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 deploy a team of agents to cooperatively accomplish an objective. While multi-agent systems provide advantages, including lower cost and robustness to failure, there is a need for additional study of principles for the design and test of these decentralized systems. The main contribution of this paper is a novel estimation and path planning algorithm that can be used for improved estimation of uncertain environments. The estimation algorithm utilizes Bayesian fusion, measurement sharing on a graph, and belief consensus. One new component of this approach is the reward-based path planning algorithm that incentivizes agents to collect the best local information as well as improve coverage of the environment. When agents plan paths to collect more valuable information, the estimation error is reduced. This approach is studied for the application of estimating the state of a forest fire but can be applied in many domains. Simulations were performed to demonstrate the effectiveness of the algorithm compared to other approaches.**

I. Introduction

Advances in technology have 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. It is more important than ever for researchers to study the scientific principles of multi-agent systems, cooperative control, and emergent behavior. There are a number of survey papers on existing approaches for these multi-agent systems [1]–[6]. In this paper, we study teams of unmanned aerial vehicles (UAVs) that collect information in an uncertain environment. Similar problems have been studied for detecting clouds of hazardous chemicals [7] or to track the spread of wildfires [8]. In the current paper, we estimate the state of a forest fire using UAVs to collect information that is uncertain and incomplete. We assume that each agent can detect fire by using a camera and a detection algorithm. We model errors in the system using false positives and negatives. Other work has demonstrated the efficacy of a decentralized UAV system versus a single aircraft for surveillance operations [9].

Cooperative control of multi-agent systems is a broad field with many promising approaches. One body of work uses the consensus algorithm for distributed rendezvous, flocking, and synchronization (see, e.g. [1], [3], [10]). A related approach uses belief consensus to reach agreement by fusing probabilistic estimations between agents [11]. Other research uses a reward function for incentivizing agents to make decisions towards cooperative team goals [12]. For improving coverage of the environment, one bio-inspired approach uses repulsive interactions between agents using *indirect communication* rather than direct communication [13]. A similar approach using indirect communication was presented in [14].

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 and path planning algorithms which can improve the quality of future measurements. The proposed method is compared to existing algorithms using simulations to determine the efficacy for the stated problem. Each agent can only sense locally and make decisions based on information available to that agent. We use belief consensus to fuse individual estimates of the state of the world. As this communication between agents is on a graph, we also use the graph to share measurements between agents. While each agent can only measure at its current location, sharing measurements allows for agents to better estimate the full environment. For path planning, each agent uses its own estimated map to decide where to move locally to maximize information gathered. This is a discrete version of moving along the gradient of mutual information [8]. This is incorporated into a reward function where agents are also incentivized to maximize coverage. This is done using visual sensing of other agents and repulsive forces between agents.

The remainder of this paper is organized as follows. In Section II, we introduce the problem including the model of the forest fire and the capabilities of the UAV team. Section III introduces the estimation approach including Bayesian fusion of measurements, sharing of measurements, and the belief consensus algorithm. Section IV introduces the path planning algorithm using a reward function. 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 can be applied to modeling any uncertain environment, we were inspired by recent events to apply this work to modeling the state of a forest fire. With the increase in large fires throughout the world, there is a need for new advances in measuring and estimating forest fires. It is challenging to measure a large fire with a single UAV due to the size and spread rate. We propose an alternative approach to employ a team of UAVs that can communicate and cooperate to more quickly

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 .

and accurately estimate the state of a fire. With a more accurate estimate, resources can be allocated more effectively to contain the fire. Improved estimates have additional benefits such as providing information for protecting firefighters and more effective evacuation. 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 forest fires, we studied existing models that estimate the spread rate of fires based on a large number of parameters [15], [16]. 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 wind conditions. Based on these detailed models, it can be determined that the majority of large forest fires spread at a rate of 0.5-2.5 meters/min. To account for the variability of fire spread rate due to 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 representation that is G_s squares by G_s squares. We index each grid square using its row and column location (r, c). In our scenario, each grid square is 25 meters by 25 meters. The simulation is discretized physically and temporally with a fixed timestep of 5 seconds for the simulated world. This is a sufficiently high rate for the spread of the fire that is measured in minutes.

Initially, one grid square is uniformly randomly chosen to be the start of the fire. The fire may spread to any adjacent grid square with a probability dependent on the fire spread rate. We chose a spread rate of 2.1 meters/min which is near the upper bound given in previous work. The spread probabilities can be determined by working backwards from the rate to get 0.007 grid squares per simulation step. This equates to a probability of 0.007 that the fire spreads to each adjacent square at each timestep. A simulated test case of this fire spread model is shown in Figure 1.

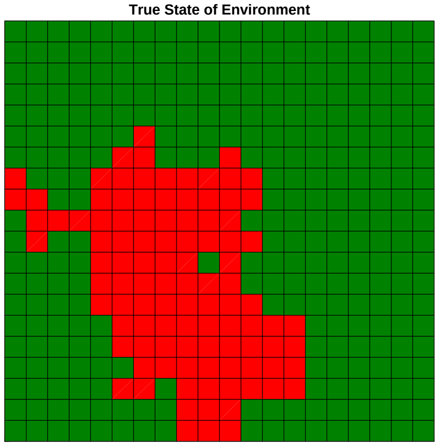

Fig. 1. The true state of each grid square is a binary variable representing the presence of fire (1, red) or the absence of fire (0, green).

B. Modeling the UAV Team

As the fire spreads, a team of UAVs is deployed to take measurements. In existing fire tracking and modeling, this is done centrally where 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 agents take measurements, communicate, and make decisions based solely on the information that is available to that agent. UAVs measure the current state of a fire with cameras and image detection algorithms which are not perfect and false measurements are produced. To simulate 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 and false negative rates may be different from each other. Finally, based on the simulation discretization, the UAVs can move at a rate of one grid square per timestep which is 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 agent, the belief of the state of the forest fire is represented as a matrix of probabilities. The belief of agent i is given by $B_i = \{b_i^{(r,c)}\}$. Each element $b_i^{(r,c)}$ represents the probability that the location (r, c) contains fire. The matrix is initialized uniformly with probability 0.5 that each location contains fire which represents maximal uncertainty in the state. As the UAVs proceed through the environment, they take measurements which will increase or decrease the likelihood that the location contains fire which will change the uncertainty of the estimate. The measurement model does include false positives and negatives with rates based on the accuracy of the underlying image detection algorithms.

When an agent visits a location, it takes a measurement of the 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 binary (either fire or no fire present), the estimate of the fire state is a continuously valued probability, $0 \leq b_i^{(r,c)}(k) \leq 1$. When a binary measurement $M_i^{(r,c)}(k)$ is taken, the updated state of the fire at this location $b_i^{(r,c)}(k^+)$ can be calculated by using Bayes' rule.

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

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

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

represents the measurement model which is developed offline using the false measurement rates.

Between measurements, the estimated fire state at each location is updated using a prediction model. The posterior estimate is determined from the prior likelihood and the estimates of the adjacent grid squares which can be denoted G_a.

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

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

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

In the absence of measurements, the likelihood that a location is on fire increases when adjacent locations have a high probability of being on fire.

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. This subsection introduces some graph theory and explains 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.

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

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 [17].

At each time step, each agent i measures the state of the forest fire at its physical location. This measurement is communicated to 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. This is done according to Bayes' rule as discussed previously. For a graph with N agents, each agent may update their own belief matrix with 1 to N measurements, depending on the graph.

C. Belief Consensus

Despite agents sharing measurements with their neighbors, their beliefs will diverge due to differences in available measurements. To reduce this divergence, another component of this algorithm is estimation fusion between agents using belief consensus. In belief consensus, agents share their belief of the full state of the environment with neighboring agents. Agent i shares its current estimate B_i 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 as in [11]. This process is repeated for all agents $j \in N_i$. This fusion is applied to update the belief of

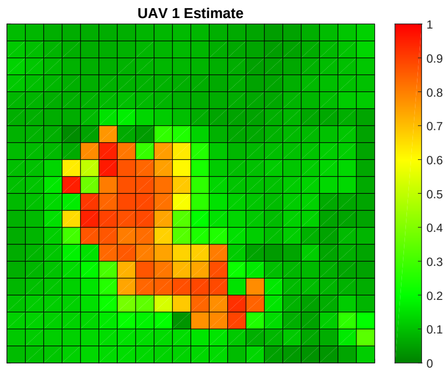

Fig. 2. This figure shows one agent's estimated state of the fire. Red indicates a high probability of a fire while dark green indicates a low probability of fire. The highest uncertainty squares are yellow to yellow-green.

agent i before the fusion (k) to immediately after (k^+) using a weighted geometric mean.

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

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 [11] for proof.

The belief consensus approach can greatly reduce the difference in beliefs between agents, resulting in reduced estimation error. 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 the available computational resources. Selecting this rate is discussed again in Section V. The belief map for one agent during one simulation run 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 estimation accuracy.

IV. Path Planning Algorithm

In this decentralized information gathering scenario, each UAV makes local decisions based on the information available from measurements, communication with other agents, and the estimation algorithm. The approach uses a reward function that incentivizes agents to make decisions that balance the goals of collecting the best possible measurements and ensuring better coverage of the environment. A well-chosen reward function will reduce the estimation error by increasing the value of future measurements. In our problem scenario, each agent iteratively collects measurements and updates its estimate of the state of the fire according to the algorithm outlined in the previous section. The agent then ranks possible future paths according to the reward function. The path with the highest reward is chosen. The reward function can be explained as a weighted average of two different path selection criteria.

The first path selection criterion is based on choosing paths that have the highest value in the information theoretic sense. This approach calculates the value of a future measurement by using the current estimate of the state of the fire at that location. Agents should plan paths that visit locations with high uncertainty while avoiding locations with lower uncertainty. 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*.

$$H_b(p) = -plog_2(p) - (1-p)log_2(1-p) \tag{12}$$

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).

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

This reward function incentivizes UAVs to visit locations with the highest uncertainty as they move through the environment. When the simulation begins, all grid squares are initialized with probability 0.5 which represents maximal uncertainty. When the estimated state is more certain, e.g. $b_i^{(r,c)}$ near zero or one, the value of measuring that location is low. This approach provides some level of cooperation. When one agent measures a physical location, it reduces the reward for a neighboring agent so agents visit distinct locations.

The second path selection criterion encourages UAVs to maximize coverage of the environment to collect measurements of different physical locations. It is assumed that agents can locally sense other agents and use this as a form of indirect communication. A reward function is introduced to penalize agents for being near other agents. This function takes the form of a repulsive force that is inversely proportional to the square of the distance between agents. Agents attempt to maximize this reward function by keeping a larger distance between them and other agents. This distance is calculated as a distance expressed in grid squares. The reward for agent i to visit and measure a location (r, c) 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.

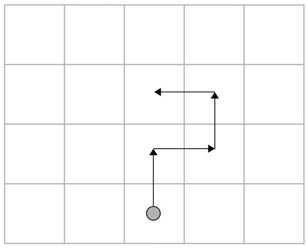

Fig. 3. This illustrates a sample path of the agent moving on a grid. All possible sample paths are generated and evaluated using the reward function.

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.

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

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 the algorithm below.

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: $P(b_i^{(r,c)}(k^+) \mid M_i^{(r,c)}(k))$
  for neighbor $j \in N_i$ **do**
    Send: $M_i^{(r,c)}(k)$
    Receive: $M_j(k)$
    Update belief: $P((b_i^{(r_j,c_j)}(k^+) \mid M_j(k))$
  end for
  if step $\mod(k, N_{BF}) == 0$ **then**
    for neighbor: $j \in N_i$ **do**
      Send belief: $B_i(k)$
      Receive belief: $B_j(k)$
      Update belief: $P(B_i(k^+) \mid B_j(k))$
    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: $R(i_p) \leftarrow R(i_p) + R(i, r_{j_p}, c_{j_p})$
    end for
  end for
  Choose path: $i_p = \max_{i_p} R(i_p)$
  Update position: $(r_{i_p}(k+1), c_{i_p}(k+1)) \leftarrow (r(k), c(k))$
end for

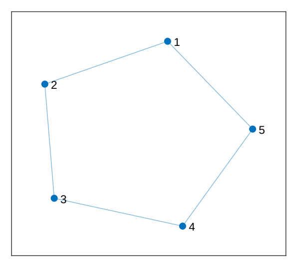

Fig. 4. All simulations were completed with five agents. For Algorithm 1, there was no communication between agents. For Algorithms 2-5 the communication between agents was implemented using a ring formation where agent 1 communicates with agents 2 and 5, etc.

V. Simulation Results

The estimation and path planning algorithm was simulated and compared to four related approaches. The five methods tested are 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. The space is divided evenly between the five agents. Each agent sweeps an equally portioned section of the environment and repeats until the simulation ends. Agents take measurements and update their own beliefs using Bayes' rule. Algorithm 2 uses Bayes' rule for measurement fusion, but adds belief consensus (BC) to fuse estimates from neighboring agents on a periodic basis. Several time periods were tested for BC and the lowest estimation error was found when BC is applied once every 10 simulation timesteps. The UAVs shared their beliefs using a graph with a ring structure as shown in Figure 4. The path planning algorithm used information maximization, i.e. entropy minimization (EM). Algorithm 3 uses Bayes' rule and BC and adds measurement sharing (MS) on the same ring graph. They continue to apply BC using their neighbors' maps every 10 timesteps. Algorithm 4 uses BC but does not include MS. It is different from Algorithm 2 in that it uses the 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 BC, MS, EM, and RF which is the complete algorithm covered in Sections III and IV. The value of α, from equation (15), was tested and the best performance was $\alpha = 0.7$.

Many parameters were held constant between all cases tested. The environment was a 20x20 grid. The fire spread was randomized with an average spread rate of 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. For the BC algorithm outlined in Section III, the parameter $\gamma = 0.3$. For the simulations that used RF, 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.

This simulation involved randomness in the spread of the fire and the measurement errors. In order to reduce the effect of randomness, each test case was repeated 1000 times as a Monte Carlo simulation. The performance of a simulation run was determined by summing the mean squared error (MSE) between the estimated state and the true state for the entire map. For comparison, the total MSE is 100 before any measurements have been taken. The MSE is averaged for all 1000 timesteps and averaged across all agents' belief maps. With 1000 Monte Carlo trials, the MSE is averaged across all trials and 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+MS,EM)	19.59
Algorithm 4 (BC,EM+RF)	44.29
Algorithm 5 (BC+MS,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 BC and EM. The error drops further when adding MS (Algorithm 3). There is a small improvement in the final simulation where RF are added. It isn't clear why the MSE increased from Algorithm 2 to 4, with the addition of RF. This change is small but implies that RF do not improve the approach using BC and EM. The only change between Algorithm 4 and 5 is the addition of MS 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 BC and MS on a graph significantly reduces the estimation error. The inclusion of the RF seems to have a small effect to the average MSE. While this reduction is small (7.1%), it does represent a statistically significant decrease.

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 estimation algorithms. The approach has a novel combination of estimation fusion methods including Bayesian fusion, measurement sharing on a graph, and belief consensus. Each agent generates its own estimated state and decides 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. One of the benefits of this approach is that it is 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 some 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] 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.

[10] 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.

[11] 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.

[12] 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

[13] 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.

[14] 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.

[15] 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.

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

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