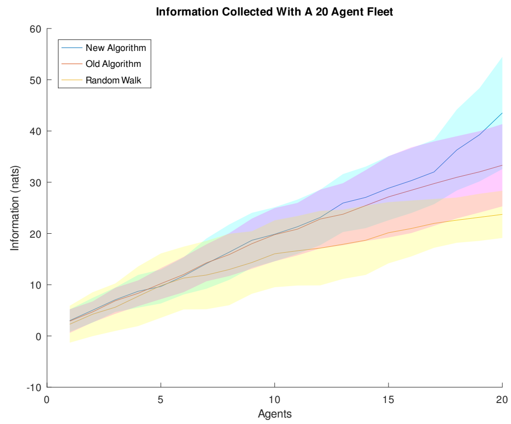

Fig. 4. The average amount of information collected by a specified number of agents. The new and old algorithms work similarly until the number of remaining agents reaches a certain threshold.

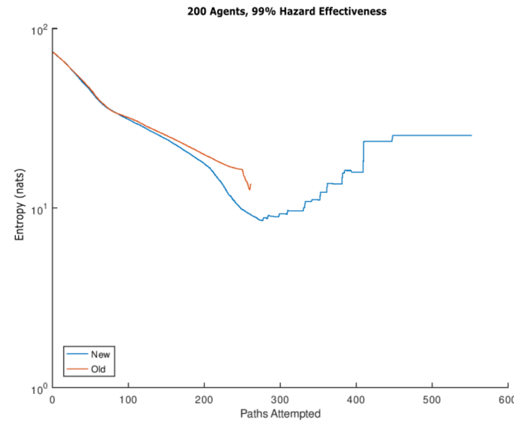

Fig. 5. The plot shows an example of notable behavior seen in large fleet sizes (typically above 100 agents) with highly effective hazards. Each value of entropy is an average of every trial that has reached that time step. The new algorithm has an upward trend toward the end, showing longer lasting trials tend to collect less information. The old algorithm has a downward trend, showing that longer lasting trials collect more information. As seen in Fig. 3, the average total information collected is still increased using our new method.

The best ratio of information gathered peaks when using 50 agents for the fleet in terms of relative efficiency. This is due to two factors. The first factor is having enough information about the effectiveness of a hazard for disabling agents for the cells in the environment in order for the new algorithm to provide noticeable improvement in information gathering. Lower numbers of agents already show that there is some threshold that needs to be reached in order for the algorithm to show successful results. The second factor depends on how much information is still available to be collected once that threshold has been met. While this can be seen in multiple plots found in Fig. 7 in the appendix, we highlight Fig. 4 to discuss this finding. In Fig. 4, the new and old algorithms collect similar levels of information until the amount of available agents in the fleet reaches a certain number. At that point the new algorithm begins collecting more information, likely due to taking safer routes leading to agents living longer. For larger numbers of agents, much of the information in the environment has been collected before reaching this critical number of agents remaining. At that point although agents are exhibiting the same risk-averse behavior, the ratio of information that can still be gathered compared to the information that has already been gathered is less than the same ratio in experiments with smaller fleets.

When results are examined by the number of paths attempted during a trial, we see different information. For experiments with small fleets, such as experiments with 20 agents, we see that the information collected is similar between the old and new algorithm across levels of hazard effectiveness. However, the new algorithm changes the fleet's behavior allowing it to survive longer and continue decreasing entropy past time steps that fleets using the old algorithm would have already died.

For larger fleets we see contrasting trends between the two methods, as can be seen in Fig. 5. The plots for both algorithms begin similarly. These plots averaged the amount of entropy remaining during each trial that had reached a certain number of paths attempted. Because trials end at different times, as shown in 6, as the plot moves along the X axis the number of trials being averaged are decreasing. Therefore, only the data for the longest living trial is displayed at the rightmost point of the plot. Towards the end of the old algorithm, the plot trends downward showing that the longest living trials gathered more data. We see the opposite for the new algorithm, where the end of the plot trends upwards showing that the longest living trials collected less information. This behavior results from agents using the new method deciding to avoid cells with a high probability of containing a hazard once the fleet size is sufficiently decreased, preventing them from collecting further information from those cells. It should be highlighted that even though there were trials for the new algorithm in Fig. 5 that ended with more entropy than some trials with the old algorithm, we observe from Fig. 3 that on average the new algorithm collected more information than the old algorithm for that set of experiments.

Our data also showed that the fleet's longevity benefited from our new algorithm. For experiments with less effective hazards, the results were relatively similar for small fleets but the new algorithm showed improvements for larger fleets. The contrast was much more drastic for experiments with more effective hazards. As seen in Fig. 6, for experiments with highly effective hazards there were sets of experiments where all trials using the new algorithm outlasted all trials using the old algorithm.

VII. Conclusion

This paper proposed a new algorithm that considers the potential loss of information collecting capability for a fleet of robots when creating a search path through a hazardous environment for an agent. Evaluation of our algorithm using Monte Carlo simulations showed that our algorithm influenced the behavior of the robots in the fleet in order to

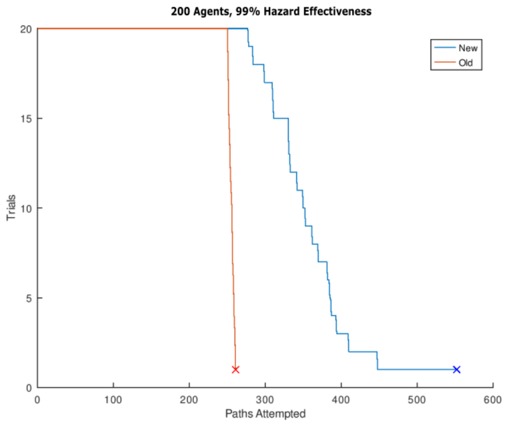

Fig. 6. This graph shows a comparison on the longevity both algorithms have. Larger fleets and more effective hazards show longer increases in longevity for agents.

extend the life of the fleet and recover more information from the environment than other methods. By incorporating a value to each agent tied to a hypothetical average amount of information it can gather from the environment, the algorithm takes into account the potential loss of the agent searching a cell based on the probability of a hazard being present within that cell. By finding the difference between the amount of information that is expected to be gained from searching a cell and the amount of information that is expected to be lost by losing the agent when searching the cell we have formed a value function that seeks to maximize information gained from a fleet with a finite number of agents.

References

[1] M. W. Otte and D. A. Sofge, "Path planning for information gathering with lethal hazards and no communication," in *Workshop on the Algorithmic Foundations of Robotics*, 2018.

[2] M. Otte and D. Sofge, "Path-based sensors: Paths as sensors, bayesian updates, and shannon information gathering," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 946–967, 2021.

[3] C. Robin and S. Lacroix, "Taxonomy on Multi-robot Target Detection and Tracking," in *Workshop on Multi-Agent Coordination in Robotic Exploration*, Prague, Czech Republic, Aug. 2014. [Online]. Available: https://hal.science/hal-01109186

[4] ——, "Multi-robot target detection and tracking: taxonomy and survey," *Autonomous Robots*, vol. 40, no. 4, pp. 729–760, Apr 2016. [Online]. Available: https://doi.org/10.1007/s10514-015-9491-7

[5] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[6] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1134–1154, 2012. [Online]. Available: https://doi.org/10.1177/0278364912452675

[7] M. Schwager, P. M. Dames, D. Rus, and V. R. Kumar, "A multi-robot control policy for information gathering in the presence of unknown hazards," in *International Symposium of Robotics Research*, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:24056004

[8] M. Schwager, P. Dames, D. Rus, and V. Kumar, *A Multi-robot Control Policy for Information Gathering in the Presence of Unknown Hazards*. Cham: Springer International Publishing, 2017, pp. 455–472.

[9] P. Dames, M. Schwager, V. Kumar, and D. Rus, "A decentralized control policy for adaptive information gathering in hazardous environments," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 2807–2813.

[10] P. Dames and V. Kumar, "Autonomous localization of an unknown number of targets without data association using teams of mobile sensors," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 850–864, 2015.

[11] J. Hu, L. Xie, K.-Y. Lum, and J. Xu, "Multiagent information fusion and cooperative control in target search," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1223–1235, 2013.

[12] G. A. Hollinger, S. Yerramalli, S. Singh, U. Mitra, and G. S. Sukhatme, "Distributed data fusion for multirobot search," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 55–66, 2015.

[13] A. Arora, P. M. Furlong, R. Fitch, S. Sukkarieh, and T. Fong, "Multi-modal active perception for information gathering in science missions," *CoRR*, vol. abs/1712.09716, 2017. [Online]. Available: http://arxiv.org/abs/1712.09716

[14] A. Khan, E. Yanmaz, and B. Rinner, "Information merging in multi-uav cooperative search," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 3122–3129.

[15] D. Vielfaure, S. Arseneault, P.-Y. Lajoie, and G. Beltrame, "Dora: Distributed online risk-aware explorer," 2021. [Online]. Available: https://arxiv.org/abs/2109.14551

[16] J. Berger and J. Happe, "Co-evolutionary search path planning under constrained information-sharing for a cooperative unmanned aerial vehicle team," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1–8.

[17] H. Yetkin, C. Lutz, and D. Stilwell, "Utility-based adaptive path planning for subsea search," in *OCEANS 2015 - MTS/IEEE Washington*, 2015, pp. 1–6.

[18] A. Khan, E. Yanmaz, and B. Rinner, "Information exchange and decision making in micro aerial vehicle networks for cooperative search," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 4, pp. 335–347, 2015.

[19] H. J. He, A. Koppel, A. S. Bedi, D. J. Stilwell, M. Farhood, and B. Biggs, "Decentralized multi-agent exploration with limited inter-agent communications," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5530–5536.

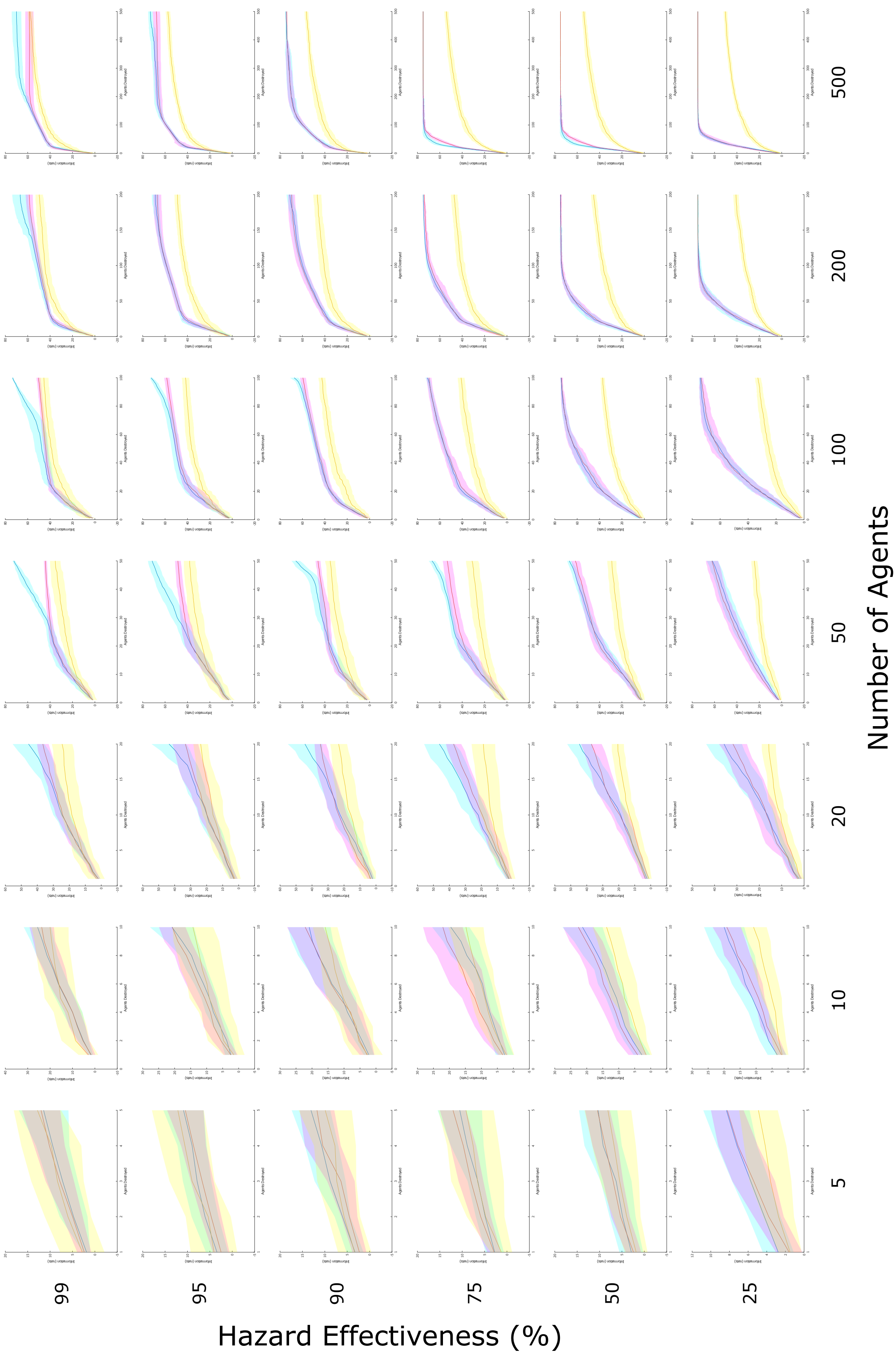

Fig. 7. Plots of experiment sets of differing agents from 5 - 500 and rates of hazard effectiveness from 0.25 - 0.99.