Distributed Estimation of an Uncertain Environment using Belief Consensus and Measurement Sharing

Rohan M. Seam and Michael J. McCourt

***Abstract*—Recent advances in technology have increased the capability of mobile platforms while decreasing the cost. It has become more feasible to create and deploy a large number of agents to cooperatively accomplish an objective. While multi-agent systems provide many advantages, such as lower cost and robustness to agent failure, there is a need for additional study of guiding principles for the design and test of these decentralized systems. The main contribution of this paper is a new estimation and path planning algorithm that can be used for better understanding of uncertain environments. The algorithm utilizes some existing estimation approaches including Bayesian fusion, measurement sharing on graph, and belief consensus. One new component of this approach is the reward-based path planning algorithm that incentivizes agents to plan their future path to collect the best local information as well as to avoid other agents to improve coverage in the environment. When agents plan ahead to collect more valuable information, the estimation error is reduced. This approach is studied in the context of measuring and estimating the state of a forest fire. Simulations were performed to test the effectiveness of this estimation algorithm compared to some existing approaches.**

I. INTRODUCTION

Advances in technology over the recent decades has enabled low-cost mobile platforms that are equipped with sensing, computational, and communication capability. This allows for multi-agent systems to be rapidly developed to solve new challenges and to improve on existing solutions. With these new advances, it is more important than ever for researchers to study the scientific principles of multi-agent systems, cooperative control, distributed systems, and emergent behavior. There are a variety of survey papers on existing approaches for these multi-agent systems [1]–[6].

In this paper, we are focused on teams of unmanned aerial vehicles (UAVs) that collect information in an uncertain and hazardous environment. Similar problems have been studied and applied to detecting clouds of hazardous chemicals [7] or to measure the spread of wildfires [8]. For the current paper, we are estimating the spread of a forest fire using UAVs to collect information about the fire that may be uncertain or incomplete. We will assume that each agent has a detection system that includes a video camera, computational capabilities, and an image detection algorithm. As this real system would not be perfectly accurate, we allow for false positives and negatives in the algorithm. The use of video surveillance for estimating the true state of the world has been used previously (see, e.g. [9] and [10]). Indeed, some researchers have demonstrated the efficacy of a decentralized UAV system versus a single aircraft system for surveillance and search operations [11].

Distributed control of multi-agent systems is a well-studied field with many promising approaches. One body of work uses the consensus algorithm applied to distributed rendezvous, flocking, and synchronization (see, e.g. [1], [3], [12]). A related approach involves using belief consensus to reach agreement by fusing estimated states between agents [8], [13]. Another decentralized control algorithm relies on randomized behavior that is constrained by local rules [7]. Other researchers utilize a reward function for incentivizing agents to make decisions that are better for the cooperative team goals [14]. For the goal of spreading agents out to sufficiently cover the environment, one bio-inspired approach has been to use only repulsive interactions between agents [15]. In that paper, agents do not use traditional communication networks but use *indirect communication* by sensing the position of other agents. A similar approach using indirect communication is presented in [16].

The main contribution of this paper is a novel approach to cooperative information gathering in an uncertain environment that synthesizes several existing approaches. This includes estimation algorithms and also path planning algorithms which can improve the quality of future measurements. The proposed method is compared to existing algorithms through simulation to determine the most effective solution for the stated problem. We are limiting our approach to a team of UAVs that can only sense locally and make decisions based on their individual estimates of the state of the world. We are inspired by the prior work in [13] to use belief consensus to fuse individual estimates of the state of the world to improve decision-making over time. As this communication between agents is implemented on a graph, we will also implement a method of sharing measurements between agents. While each agent can only measure the state of the fire at their current location, they can share measurements between agents which allows for a better global estimate of the true state of the environment. For path planning, we start with an algorithm where each agent uses its own estimated map to decide where to move locally to maximize information. This is similar to the approach of moving along the gradient of mutual information which was presented in [8]. By selecting paths that maximize the information gathered from measurements, the uncertainty of the estimated state is reduced. While this decision making algorithm works well in many scenarios, we have explored methods of improving it by making it a piece

R. M. Seam is with the Dept. of Computer Science and Systems at the University of Washington Tacoma, email: rseam@uw.edu . M. J. McCourt is with the Dept. of Electrical and Computer Engineering at the University of Washington Tacoma, email: mmccourt@uw.edu .

of a generalized reward function. Using this reward function, agents are incentivized to collect the best local measurements but they are also incentivized to spread out and collect unique information. This is done by the use of local sensing of other agents (indirect communication) and repulsive forces between agents.

The remainder of this paper is organized as follows. In Section II, we introduce the main problem that is studied in the paper. This includes the model of the forest fire spreading and the team of UAVs available to collect information about the environment. Section III introduces the Bayesian fusion estimation algorithm that is used to update individual agents' beliefs based on measurements acquired. This section also introduces the communication network, the sharing of information, and the belief consensus algorithm. Section IV introduces the UAV path planning and the reward function utilized to govern the decision making process of agents. In Section V, we provide simulated results of the proposed solution. Finally, the paper is concluded in Section VI.

II. Problem Scenario

While the approach presented in this paper is general and can be applied to modeling any uncertain and hazardous environment, we were inspired by recent events to apply our work to modeling the spread of large forest fires. With the recent increase in large fires throughout the world, there is a clear need for new advances in measuring and estimating forest fires. It is challenging to measure and estimate the state of a fire with a single UAV due to the size and spread rate of a large fire. This would require speed and sensing capabilities beyond what is currently available. An alternative approach is to employ a team of UAVs that can communicate and cooperate to more quickly and accurately estimate the state of the fire. With a more accurate estimate of the fire, resources such as water and fire retardant can be allocated more effectively to contain the fire. A better estimate has additional benefits such as providing better information for protecting firefighters and more effective evacuations. There are clearly many benefits to a decentralized, autonomous system that effectively estimates the state of a fire. The remainder of this section will provide an overview of a simplified fire-spread simulation model and the capabilities of a team of UAVs.

A. *Modeling Ground Truth*

In order to model the spread of a forest fire, we studied existing work in [17], [18]. These papers present a set of models that track the spread of fire based on a large number of parameters. These parameters include the type, density, and moisture of vegetation. The rate also depends on the temperature of the air, temperature of the ground surface, and a variety of variables related to wind conditions. Based on these detailed models, it can be determined that the majority of large forest fires spread at a rate in the range of 2 to 8 feet/min. Converting this range to meters gives a spread rate of 0.6 to 2.4 meters/min. To factor in the variability of fire spread rate due to all of these underlying factors, we chose to simulate the spread of a forest fire by randomizing the actual fire spread around an average rate.

In this paper, we discretize the environment to a grid-based model for ease of study. While the accuracy may be reduced, the size of the grid squares can be scaled down, and the number of squares scaled up, to get any desired level of accuracy. We chose a square grid with the same number of rows and columns. The location of a grid square can be referred to as (r, c). In our scenario, each grid square is 25 meters by 25 meters. The simulation is discretized both physically and temporally with a fixed timestep that represents about 5 seconds of time passing for the simulated world. This is a sufficiently high rate for the spread of the fire that is usually measured in minutes.

Initially, one grid square is uniformly randomly chosen to be the start of the fire. For a map that is G_s squares by G_s squares, the probability of selecting a given grid square is $\frac{1}{G_s^2}$. The fire may spread to any adjacent grid square with a probability dependent on the fire spread rate. At each timestep, there is a chance that the fire may not spread to any of the adjacent squares, but also a chance that the fire spreads to multiple adjacent squares at the same time. This simulates the variability in the spread of a forest fire due to the underlying parameters of the vegetation which may be unknown. We chose a fire spread rate of 2.1 meters/min as a value near the upper bound of the range given in previous work. The randomized spread rates can be determined by working backwards from 2.1 meters/min. Using the discretized simulation parameters, this can be converted to 0.007 grid squares per simulation step. This equates to a probability of 0.007 that the fire spreads in each direction at each timestep. A simulated test case of this fire spread model is shown in Figure 1. The fire started at a particular grid location and has now spread to the area shown in red. The portion of the map that is dark green is currently not on fire.

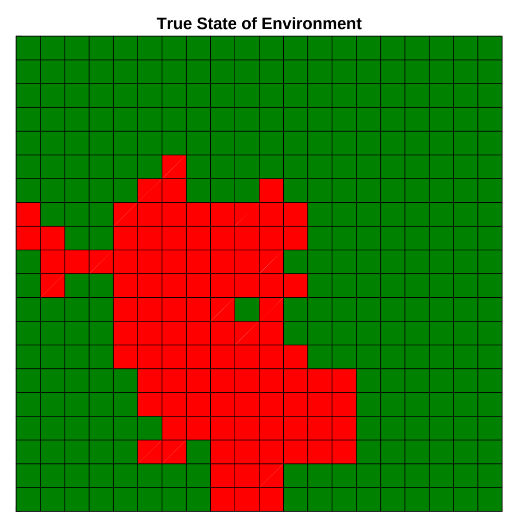

Fig. 1. The true state of each grid square is a binary variable representing the presence of fire (1) or the absence of fire (0). For visualizing the fire, the grid squares that contain fire are shown as red while the locations that are not on fire are displayed in green.

B. Modeling the UAV Team

As the fire spreads through the environment, a multi-agent team of UAVs is deployed to collect information about the fire. In current fire tracking and modeling, this is done in a centralized way where individual sensing is typically controlled by humans and manually entered into predictive models. The goal of the current paper is to provide a decentralized algorithm where individual agents take measurements, communicate, and make decisions based solely on the information that is available to that agent.

In a real scenario, UAVs would monitor the spread of a fire with cameras and image detection algorithms to measure the environment as they traverse it. These existing algorithms are not perfect and false measurements are produced based on a complex underlying process. To simulate these false measurements, we randomly generate measurement error in the form of false positives and false negatives. We model these random measurement errors using Bernoulli random trials. The false positive rate is given by P_{FP}, and the false negative rate is P_{FN}. The true state of the fire at a location (r, c) is either $T^{(r,c)} = 0$ (no fire) or $T^{(r,c)} = 1$ (fire). When agent i takes a measurement $M_i^{(r,c)}$ at that location, it is generated by the Bernoulli distribution.

$$P(M_i^{(r,c)}|T^{(r,c)} = 0) = \begin{cases} 1 - P_{FP} & \text{for } M_i^{(r,c)} = 0 \\ P_{FP} & \text{for } M_i^{(r,c)} = 1 \end{cases} \quad (1)$$

$$P(M_i^{(r,c)}|T^{(r,c)} = 1) = \begin{cases} P_{FN} & \text{for } M_i^{(r,c)} = 0 \\ 1 - P_{FN} & \text{for } M_i^{(r,c)} = 1 \end{cases} \quad (2)$$

The false positive rate and false negative rate may be different from each other and also different for each agent. While we did not that in this paper, it may be needed in a real scenario where UAVs are equipped with different sensing capabilities. Finally, based on the simulation discretization, the UAVs can move at a rate of one grid square per timestep. With a stepsize of 5 seconds, the UAV travel speed is fixed at 25 meters per 5 seconds or 18 km/hour.

III. Estimation Approach

The problem introduced in the previous section has a two part solution that involves each agent periodically estimating the true state of the forest fire and then planning future paths to collect the best local information in the environment. This section will cover the estimation algorithm that each agent uses to fuse all available information into a coherent belief of the state of the forest fire. This estimation algorithm includes individual agents collecting measurements about the true state of the forest fire, sharing this information with other agents and fusing these measurements, and fusing the current beliefs about the state of the fire between agents.

A. Bayesian Measurement Fusion

For each particular agent, the belief in the state of the forest fire can be represented as a matrix of probabilities. The belief of agent i can be represented by the matrix $B_i = \{b_i^{(r,c)}\}$. Each element $b_i^{(r,c)}$ corresponds to a probability that a location (r, c) is either partially or fully on fire. The matrix is initialized with probability 0.5 that each location has fire, which implies a probability of 0.5 of no fire present. This represents maximal uncertainty in the state of the environment. As the UAVs proceed through the environment, they take measurements at specific locations. Each measurement may increase or decrease the likelihood that the location is on fire which may increase or decrease the uncertainty of the estimate. As discussed before, the measurement model includes a possibility of false negatives and false positives with rates based on the camera properties and accuracy of those underlying algorithms.

When an agent visits a physical location, it takes a measurement of the current state of the fire in that grid square. The agent uses this measurement to update its current belief of the state of the fire. At time k the estimated state of the fire by agent i at location (r, c) can be denoted $b_i^{(r,c)}(k)$. While the true state of the fire is a binary variable (either on fire or not on fire), the estimate of the fire state is a continuously valued probability in the range $0 \leq b_i^{(r,c)}(k) \leq 1$. When a binary measurement $M_i^{(r,c)}(k)$ is taken at time k, the updated state of the fire at this location $b_i^{(r,c)}(k^+)$ can be calculated by using Bayes' rule.

$$P(b_i^{(r,c)}(k^+)|M_i^{(r,c)}(k)) = \frac{P(M_i^{(r,c)}(k)|b_i^{(r,c)}(k))P(b_i^{(r,c)}(k))}{P(M_i^{(r,c)}(k))} \quad (3)$$

The belief prior to the measurement is $P(b_i^{(r,c)}(k))$ while $P(b_i^{(r,c)}(k^+))$ is the posterior belief. The factor

$$\frac{P(M_i^{(r,c)}(k)|b_i^{(r,c)}(k))}{P(M_i^{(r,c)}(k))} \quad (4)$$

represents the measurement model which is developed offline using training data.

Between measurements, the estimated state of the fire at each grid location is updated according to a prediction model. The prediction model is based on the prior likelihood that the grid square contained fire and the estimated state of the neighboring squares. The adjacent grid squares for location (r, c) can be denoted G_a.

$$G_a = \{(r-1, c), (r+1, c), (r, c-1), (r, c+1)\} \quad (5)$$

When adjacent grid squares have a higher likelihood of containing fire, the probability that the current square contains fire increases over time to model the chance of the fire spreading. The prediction step can be summarized with the following equation where R_F represents the average fire spread rate.

$$P(b_i^{(r,c)}(k+1)|b_i^{(r,c)}(k)) = \frac{P(b_i^{(r,c)}(k)) + \sum_{(\tilde{r},\tilde{c})\in G_a} R_F b_i^{(\tilde{r},\tilde{c})}(k)}{1 + \sum_{(\tilde{r},\tilde{c})\in G_a} R_F b_i^{(\tilde{r},\tilde{c})}(k)} \quad (6)$$

In the absence of measurements, the likelihood that a location is on fire increases when adjacent locations have a high probability that they are on fire. Similarly, the likelihood that the location is on fire decreases when the probability of nearby locations being on fire is low.

B. Communicating Measurements between Agents

The goal of this work is to provide an approach for agents to cooperatively collect and share information to improve their decision-making capability. The next component of this approach is a method for agents to communicate and fuse their measurements. This communication strategy uses graph theory to model information sharing between agents. The following discussion will introduce graph theory and explain how it is used for communicating measurements.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph with set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$. Each edge can be denoted by $e_{ij} = \{v_i, v_j\}$ where $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$. We can refer to v_i and v_j as the *tail* and the *head*, respectively, of the edge. In this paper, all edges are undirected, i.e. if an agent shares its information with another agent, it will also receive that agent's information. Because the graphs are undirected, if $\mathcal{E}$ contains edge $\{v_i, v_j\}$, it also contains edge $\{v_j, v_i\}$. The degree of a vertex is defined as the number of agents that it communicates with and is denoted by $deg(v_i)$. If $N_v = |\mathcal{V}|$, then the degree matrix $\mathcal{D} \in \mathbb{Z}^{N_v \times N_v}$ (an N_v by N_v matrix of integers) is defined as $\mathcal{D}(\mathcal{G}) = \{d_{i,j}\}$. The element $\{d_{i,j}\}$ is in the i^{th} row and j^{th} column of the matrix.

$$d_{i,j} = \begin{cases} deg(v_i) & i = j \\ 0 & i \neq j \end{cases} \tag{7}$$

The adjacency matrix of $\mathcal{G}$ is denoted by A, where $A \in \mathbb{Z}^{N_v \times N_v}$. The elements of A can be specified by $A = \{a_{i,j}\}$ and the following equation.

$$a_{i,j} = \begin{cases} 1 & \{v_i, v_j\} \in \mathcal{E} \\ 0 & \{v_i, v_j\} \notin \mathcal{E} \end{cases} \tag{8}$$

If there is an edge between agent v_i and v_j, the index $a_{i,j} = 1$. If there is not an edge between the two, then $a_{i,j} = 0$. One important feature of the adjacency matrix is that it does not allow self-loops. In other words, there is no way for an agent to have communication with itself and $a_{i,i} = 0$ for all i. The graph Laplacian $\mathcal{L}$ of graph $\mathcal{G}$ is defined using the degree matrix and adjacency matrix.

$$\mathcal{L} = \mathcal{D} - A \tag{9}$$

An important property of $\mathcal{L}$ is that each row of the matrix sums to 0. A *path* is a sequence of vertices where there exists an edge from each vertex to the next. A graph is *connected* if it has at least one vertex and there exists a path between any two vertices. A *strongly connected* graph is one where there exists an edge from each vertex v_i to all other vertices v_j for all i and j. The diagonal elements of the graph Laplacian are positive and represent the degree of each agent. The other elements of the Laplacian matrix are either 0, indicating no communication between the two corresponding agents, or -1 indicating communication between those agents. When an agent communicates with another agent, the two are considered neighbors in the graph theoretic sense. The set of all neighbors of an agent i can be indicated by N_i. More detail about these concepts and notation can be found in [19].

At each simulation step, each agent i measures the state of the forest fire at its physical location. This measurement is communicated to all neighboring agents $j \in N_i$ on the graph $\mathcal{G}$. All neighboring agents apply this measurement to their own estimated state as if they measured it directly. This is done according to Bayes' rule as discussed in the previous subsection. For a graph with N agents, each agent may update their own belief matrix with 1 to N measurements, depending on the graph Laplacian.

C. Belief Consensus

Despite agents sharing information with their neighbors, the agents' beliefs of the state of the fire will diverge over time due to the differences in measurements taken both physically and temporally. In order to counteract this divergence, the last component of this estimation algorithm is belief fusion between agents using belief consensus. Belief consensus is an important innovation in estimation algorithms where agents directly share their belief of the state of the full environment with neighboring agents. The belief of agent i is a matrix of probabilities B_i representing the current estimated state of the fire for the entire environment. This belief is shared with other agents on the same graph with Laplacian $\mathcal{L}$. When an agent receives the belief matrix B_j of one of its neighbors, $j \in N_i$, the belief is fused with its own prior belief to arrive at a new posterior belief. This belief consensus algorithm follows the approach introduced in [13]. This process is repeated for all agents $j \in N_i$. This fusion is applied to update the belief of agent i before the fusion (k) to immediately after (k^+) using a weighted geometric mean according to the following equation.

$$B_i(k^+) = B_i(k) \prod_{j \in N_i} \left(\frac{B_j(k)}{B_i(k)} \right)^{\gamma} \tag{10}$$

The parameter γ determines the weighting placed on the prior belief of agent i and a complementary weight placed on the received belief of agent j. A graph has a maximum degree d_{max} that is defined by the following equation.

$$d_{max} = \max_{v_i \in \mathcal{V}} \{deg(v_i)\} \tag{11}$$

The value of γ must be chosen according to $0 < \gamma < \frac{1}{d_{max}}$, see [13] for proof. A γ value near the upper bound will result in faster convergence between differing beliefs while a smaller value will slow the rate of convergence between agents' beliefs.

The belief consensus approach can greatly reduce the difference in beliefs between agents, resulting in reduced estimation error for all agents. However, this algorithm requires significant resources both in communication and computation. We propose a modified form of this belief consensus algorithm where agents do not communicate their beliefs at every timestep, but at a reduced frequency that can be accommodated both by the communication network and

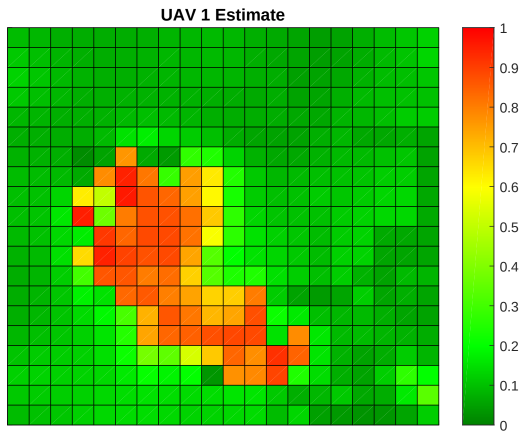

Fig. 2. This figure shows the estimated state of the fire as estimated by a particular UAV. The red squares indicate a high probability of a fire being present ($P > 0.9$) while the dark green squares indicate a low probability of fire ($P < 0.1$). The highest uncertainty squares are yellow to yellow-green.

the available computational resources. The exact rate can be determined based on the given application. Selecting this rate is discussed again in Section V. The belief map for one agent during one run of the estimation algorithm is displayed in Figure 2. This was taken from the same sample simulation and time as in Figure 1. The two figures can be visually compared to get a sense of the size of the error in the estimation algorithm. The color ranges from dark green (low probability of fire) to red (high probability of fire).

IV. Path Planning Algorithm

In this decentralized information gathering scenario, each UAV must make local decisions based on the information available from measurements, communication with other agents, and the estimation algorithm. The approach considered here is a reward function that incentivizes agents to make decisions that balance competing goals including collecting the best possible measurements and distancing from other agents to ensure better coverage of the environment. An appropriately chosen reward function should improve the estimation algorithm by increasing the value of the measurements taken which will reduce the average estimation error over time. In the problem scenario, each agent is iteratively collecting measurements and updating its estimate of the state of the fire according to the algorithm outlined in the previous section. The agent then determines possible future paths and ranks these paths according to the reward function. The path that has the highest reward is the one chosen by the agent. The reward function can be explained as a weighted average of two different path selection criteria.

The first path selection criterion is an algorithm that chooses paths based on an information theoretic approach. This approach calculates the value of a future measurement at a nearby location by using the current estimate of the state of the fire at that location. If the location has a high level of uncertainty, the value of that future measurement is higher. If the uncertainty is lower, the value of that potential measurement is lower. This uncertainty can be captured using the information theoretic concept of entropy. The entropy of a probabilistic belief can be calculated using the *Binary Entropy Function*.

$$H_b(p) = -plog_2(p) - (1-p)log_2(1-p) \qquad (12)$$

For our problem, p represents the probability that a given location is on fire, and $H_b(p)$ is the entropy of that location. As an example, entropy is maximized when the likelihood of a grid square containing fire has a probability of 0.5, which also represents maximal uncertainty of the true state. Measuring the location with the highest entropy will result in the largest reduction in overall entropy in the environment which provides the highest value of information. A reward function can be defined for agent i considering moving to and measuring a location (r, c).

$$R_{EM}(i, r, c) = H_b(b_i^{(r,c)}) \qquad (13)$$

This reward function incentivizes agents to visit physical locations with the highest uncertainty. When the simulation begins, all grid squares are initialized with probability 0.5. This function incentivizes agents to visit these locations that have not been measured yet. When the estimated state is more certain, e.g. $b_i^{(r,c)} < 0.1$ or $b_i^{(r,c)} > 0.9$, the value of that measurement is low. This function ensures that the UAVs maneuver through the environment based on which nearby locations are the most uncertain. This also ensures cooperation between the agents in the environment. When one agent measures a physical location, it reduces the reward provided to a neighboring agent. This encourages the other agent to visit grid squares that have not been recently measured.

The second path selection criterion is designed to encourage UAVs to spread throughout the environment to maximize the coverage of the space. The goal of increased coverage is to ensure that agents are collecting different measurements as well as physically covering the space. Increased coverage increases the likelihood that there is an agent near an area with rapidly changing state. In order to do this, it is assumed that agents can locally sense other agents in a small area and use this as a form of indirect communication. This can be seen as each agent having a ring of vision. When other agents are outside of this ring of vision, they are not considered. A reward function is introduced to penalize agents for traveling close to other agents. This takes the form of a repulsive force where the reward function decreases rapidly as agents move closer together. The repulsive force is inversely proportional to the square of the distance between agents. As agents get closer, the repulsive force is larger. Agents attempt to maximize this reward function by keeping a larger distance between them and other agents. In practice, this could be calculated in meters, but with this discretized simulation, this distance is calculated as a distance expressed in grid squares. The reward for agent i to visit and measure a location (r, c)

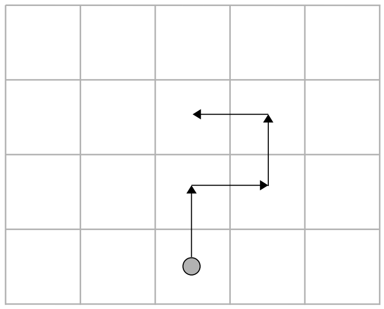

Fig. 3. This illustrates a sample path of the agent moving on a grid. All possible sample paths are generated and evaluated using the reward function.

is given by the following equation.

$$R_{RF}(i,r,c) = \frac{-1}{\sum_{j=1}^{N_r} (\mid r - r_j \mid + \mid c - c_j \mid)^2} \tag{14}$$

The sign of R_{RF} is negative to make this a repulsive force rather than an attractive force. The parameter N_r is the number of agents near enough to be visually seen by agent i. With this form of indirect communication, there is a limit to how far away agents can see each other. In practice this does not matter as agents that are sufficiently far away have very small reward function penalty. If one agent cannot see another agent, the distance can be treated as infinite which results in a penalty of zero.

One contribution of this paper is in presenting a reward function that balances these two competing objectives. This function calculates the total reward by taking a weighted average of the two path selection criteria discussed above. The balance between these objectives can be set by choosing α in the following equation. The selection of this parameter will depend on the specific application.

$$R(i,r,c) = \alpha R_{EM}(i,r,c) + (1-\alpha) R_{RF}(i,r,c) \tag{15}$$

Finally, the path planning algorithm is run discretely, over grid squares, and on a receding horizon. Since this is on a grid, at each step an agent can only move to one of four adjacent grid squares that are in the cardinal directions. All possible future paths N_s steps into the future can be determined using a tree data structure. The reward function is calculated for all 4^{N_s} possible future paths. A sample path is given in Figure 3. The path with the highest reward is the path selected. The agent only commits to taking the first step of this best path. During the next simulation iteration, the path optimization algorithm is run again to decide on a new path. This may be the same decision as the second step of the previously determined best path, but it often changes due to the rapidly changing nature of the estimated state of the fire. This entire algorithm, both estimation and path planning, is summarized in Algorithm 1.

Algorithm Estimation and path planning for agent i

```
for time k = 1 to k_final do
  Propagate belief: P(b_i^(r,c)(k) | b_i^(r,c)(k-1))
  Take measurement: M_i^(r,c)(k)
  Update belief: P(b_i^(r,c)(k+) | M_i^(r,c)(k))
  for neighbor j ∈ N_i do
    Send: M_i^(r,c)(k)
    Receive: M_j(k)
    Update belief: P((b_i^(r_j,c_j)(k+) | M_j(k))
  end for
  if step mod (k, N_BF) == 0 then
    for neighbor: j ∈ N_i do
      Send belief: B_i(k)
      Receive belief: B_j(k)
      Update belief: P(B_i(k+) | B_j(k))
    end for
  end if
  Generate 4^(N_s) paths
  for paths i_p = 1 to 4^(N_s) do
    Set reward: R(i_p) = 0
    for step j_p = 1 to N_s do
      Calculate reward: R(i_p) ← R(i_p) + R(i, r_(j_p), c_(j_p))
    end for
  end for
  Choose path: i_p = max_(i_p) R(i_p)
  Update position: (r_(i_p)(k+1), c_(i_p)(k+1)) ← (r(k), c(k))
end for
```

V. Simulation Results

The estimation and path planning approach detailed in this paper was simulated to test its performance. For comparison, four related approaches were also simulated. The five methods tested will be summarized using the naming convention given on Table I. Algorithm 1 uses the classic lawnmower search strategy (LM) with no communication (NC) between agents. The UAVs start at specific locations and sweep through the grid in a lawnmower pattern. Each agent sweeps an equally portioned section of the grid and repeats the pattern until the simulation ends. Each agent takes measurements and fuses them with its own belief using Bayes rule. Algorithm 2 still relies on individual agents collecting measurements and fusing them using Bayes rule. In addition, the agents use belief consensus (BC) to fuse the maps shared by neighboring agents with their own estimated belief maps on a periodic basis. The belief consensus period was initially set to be the same as the simulation step size. In order to reduce the computational requirements, we tested reduced frequencies of updates every 1, 2, 5, 10, 15, 20, 50, 100, and 200 timesteps. Out of the tests, we saw the best performance (lowest estimation error) by reducing the fusion period to once every 10 simulation timesteps. This is a more realistic rate considering the time involved for both communication and computation. The UAVs shared their beliefs using a graph with a ring structure as shown in Figure 4. This formation was chosen as a relatively low number of connections between agents to test the distributed nature of this approach. When a strongly connected graph is used, the performance is as good

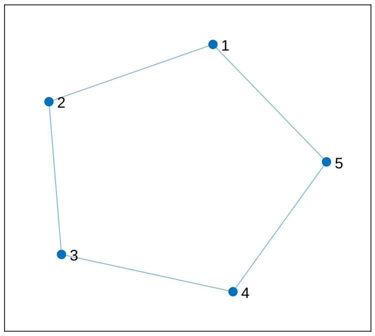

Fig. 4. All simulations were completed with five agents. For Algorithm 1, there was no communication between agents. For Algorithms 2-5 the communication between agents was implemented using a ring formation where agent 1 communicates with agents 2 and 5, etc.

as a centralized algorithm. The path planning algorithm was implemented using the approach of maximizing mutual information, i.e. entropy minimization (EM). Algorithm 3 uses both Bayes rule and belief consensus and adds measurement sharing (MS). The UAVs share their measurements with other agents according to the same ring graph. They continue to apply belief consensus using their neighbors' maps every 10 timesteps. Algorithm 4 uses belief consensus but does not include measurement sharing. It is different from Algorithm 2 in that it introduces a revised reward function with repulsive forces (RF) to incentivize agents to improve coverage as described in Section IV. Algorithm 5 combines the estimation approach of Algorithm 3 with the path planning of Algorithm 4. It uses belief consensus, measurement sharing, entropy minimization, and repulsive forces. This is the complete algorithm as discussed in Sections III and IV. The value of α, from equation (15), was tested in the range $0 < \alpha < 1$. The performance did not vary much, but the best performance came from $\alpha = 0.7$. This value can be seen as weighting the entropy minimization algorithm by 0.7 and the repulsive force algorithm by 0.3.

This simulation involved several parameters that were constant between all five algorithms. The environment was a 20x20 grid. The actual fire spread was randomized but the average spread rate was 2.1 meters per minute. The measurement algorithm was implemented with a false positive rate of 0.1 and a false negative rate of 0.1. There were 5 agents searching the space and communicating in a ring formation, when communicating. For the belief consensus algorithm outlined in Section III, the parameter $\gamma = 0.3$. For the simulations that used repulsive forces, the agents could not directly communicate with all agents but could see other agents up to 5 grid squares away in the L_1-norm (taxicab norm) sense. When path planning was used, the agents planned $N_s = 5$ steps into the future. A single run of the simulation was 1000 timesteps which corresponds to about 83 minutes of real time for the UAVs to collect information.

This simulation framework involved randomness in both the spread of the fire and the measurement errors due to false positives and negatives. In order to reduce the effect of the randomness, the simulation was repeated 1000 times as a Monte Carlo simulation. The performance of a single run of the simulation was determined by calculating the mean squared error (MSE) between the estimated state and the true state for the entire map. For example, the estimated map was initialized so that each square has a 0.5 probability of fire present at that location. The squared error for each grid square square is 0.25. When the error at all 400 grid squares is summed, the total mean squared error is 100. This value provides a comparison point for the simulation results. The MSE is averaged for all 1000 timesteps of each test and averaged across all agents' belief maps. With 1000 Monte Carlo trials, the MSE is averaged across all trials. The resulting average MSE is shown on Table I.

TABLE I
COMPARISON OF ESTIMATION ERROR BETWEEN THE DIFFERENT ESTIMATION AND PATH PLANNING ALGORITHMS.

Estimation and Planning Algorithm	Avg. MSE
Algorithm 1 (NC,LM)	79.51
Algorithm 2 (BC,EM)	43.47
Algorithm 3 (BC+SM,EM)	19.59
Algorithm 4 (BC,EM+RF)	44.29
Algorithm 5 (BC+SM,EM+RF)	18.20

We see a significant decrease in the average MSE from the simple Lawnmower algorithm (Algorithm 1) to Algorithm 2 with both belief consensus and entropy minimization. The error drops further when adding measurement sharing (Algorithm 3). There is a small improvement in the final simulation where the repulsive forces are added. It isn't clear why the MSE increased from Algorithm 2 to 4, with the addition of repulsive forces. This change was small but seems to imply that repulsive forces would not improve the approach using belief consensus and entropy minimization. The only change between Algorithm 4 and 5 is the addition of shared measurements which significantly decreased the error once again. For this estimation and path planning scenario of a forest fire spreading on a grid, it is clear that using belief consensus and shared measurements on a graph significantly reduces the estimation error. The inclusion of the repulsive force seems to have a minor effect to the average MSE. While this reduction is small (7.1%), it does represent a statistically significant decrease. In addition, this decrease persisted while varying design parameters.

VI. CONCLUSIONS

This paper presented a decentralized estimation and path planning algorithm that can be used to reduce estimation error and improve the performance of cooperative search algorithms. The approach has a unique combination of belief fusion methods including Bayesian fusion, measurement sharing and fusion on a graph, and belief consensus. Each agent generates its own estimated state and makes decisions on where to collect future measurements by using a reward-based path planning algorithm. The reward function balances the objectives of collecting the best local information and avoiding other agents to provide better coverage of the environment. This function incentivizes agents to collect better measurements which further reduces estimation error and improves decision-making under uncertainty. One of the main benefits of this approach is that it is easily scalable as each agent is running an identical algorithm. This estimation approach was tested on the scenario of modeling and tracking the spread of a forest fire. The algorithm performed better than existing approaches under the conditions tested. Further testing should be completed to determine if this approach is applicable to a general problem of estimating the state of an environment with uncertain information.

REFERENCES

[1] W. Ren, R. W. Beard, and E. M. Atkins, "A Survey of Consensus Problems in Multi-agent Coordination," in *American Control Conference*, 2005, pp. 1859–1864.

[2] R. M. Murray, "Recent research in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement and Control*, vol. 129, pp. 571–583, 2007.

[3] F. Garin and L. Schenato, "A Survey on Distributed Estimation and Control Applications Using Linear Consensus Algorithms," in *Networked Control Systems*, A. Bemporad, M. Heemels, and M. Johansson, Eds. Springer, 2010, pp. 75–107.

[4] K. K. Oh, M. C. Park, and H. S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[5] F. Rossi, S. Bandyopadhyay, M. Wolf, and M. Pavone, "Review of Multi-Agent Algorithms for Collective Behavior: a Structural Taxonomy," *IFAC-PapersOnLine*, vol. 51, no. 12, pp. 112–117, 2018.

[6] M. M. Gulzar, S. T. H. Rizvi, M. Y. Javed, U. Munir, and H. Asif, "Multi-agent cooperative control consensus: A comparative review," *Electronics (Switzerland)*, vol. 7, no. 22, pp. 1–20, 2018.

[7] M. A. Kovacina, D. Palmer, G. Yang, and R. Vaidyanathan, "Multi-agent control algorithms for chemical cloud detection and mapping using unmanned air vehicles," *IEEE International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2782–2788, 2002.

[8] 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.

[9] C. Dupont, L. Tobias, and B. Luvison, "Crowd-11: A dataset for fine grained crowd behaviour analysis." in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 9–16.

[10] M. Yang, L. Rashidi, S. Rajasegarar, C. Leckie, A. Rao, and M. Palaniswami, "Crowd Activity Change Point Detection in Videos via Graph Stream Mining," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 215–223.

[11] P. DeLima and D. Pack, "Toward developing an optimal cooperative search algorithm for multiple unmanned aerial vehicles," in *IEEE International Symposium on Collaborative Technologies and Systems*, 2008, pp. 506–512.

[12] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[13] R. Olfati-Saber, E. Franco, E. Frazzoli, and J. S. Shamma, "Belief Consensus and Distributed Hypothesis Testing in Sensor Networks," in *Networked Embedded Sensing and Control*, 2006, pp. 169–182.

[14] A. Agarwal, S. Kumar, and K. Sycara, "Learning Transferable Cooperative Behavior in Multi-Agent Teams," *arXiv*, pp. 1–10, 2019. [Online]. Available: http://arxiv.org/abs/1906.01202

[15] N. P. Tani, A. Blatt, D. Quint, and A. Gopinathan, "Optimal Cooperative Searching using Purely Repulsive Interactions," *Journal of Theoretical Biology*, vol. 361, pp. 159–164, 2014.

[16] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous agents and multi-agent systems*, vol. 11, no. 3, pp. 387–434, 2005.

[17] R. C. Rothermel, *A mathematical model for predicting fire spread in wildland fuels*. Intermountain Forest and Range Experiment Station, Forest Service, United States Department of Agriculture, 1972.

[18] M. Drissi, "Modelling the spreading of large-scale wildland fires," *arXiv preprint arXiv:1402.6187*, 2014.

[19] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer, 2001.