IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Spatial Scheduling of Informative Meetings for Multi-Agent Persistent Coverage

Ravi N. Haksar[1], Sebastian Trimpe[2], and Mac Schwager[3]

Abstract— **In this work, we develop a novel decentralized coordination algorithm for a team of autonomous unmanned aerial vehicles (UAVs) to surveil an aggressive forest wildfire. For dangerous environmental processes that occur over very large areas, like forest wildfires, multi-agent systems cannot rely on long-range communication networks. Therefore, our framework is formulated for very restrictive communication constraints: UAVs are only able to communicate when they are physically close to each other. To accommodate this constraint, the UAVs schedule a time and place to meet in the future to guarantee that they will be able to meet up again and share their belief of the wildfire state. In contrast with prior work, we allow for a discrete time, discrete space Markov model with a large state space as well as restrictive communication constraints. We demonstrate the effectiveness of our approach using simulations of a wildfire model that has 10^{298} total states.**

I. Introduction

We consider the task of using a cooperative team of autonomous aerial vehicles (UAVs) to continuously surveil an aggressive forest wildfire. We call a wildfire "aggressive" when its rate of spread is comparable to the movement speed of the UAVs. In such cases, the agents cannot rely on neglecting the travel time to points of interest in the environment. There is significant interest in using cooperative multi-agent teams to mitigate natural disasters. In California, 10 of the 20 most destructive wildfires in state history occurred within the last four years, including the current most destructive wildfire in November 2018 [1]. Furthermore, the number of wildfires and their intensity will increase in the United States [2]. For natural disaster response, multi-agent systems also cannot rely on communication networks that operate over unlimited range [3]. Issues such as high transmitter power requirements, limited on-board power, and weather conditions typically preclude robust, long-range communication networks. Therefore, we assume that agents communicate only when they are physically close. We model this constraint as requiring agents to be at the same location in the forest in order to exchange information. Furthermore, since wildfires naturally occur over large areas, agents must communicate to maintain an accurate belief of the wildfire as traversing the entire domain individually is ineffective.

In prior work [4], [5], [6], we developed control and estimation methods for wildfires modeled by a class of large discrete graph-based Markov models, including a decentralized multi-agent framework to suppress a wildfire using perfect state information. In this work, we use a realistic sensing model and build a scalable coordination algorithm to enable the use of our control framework with state uncertainty.

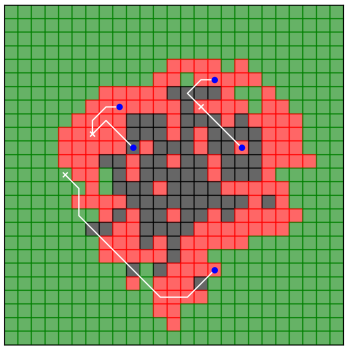
Fig. 1: The forest lattice is visualized as a grid of cells. Multiple agents (blue circles) are tasked with monitoring an aggressive wildfire (red are fires, black are burnt, and green are healthy trees). Agents schedule meetings (white X's) to periodically communicate and coordinate their efforts. Communication only occurs in meetings.

Many approaches have been proposed in literature for the persistent coverage problem and some directly consider communication constraints. However, the majority of these methods still allow for communication over arbitrary distances. In addition, prior work also does not scale to processes modeled by large discrete Markov models. Therefore, we propose a novel decentralized coordination algorithm that is scalable to large process models and large multi-agent teams. Agents coordinate by scheduling a future time and location in the forest to meet to ensure information is continuously shared. By using this approach, agents avoid duplicating efforts when they are unable to communicate. Our framework approximately solves the problem of multi-agent exploration with limited communication, and we show through simulations that our approach is effective.

The main contributions of this work are: (1) we propose a novel scheduling scheme which is executed in a decentralized manner; (2) we create a decentralized information-based planning algorithm based on the scheduling scheme which leads to effective coordination; and (3) we demonstrate the effectiveness of our approach using simulation experiments.

The remainder of this work is organized as follows. Section II reviews relevant prior work. We introduce a lattice-based Markov model of a forest wildfire and define an agent model in Section III. An overview of our framework is provided in Section IV. Details for the novel scheduling

*This research was supported by NSF grant IIS-1646921, DARPA YFA award D18AP00064, ONR grant N00014-18-1-2830, the Cyber Valley Initiative, and the Max Planck Society. We are grateful for this support.

[1]Department of Mechanical Engineering, Stanford University, Stanford, USA rhaksar@stanford.edu

[2]Max Planck Institute for Intelligent Systems, Intelligent Control Systems Group, Stuttgart, Germany trimpe@is.mpg.de

[3]Department of Aeronautics & Astronautics, Stanford University, Stanford, USA schwager@stanford.edu

scheme and the path planning methods are provided in Sections V and VI. We demonstrate our approach with simulations in Section VII and provide concluding remarks in Section VIII.

II. Prior Work

Many methods have been proposed for persistent coverage. We review the approaches most relevant to our work here.

Coverage Control. Coverage frameworks have been developed to account for communication limitations [7], [8], but are typically applied to static or slowly changing environmental processes whereas we aim to monitor a fast process. In addition, many methods require a connected communication graph in order to prove stability properties. In contrast, we consider a problem setup where communication connectivity cannot be guaranteed.

Informative Planning. Information-based metrics have been used to create multi-agent exploration frameworks, such as sequential allocation [9], [10], sampling [11], [12], [13], and mixed-integer optimization [14]. Similar to coverage control, many of these methods assume a connected or fully-connected communication graph. While some methods include communication limitations, such as link failures [12] or lossy channels [13], these frameworks still assume communication can occur over arbitrarily large distances.

Wildfire Surveillance. There is a significant amount of literature on surveillance specifically for wildfires, some of which propose centralized frameworks [15], [16], [17]. Decentralized methods are based on wildfire boundaries or perimeters [18], [19], image-based feature processing [20], [21], potential field controllers [22], [23], and team coordination or assignment [24], [25]. Almost all of these methods assume a large or unlimited communication radius, e.g. Ure et al. [21] only consider bandwidth limitations, whereas we enforce limited range communication.

Teamwork Strategies. Our framework can be viewed as a synchronization (or rendezvous) strategy [26], [27], [28], but we do not assume periodic synchronization of all agents (which may occur over large distances). In addition, some methods [28] require precise boundary information that cannot be known in advance, but must be estimated as the wildfire spreads, and is non-trivial for multiple wildfire sources. In contrast, our approach handles multiple sources which may merge or split over time.

Prior work also does not address large discrete space and discrete time models, as we do in this work. Many methods use discrete process filters or enumerate (or sample) paths through the domain, and do not scale to large state spaces. Likewise, the multi-agent exploration problem is naturally described by partially observable Markov decision processes [29], but exact and many approximate methods are intractable for our problem. In the next section, we introduce the wildfire model along with an agent model.

III. Modeling

A. Process Model

The forest is modeled as a finite 2D lattice of dimensions $d_1 \times d_2$ with $d_1 d_2$ total nodes [30], [4]; see Fig. 1. The position of node $i \in \{1, \dots, d_1 d_2\}$ on the lattice is given by $q_i \in \mathcal{Q} = \mathbb{Z}^2 \cap \{[1, d_1] \times [1, d_2]\}$. The state of lattice node i at time t is represented by x_i^t and the set $\mathcal{N}(i)$ refers to the lattice neighbors of node i. Each lattice node corresponds to a tree with dynamics represented by a discrete space and discrete time Markov model. A tree has three possible states,

TABLE I: Tree transition probabilities for wildfire model. Blank entries are zero.

		x_i^{t+1}		
		H	F	B
x_i^t	H	$(1-\alpha_i)^{f_i^t}$	$1-(1-\alpha_i)^{f_i^t}$	
	F		β_i	$1-\beta_i$
	B			1

$$x_i^t \in \{\text{Healthy}, \text{On Fire}, \text{Burnt}\} = \{H, F, B\}.$$

A healthy tree remains healthy unless at least one neighboring tree is on fire, in which case the parameter $0 \leq \alpha_i \leq 1$ determines the likelihood that the tree will be on fire at the next time step. This transition is based on the number of neighboring trees on fire, $f_i^t = \sum_{j \in \mathcal{N}(i)} \mathbf{1}(x_j^t = F)$, where the notation $\mathbf{1}(\cdot)$ represents the indicator function. A tree on fire remains on fire with likelihood determined by the parameter $0 \leq \beta_i \leq 1$. Lastly, a burnt tree remains burnt for all time. Table I summarizes the dynamics of trees.

B. Agent Model

Dynamics. Each agent moves on the same lattice as the forest. The position of an agent k is represented by $z_k^t \in \mathbb{Z}^2$. The agent action space contains nine possible actions which modify the position, $z_k^{t+1} = z_k^t + u_k^t$ with $u_k^t \in \mathcal{U}$ and,

$$\mathcal{U} = \left\{ \begin{bmatrix} i \\ j \end{bmatrix} \middle| (i,j) \in \{\{-1,0,1\} \times \{-1,0,1\}\} \right\}.$$

Sensors. Each agent has a downward facing camera which produces a noisy estimate of the states of an $h \times w$ sized grid of trees. The agent is located at the center of the image.

Definition 1 (Camera Function $I(q)$). The function $I(q)$ returns a set of size hw containing the trees $i \in \{1, \dots, d_1 d_2\}$ observed by a camera centered at location q.

The accuracy of each tree state in the image is parameterized by $0 \leq p_c \leq 1$, which is the probability that the observation is the ground truth tree state x_i^t. Each tree in the image is observed independently of all other trees. Equation (1) summarizes the sensor model,

$$p(y_i^t \mid x_i^t) = \begin{cases} \frac{1}{2}(1-p_c) + \mathbf{1}(y_i^t = x_i^t)\left(\frac{3}{2}p_c - \frac{1}{2}\right) & \text{if } i \in I(z_k^t), \\ \frac{1}{3} & \text{otherwise.} \end{cases} \quad (1)$$

In this model, observations of trees in the agent's camera ($i \in I(z_k^t)$) have probability p_c of matching the ground truth ($y_i^t = x_i^t$) and have probability $\frac{1}{2}(1-p_c)$ of not matching ($y_i^t \neq x_i^t$). Observations of trees outside the camera view have uniform uncertainty (probability $\frac{1}{3}$) for each of the three tree states.

Wildfire Belief. Agents maintain their own belief over the state of the wildfire that is updated every time step using their observations with an approximate Bayesian filter. Agents merge beliefs when in a meeting to improve coordination and wildfire tracking. Details on the belief update and merge are provided in Section IV-A.

Communication. Agents only exchange information at prearranged meetings and information is shared by using a Bayesian information fusion strategy so that all agents in the meeting have the same belief. Agents then use their belief to schedule their next meeting and compute nominal paths to the next meeting.

Schedule Information. Agents maintain information related to the meetings they participate in, described by schedules $\mathcal{S}$ and $\mathcal{S}'$. Details on these quantities are provided in Section V.

i. Time budget, d_k. The amount of time remaining until the next meeting F_k will occur.
ii. Next meeting location, F_k. A lattice location that the agent must be at in d_k time steps to satisfy the schedule.
iii. Last meeting location, L_k. A lattice location that represents the meeting after the meeting at location F_k.
iv. Stored paths, $\mathcal{R}_k$. A set representing the nominal paths of other agents, which is used to improve the individual path planning of an agent between meetings, when no communication occurs.

The agent modeling assumptions can be reasonably implemented for a sub-class of UAVs. For the discrete motion model, a 3D trajectory and altitude controller [31] can be used to have quadrotors maintain a constant altitude and move laterally. Agents also use a positioning system (e.g. GPS) to maintain a common coordinate frame.

IV. Decentralized Information Gathering Framework

The main idea behind our algorithm is to have pairs of agents schedule their next meeting during their current meeting, subject to all previously scheduled meetings. The agents achieve this by jointly solving a path optimization problem in which they are constrained to end their paths after a prescribed time at the same location (i.e. the next meeting). The last meeting for each agent appear as constraints in this optimization. The objective is to jointly maximize an information gathering metric subject to these constraints. This algorithm guarantees that the future meetings are always kept, even though they are planned in a pairwise distributed and asynchronous fashion. In other words, pairs of agents plan meetings for themselves without knowledge of what all other agents are doing. Between scheduled meetings, an agent can change its path based on new sensed information, as long as the first and last meeting constraints are still satisfied. Our framework results in a distributed, reactive information gathering system.

We only consider pairwise agent meetings in this work, with the aim of maximizing the team's tracking ability of the wildfire. Agents must move to a common location to share beliefs, which leads to many observations of the same area. At the same time, the wildfire is spreading in all directions and therefore the agents sacrifice tracking the wildfire in order to improve their collective belief. In the case of aggressive wildfires, this trade-off may significantly impact the team's performance. We plan to investigate different meeting sizes, along with dynamic scheduling ideas, in future work.

Algorithm 1 Decentralized Information Gathering Framework

```
1:  Schedule initial meetings (Algorithm 2)
2:  Deploy agents
3:  for each time step $t$ do
4:      if a meeting $s \in \mathcal{S} \cup \mathcal{S}'$ occurs then
5:          Merge agents' $k \in s$ beliefs using Eq. (3)
6:          Schedule next meeting for $s$ (Algorithm 3)
7:          Perform team path planning (Algorithm 5)
8:      for each agent $k$ do
9:          Plan path to next meeting $F_k$ (Algorithm 6)
10:         Move to first location of planned path
11:         Reduce time budget, $d_k \leftarrow d_k - 1$
12:         Take image and update individual belief
13:     Update wildfire process every $\rho$ time steps
```

Algorithm 1 provides an overview of the decentralized framework. When agents are meeting, they fuse their individual beliefs, schedule a future meeting, and perform joint path planning to generate nominal paths. Note that these steps can be performed individually (decentralized) or by having one agent perform all steps (centralized) and sharing the results, since each agent has the exact same set of information after fusing beliefs. Outside of meetings, agents do not communicate and individually re-plan their nominal paths based on their own belief of the wildfire. Each agent also updates their own belief at each time step using observations from their camera.

A. Process Filter

We now describe a scalable approximate Bayesian filter that each agent uses to produce a belief over the state of the wildfire at each time step. We note that the tree dynamics need to be modified to account for the relative speed of the agents moving and the wildfire spreading. We model this relative speed by updating the wildfire every ρ time steps; between updates, the wildfire does not change (Alg. 1, line 13). The dynamics of a tree are then,

$$p_i(x_i^{t+1} \mid x_i^t, f_i^t) = \begin{cases} \text{Table I} & \text{every } \rho \text{ time steps,} \\ \mathbf{1}(x_i^{t+1} = x_i^t) & \text{otherwise.} \end{cases} \tag{2}$$

Let $b(x^t) = p(x^t \mid \{y^1, \dots, y^t\})$ represent the belief over the states of all trees given a history of measurements $\{y^1, \dots, y^t\}$; here, y^t represents a measurement of all trees at time t. Given the dynamics (2) and sensor model (1), the exact recursive Bayesian filter [5] to update the belief is,

$$b(x^t) \propto \prod_{i=1}^{d_1 d_2} p_i(y_i^t \mid x_i^t) \sum_{x^{t-1}} b(x^{t-1}) \prod_{i=1}^{d_1 d_2} p_i(x_i^t \mid x_i^{t-1}, f_i^t),$$

which is initialized by the prior $b(x^1)$. However, this filter requires the marginalization of all trees which involves enumerating $3^{d_1 d_2}$ values. This is not tractable for large forest sizes and thus we use an approximate filter instead. The belief is a product of individual beliefs for each tree, $b(x^t) \approx \prod_{i=1}^{d_1 d_2} b(x_i^t)$, where $b(x_i^t) = p(x_i^t \mid \{y_i^1, \ldots, y_i^t\})$. The belief update for each tree is,

$$\begin{aligned} b(x_i^t) \propto\, & p_i(y_i^t \mid x_i^t) \\ & \sum_{x_i^{t-1}} \sum_{x_{\mathcal{N}(i)}^{t-1}} p_i(x_i^t \mid x_i^{t-1}, f_i^{t-1}) b(x_i^{t-1}) \prod_{x_{\mathcal{N}(i)}^{t-1}} b(x_j^{t-1}). \end{aligned}$$

The cost of updating the forest belief per agent is now $\sum_{i=1}^{d_1 d_2} 3^{|\mathcal{N}(i)|+1}$ operations, and storing the belief requires $\mathcal{O}(d_1 d_2)$ space. The approximate filter can be interpreted as a first order approximation to the true Bayesian filter [5].

B. Merging Beliefs

We adopt a straightforward approach for fusing beliefs of multiple agents. Let $b_k(x_i^t)$ represent the belief of agent k about tree i at time t. For agents in a meeting s, a merged belief for a single tree is the average of individual beliefs,

$$\bar{b}(x_i^t) = \frac{1}{|s|} \sum_{k \in s} b_k(x_i^t). \tag{3}$$

The merged belief for all trees is then the application of (3) for each tree to produce $\{\bar{b}(x_i^t)\ \forall i\}$. Each agent in the meeting then replaces its own belief with the merged belief. We use averaging to avoid assigning high confidence to poor beliefs, since agents observe a small fraction of the forest at each time step and must rely on open-loop predictions of un-observed areas for planning. We stress that belief fusion has been studied extensively [32], and is not the focus of our contribution in this work. Our focus is on the distributed, online scheduling of spatial meetings and informative planning between meetings, and any particular belief fusion algorithm can be used within this scheduling framework. We plan to investigate the use of other belief fusion methods in future work. In the next section, we describe our novel scheduling framework.

V. Schedule Framework

We now propose a strategy to create a feasible schedule that can be executed in a decentralized manner. A meeting consists of a location on the lattice and a time interval to describe when the meeting should occur.

A. Meeting Intervals

We specify the timing of meetings with the parameter τ. Every τ time steps, the schedule alternates between two sets of assigned meetings $\mathcal{S}$ and $\mathcal{S}'$,

$$\mathcal{S} = \left\{ \{1,2\}, \ldots, \{2i-1, 2i\} \mid i \in \left\{ 1, \ldots, \left\lfloor \frac{C}{2} \right\rfloor \right\} \right\}$$

$$\mathcal{S}' = \left\{ \{2,3\}, \ldots, \{2i, 2i+1\} \mid i \in \left\{ 1, \ldots, \left\lfloor \frac{C-1}{2} \right\rfloor \right\} \right\}$$

where set $\{i,j\}$ indicates agents i and j will meet and C is the total number of agents. This construction specifies which agents will meet. In addition, agents 1 and C each only have one meeting in $\mathcal{S} \cup \mathcal{S}'$, while all other agents each have two. Fig. 2 shows an example schedule for $C = 5$.

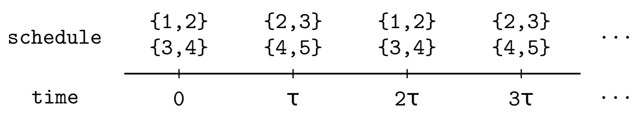

Fig. 2: For $C = 5$ agents, the schedule is $\mathcal{S} = \{\{1,2\},\{3,4\}\}$ and $\mathcal{S}' = \{\{2,3\},\{4,5\}\}$. The set $\{i,j\}$ indicates agents i and j will meet at the same lattice location and meetings between the same pair of agents re-occur every 2τ time steps.

B. Meeting Locations

Given the interval τ, agents must schedule a future meeting location which is reachable, accounting for other meetings that occur in between as well. This is achieved by finding the set of forest locations that can be reached in τ time steps. An information metric is computed for all reachable locations and a "high value" location is chosen.

Initial Meetings. Prior to the deployment of agents, Algorithm 2 is used in a centralized manner to assign meeting locations for all meetings in $\mathcal{S}'$. This procedure sets L_k as long as agent k has a meeting in $\mathcal{S}'$ which ensures that schedule $\mathcal{S}'$ will be satisfied in τ time steps. Reachable meeting locations (Alg. 2, line 4) are tree locations that are at most τ distance from any agent in the meeting to ensure that all agents will be able to reach the location. The chosen meeting location (Alg. 2, line 5) is based on the residual information λ_i at each location, which accounts for the previously assigned locations $\mathcal{B}$. Details on this metric are provided in Section VI.

Next, agents are deployed at locations in the forest according to the schedule $\mathcal{S}$ and individually update their schedule information F_k and L_k. For example, for meeting $\{1,2\}$, agents 1 and 2 are deployed at the same location in the forest and both agents set F_k equal to their deployment location and set $L_k = \emptyset$. Agents with no meeting in $\mathcal{S}$ are deployed by themselves and set $F_k = L_k = \emptyset$. This construction ensures that agents will meet according to schedule $\mathcal{S}$ at the first time step. Furthermore, agents are deployed such that any pair of agents with a meeting $k \in \mathcal{S}'$ are initially within τ distance of each other.

After Algorithm 2, agents that have a meeting in $\mathcal{S}'$ and do not have a meeting in $\mathcal{S}$ will have $F_k = \emptyset$. These agents set $F_k \leftarrow L_k$ and $d_k = \tau$ to have a reachable next meeting. Agents that have a meeting in $\mathcal{S}$ and do not have a meeting in $\mathcal{S}'$ will have $L_k = \emptyset$. These agents set $L_k \leftarrow F_k$ and $d_k = 2\tau$ to have the correct time budget for the next meeting.

Subsequent Meetings. For meetings that occur after the initial time step, agents use Algorithm 3 as a team to schedule future meetings after merging their belief. First, a predicted belief is created by predicting forward τ time steps without any measurements; this corresponds to an open-loop prediction of the wildfire. Next, each agent adds their first and last meeting locations, as well as their stored nominal

Algorithm 2 Schedule Initial Meetings

```
1: Input: schedule $\mathcal{S}'$, meeting interval $\tau$,
          agent initial positions $z_k^1$
2: Initialize observed locations, $\mathcal{B} \leftarrow \emptyset$
3: for each meeting $s \in \mathcal{S}'$ do
4:     Find reachable meeting locations $\mathcal{M}_1$,
       $\mathcal{M}_1 = \{ i \in \{1, \dots, d_1 d_2\} \mid \tau = \max_{k \in s} \|q_i - z_k^1\|_\infty \}.$
5:     Choose location $q_m$ where $m = \arg \max_{i \in \mathcal{M}_1} \lambda_i$ (4)
6:     Set $L_k = q_m$ and $d_k = \tau \ \forall k \in s$
7:     Update observed locations, $\mathcal{B} \leftarrow \mathcal{B} \cup I(q_m)$
```

Algorithm 3 Schedule Next Meeting

```
1: Input: current meeting $s$, merged belief,
          meeting interval $\tau$, agent data $F_k, L_k$
2: Predict future belief $\{ b(x_i^{t+\tau}) \ \forall i \}$
3: Initialize observed locations, $\mathcal{B} \leftarrow \bigcup_{k \in s} I(F_k) \cup I(L_k)$
4: for each agent's set of stored paths $\mathcal{R}_k$ do
5:     for each path $\mathcal{P}_j \in \mathcal{R}_k$ do $\mathcal{B} \leftarrow \mathcal{B} \cup \{ I(q) \mid q \in \mathcal{P}_j \}$
6: Find reachable meeting locations $\mathcal{M}_\tau$,
   $\mathcal{M}_\tau = \{ i \in \{1, \dots, d_1 d_2\} \mid \tau = \max_{k \in s, k \notin \{1, C\}} \|q_i - L_k\|_\infty \}.$
7: for each location $q_i, i \in \mathcal{M}_\tau$ do
8:     for each agent $k \in s$ do
9:         if $j \in \{1, C\}$ then use $d_k = 2\tau$ else use $d_k = \tau$
10:        $\mathcal{P}, w_{ik} \leftarrow$ Search$(L_k, q_i, d_k, \{ b(x_i^{t+\tau}) \ \forall i \}, \mathcal{B})$
11:    $v_i = \frac{1}{|s|} \sum_{k \in s} w_{ik}$
12: Choose meeting location $q_m$ with $m$ chosen randomly
    from the set $\{ i \mid v_i \geq \gamma \max_{j \in \mathcal{M}_\tau} v_j \}$
13: for each agent $k \in s$ do
14:     if $j \in \{1, C\}$ then Set $F_k = L_k = q_m$ and $d_k = 2\tau$
15:     else Set $F_k \leftarrow L_k$ then $L_k \leftarrow q_m$ and $d_k = \tau$
```

paths, to the set of observed locations $\mathcal{B}$ (Alg. 3, lines 3 and 4), to account for locations that will be observed by other agents not participating in the agents' meeting.

Reachable meeting locations (Alg. 3, line 6) are lattice locations that are τ time steps away from the agents' last meetings L_k to ensure that agents are also able to satisfy their other meetings. Note that agents with only one meeting in $\mathcal{S} \cup \mathcal{S}'$ (i.e. agents $j \in \{1, C\}$) are excluded, as these agents do not have another meeting in τ time steps. A weight for each meeting location (Alg. 3, line 11) is computed by averaging the maximum weight paths each agent would take to the location. A location is randomly chosen from a set of "high" weights (Alg. 3, line 12), where $0 \leq \gamma \leq 1$, to prevent multiple separate meetings from inadvertently choosing the same location. Once a location is chosen, agents update their first and last meeting values and their time budget (Alg. 3, line 13). An example of Algorithm 3 is provided in Fig. 3. Next, we provide details on the path planning methods.

VI. Path Planning

A. Information Metric

We first describe an information metric for the value of observing different areas of the wildfire. The entropy of the belief of a tree is $H_i(x_i^t) = \sum_{x_i^t} b(x_i^t) \log b(x_i^t)$. The expected conditional entropy is, $\overline{H}_i(x_i^t \mid y_i^t) = -\sum_{x_i^t} \sum_{y_i^t} p(y_i^t) b(x_i^t) \log b(x_i^t) \geq 0$, where $p(y_i^t) = \sum_{x_i^t} p(y_i^t \mid x_i^t) b(x_i^t)$. This quantity describes the expected decrease in entropy in the belief $b(x_i^t)$ after measuring tree i using the camera sensor model (1). The mutual information gain for a path $\mathcal{P}_k$ [9] is, $\mathcal{T}(\mathcal{P}_k) = \sum_{q_i \in \mathcal{P}_k} \sum_{j \in I(q_i)} H_j(x_j^t) - \overline{H}_j(x_j^t \mid y_j^t)$. For the multi-agent informative planning problem, we use residual information [9], [10], $\mathcal{T}(\mathcal{B} \cup \mathcal{P}_k) - \mathcal{T}(\mathcal{B})$, where the set $\mathcal{B}$ accounts for other paths taken prior to planning path $\mathcal{P}_k$. Intuitively, the residual information motivates agents to observe different areas of the forest as multiple observations of the same locations have diminishing returns. Each agent needs to compute a path which maximizes the metric, starts at their current position, and passes through their meetings F_k and L_k at the correct times. This problem can be re-framed as finding a maximum path weight given a length constraint, as we discuss next.

B. Maximizing Path Weight Given Length Constraint

The problem of finding a maximum weight path on a graph given a path length constraint and start and end locations is known as the orienteering problem [33]. As the orienteering problem is NP-hard, prior work has focused on solutions with sub-optimality bounds. Since our framework relies on solving this problem many times to set meeting locations and to plan paths to meetings, we use a fast heuristic. Agents then re-plan their paths between meetings to improve their performance.

We now describe our heuristic approach. The value of moving to a tree is based on the residual information,

$$\lambda_i = \mathcal{T}(\mathcal{B} \cup \{q_i\}) - \mathcal{T}(\mathcal{B}). \tag{4}$$

We also add a small positive bias ϵ to the expected conditional entropy to account for situations where the conditional entropy is zero for all trees, e.g. when the initial belief of all agents is the ground truth. Given the weights $\{\lambda_i \ \forall i\}$, a start and end location, and a total path length, we create a directed, acyclic graph (DAG) representation by augmenting each location with a distance. For an agent to plan a path from location e to location h with a maximum length of l, the DAG root vertex is (e, l). The children of this vertex are the locations f that satisfy $\|f - h\|_\infty \leq l - 1$ (i.e. the agent moves closer to h) and $\|e - f\|_\infty \leq 1$ (i.e. the new location satisfies the agent dynamics). The location f is then added to the DAG by adding the vertex $(f, l-1)$ with edges from (e, l) to $(f, l-1)$ with weight λ_f. Likewise, the children of f are the locations g that satisfy $\|g - h\|_\infty \leq l - 2$ and $\|f - g\|_\infty \leq 1$. This process continues until there are no more locations to add as decreasing the distance after each new location is added enforces that the agent ends

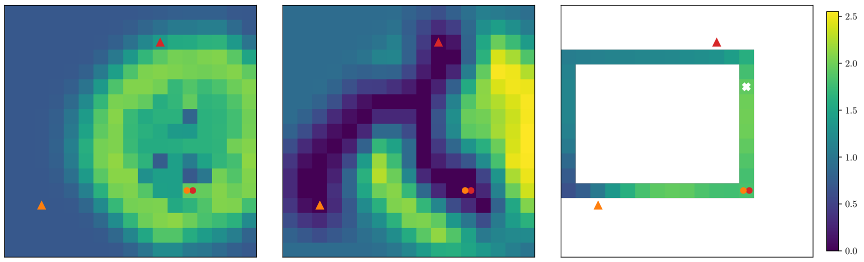

Fig. 3: Example of scheduling a next meeting. (left) The expected conditional entropy is visualized as a heatmap. The orange and red agents are meeting (circles) and each have another meeting to satisfy (orange and red triangles, respectively). (center) The residual information after each agent adds their stored paths and meeting locations. (right) The reachable meeting locations and their weights, computed by averaging the highest weight paths by each agent. The chosen meeting location is denoted by the white X.

Algorithm 4 `Search`

```
1: Input: start $e$, end $f$, length $l$, belief $\{b(x_i^t)\ \forall i\}$,
          observed locations $\mathcal{B}$
2: Output: path $\mathcal{P}$, path weight $w$
3: Compute weights for lattice locations (4)
4: Build directed acyclic graph with $(e, l)$ as the root vertex
5: Use Bellman-Ford to find longest path $\mathcal{P}$ from vertex
   $(e, l)$ to vertex $(f, 0)$ and its associated weight $w$
```

at h after l actions. The Bellman-Ford algorithm is then used to find a maximum weight path (Alg. 4). The `Search` function therefore has $\mathcal{O}(\tau^4)$ time complexity, which is also the dominating complexity in our framework, as this function is used for both scheduling meetings and planning paths.

C. Individual and Team Path Planning

Agents perform team path planning (Alg. 5) when meeting. The team orienteering problem is challenging and we adopt the multi-agent strategy from [9], which is a sequential allocation planning method. Sequential allocation is an appealing approach as it is a relatively straightforward method to implement for a variety of multi-agent planning problems, and there is significant research on sub-optimality bounds. Furthermore, as our main contribution is a scheduling framework, this method can be changed to any other appropriate planning method. First, one agent plans a path ignoring other agents and updates the set of observed locations $\mathcal{B}$. Each subsequent agent then plans a path, accounting for the previous paths through the set $\mathcal{B}$. To plan paths for two separate meetings, agents first plan from their current position to their next meeting, and add a path from their next meeting to their last meeting (Alg. 5, lines 9 and 10). If agents only have one meeting, a single path is planned with a longer length constraint (Alg. 5, line 7). Agents store the nominal paths computed by the joint path planning to use when there is no communication between meetings (Alg. 5, line 14), which requires $\mathcal{O}(\tau)$ space. Agents remove elements from the stored paths during individual path planning, as detailed next, and therefore $\mathcal{R}_k$ does not grow unbounded. An example of Algorithm 5 is provided in Fig. 4.

Algorithm 5 Team Path Planning

```
1: Input: current meeting $s$, merged belief $\{\bar{b}(x_i^t)\ \forall i\}$,
          meeting interval $\tau$, agent data $z_k^t, F_k, L_k, \mathcal{R}_k$
2: Initialize observed locations, $\mathcal{B} \leftarrow \emptyset$
3: for each agent's set of stored paths $\mathcal{R}_k$ do
4:     for each path $\mathcal{P}_j \in \mathcal{R}_k$ do $\mathcal{B} \leftarrow \mathcal{B} \cup \{I(q) \mid q \in \mathcal{P}_j\}$
5: for each agent $k \in s$ do
6:     if $j \in \{1, C\}$ then
7:         $\mathcal{P}_k, w \leftarrow$ Search$(z_k^t, L_k, 2\tau, \{\bar{b}(x_i^t)\ \forall i\}, \mathcal{B})$
8:     else
9:         $\mathcal{P}_F, w \leftarrow$ Search$(z_k^t, F_k, \tau, \{\bar{b}(x_i^t)\ \forall i\}, \mathcal{B})$
10:        $\mathcal{P}_L, w \leftarrow$ Search$(F_k, L_k, \tau, \{\bar{b}(x_i^t)\ \forall i\}, \mathcal{B})$
11:        $\mathcal{P}_k \leftarrow \mathcal{P}_F \cup \mathcal{P}_L$
12:    Update observed locations, $\mathcal{B} \leftarrow \mathcal{B} \cup \{I(q) \mid q \in \mathcal{P}_k\}$
13: for each agent $k \in s$ do
14:    Update paths, $\mathcal{R}_k \leftarrow \mathcal{R}_k \cup \{\mathcal{P}_j \mid j \in s, j \neq k\}$
```

Algorithm 6 Individual Path Planning

```
1: Input: agent belief $\{b(x_i^t)\ \forall i\}$, time budget $d_k$,
          stored paths $\mathcal{R}_k$, position $z_k^t$
2: Output: Planned agent path $\mathcal{P}_k$
3: Initialize observed locations, $\mathcal{B} \leftarrow \emptyset$
4: for each stored path $\mathcal{P}_j \in \mathcal{R}_k$ do
5:     Remove first location $q$ from path, $\mathcal{P}_j \setminus q$
6:     Update observed locations, $\mathcal{B} \leftarrow \mathcal{B} \cup I(q)$
7: $\mathcal{P}_k, w \leftarrow$ Search$(z_k^t, F_k, d_k, \{b(x_i^t)\ \forall i\}, \mathcal{B})$
```

Agents individually perform receding horizon style planning to account for the wildfire process updating as they move (Alg. 6). Agents use and remove the first location in their stored nominal paths to account for actions of other agents. The weights (4) are then calculated from the belief and the set $\mathcal{B}$. In the next section, we present simulation experiments to demonstrate the framework performance.

VII. Simulations

For the simulations, the forest is a lattice of size $d_1 = d_2 = 25$ for a total of 625 trees and a state space of size 10^{298}. The camera sensing area is $h = w = 3$ and the

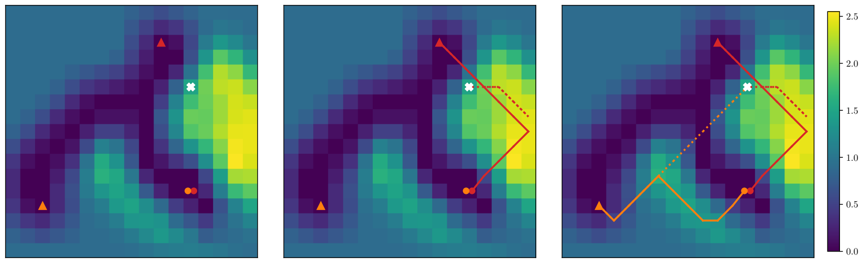

Fig. 4: Example of joint path planning, based on Fig. 3. (left) Residual information heatmap and next meeting location (white X). (center) The red agent first plans a path from its position to its next meeting (solid line), then from its next meeting to its last meeting (dashed line). (right) Given the red agent's path, the orange agent plans its path. Each agent then stores the other agent's nominal path. Note that the `Search` heuristic produces backtracking paths. Paths are improved by re-planning between meetings.

accuracy is $p_c = 0.95$. While it may seem that the camera is extremely accurate, there is a $0.95^9 = 0.63$ probability that the image returns the true state for all trees in the image. Furthermore, the relatively small field of view also reduces the filter performance due to partial observability. All agents use the ground truth as the initial belief. For scheduling meetings, $\gamma = 0.9$ was used (Alg. 3, line 12).

Performance Metric. Performance for information-based frameworks is typically based on the remaining entropy after running the framework [9], [10]. In addition, the underlying model is typically static (e.g. visual reconstruction) or slowly changing (e.g. on the scale of weeks or months), and does not have an absorbing state. However, the wildfire process quickly reaches an absorbing state which corresponds to no fires in the forest. If the agents never take measurements or move, their belief will converge to 100% accuracy and zero entropy as the initial fire quickly burns down the forest. As a result, poor coordination appears to be successful. Therefore, we introduce a metric to evaluate the transient performance,

$$\frac{1}{T}\sum_{t=1}^{T}\frac{\left|\bigcup_{k\in\{1,\dots,C\}}\left\{i \in I(z_k^t) \mid x_i^t = F\right\}\right|}{\left|\left\{i \in \{1,\dots,d_1 d_2\} \mid x_i^t = F\right\}\right|},$$

where T is the total number of simulation time steps. This metric represents the average fraction of ground truth fires covered by the agents over a full simulation. For aggressive wildfires, this metric is typically less than one due to the comparable rate of agents moving and the wildfire spreading. When there are no fires at a given time step, the corresponding term in the summation is zero.

Comparison Methods. We compare with two baseline algorithms. First, there is no communication and at each time step, agents predict their belief 8 steps in the future and choose the highest entropy location that is 8 actions away to move towards. Second, the agents are considered a single team and always communicate; we again stress this is infeasible in practice as it requires long-range communication and only serves as an ideal baseline. The agents have a single belief that is updated in a centralized manner using all observations. Every 8 time steps, the agents use a sequential allocation strategy, similar to Algorithm 3, to determine a unique F_k and a nominal path for each agent. For other time steps, agents simply execute their planned path. Both baselines are heuristics, as the multi-agent persistent exploration problem with and without unlimited communication are open problems.

Results. Fig. 5 shows simulation results for two cases, (top) process update rate $\rho = 1$ with $T = 60$ total time steps and (bottom) $\rho = 2$ with $T = 120$. For both cases, 10 simulations were run, and the first quartile, median, and third quartile are shown for each configuration of parameters. For all cases, the "no communication" baseline is ineffective. The "team communication" baseline has the best performance, due to the benefit of communication between all agents at every time step. Furthermore, the team communication baseline highlights the benefit of maintaining a single belief using all agent observations without requiring agents to meet to communicate. Overall, our framework is better than the no communication baseline given enough agents, and is comparable to the team communication baseline for many cases. Finally, it is clear that the $\rho = 1$ case is a challenging scenario which mainly requires more agents to be effective.

VIII. Conclusions

In this work, we proposed a novel scheduling method along with a decentralized framework to effectively coordinate UAVs in surveying an aggressive wildfire. Since our framework solves the orienteering problem many times, we plan to investigate solution methods with theoretical guarantees or improved time complexity (or both). We also plan to extend our framework to dynamically scheduling meetings with varying number of agents and meeting intervals, as well as integrate the framework with our work on a decentralized control scheme [6].

References

[1] CAL FIRE, "California fire historical incident statistics," https://www.fire.ca.gov/incidents/, Accessed 2019-09-04.

[2] U. G. C. R. Program, "National climate assessment report," https://nca2014.globalchange.gov/report, Accessed 2019-09-04.

[3] Y. Meng, J. V. Nickerson, and J. Gan, "Multi-robot aggregation strategies with limited communication," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 2691–2696.

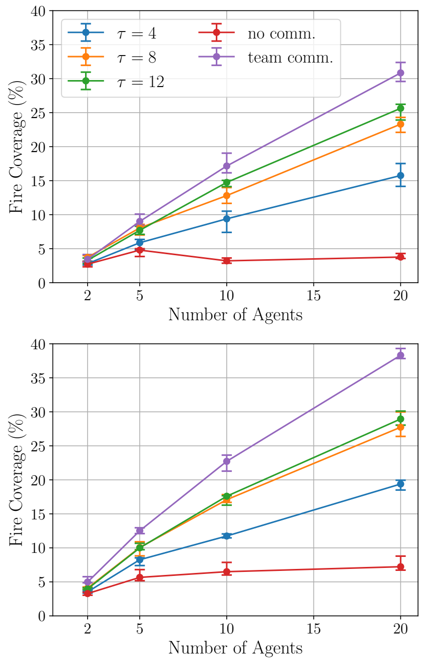

Fig. 5: Simulation results for different wildfire scenarios, (top) $\rho = 1$ with $T = 60$ and (bottom) $\rho = 2$ with $T = 120$. The "no communication" baseline is ineffective for all cases, whereas the "team communication" baseline benefits from additional communication. Our framework outperforms the no communication baseline and is comparable to the team communication baseline for many cases. In general, more agents and longer meeting intervals τ improve the framework performance, with diminishing returns as τ increases.

[4] R. N. Haksar and M. Schwager, "Controlling large, graph-based MDPs with global control capacity constraints: An approximate lp solution," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 35–42.

[5] R. N. Haksar, J. Lorenzetti, and M. Schwager, "Scalable filtering of large graph-coupled hidden Markov models," in *2019 IEEE Conference on Decision and Control (CDC)*, Dec 2019, in press.

[6] R. N. Haksar and M. Schwager, "Distributed deep reinforcement learning for fighting forest fires with a network of aerial robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 1067–1074.

[7] C. Nowzari and J. Cortés, "Self-triggered coordination of robotic networks for optimal deployment," *Automatica*, vol. 48, no. 6, pp. 1077 – 1087, 2012.

[8] Y. Kantaros and M. M. Zavlanos, "Distributed communication-aware coverage control by mobile sensor networks," *Automatica*, vol. 63, pp. 209 – 220, 2016.

[9] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, no. 1, pp. 707–755, Apr. 2009.

[10] M. Corah and N. Michael, "Efficient online multi-robot exploration via distributed sequential greedy assignment," in *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, July 2017.

[11] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "A scalable information theoretic approach to distributed robot coordination," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 5187–5194.

[12] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Dec-MCTS: Decentralized planning for multi-robot active perception," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 316–337, 2019.

[13] S. Moon and E. W. Frew, "A communication-aware mutual information measure for distributed autonomous robotic information gathering," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3137–3144, Oct 2019.

[14] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to persistent monitoring tasks," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1106–1118, Oct 2016.

[15] C. Phan and H. H. T. Liu, "A cooperative UAV/UGV platform for wildfire detection and fighting," in *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing*, Oct 2008, pp. 494–498.

[16] D. Stipaničev, M. Štula, D. Krstinić, L. Šerić, T. Jakovčević, and M. Bugarić, "Advanced automatic wildfire surveillance and monitoring network," in *6th International Conference on Forest Fire Research*, 2010.

[17] K. D. Julian and M. J. Kochenderfer, "Distributed wildfire surveillance with autonomous aircraft using deep reinforcement learning," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 8, pp. 1768–1778, 2019.

[18] S. Susca, F. Bullo, and S. Martinez, "Monitoring environmental boundaries with a robotic sensor network," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 2, pp. 288–296, March 2008.

[19] S. W. Feng, S. D. Hand, and J. Yu, "Efficient algorithms for optimal perimeter guarding," in *Proceedings of Robotics: Science and Systems*, June 2019.

[20] L. Merino, F. Caballero, J. R. Martínez-de Dios, I. Maza, and A. Ollero, "An unmanned aircraft system for automatic forest fire monitoring and measurement," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1, pp. 533–548, Jan 2012.

[21] N. K. Ure, S. Omidshafiei, B. T. Lopez, A. Agha-Mohammadi, J. P. How, and J. Vian, "Online heterogeneous multiagent learning under limited communication with applications to forest fire management," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 5181–5188.

[22] M. Kumar, K. Cohen, and B. Homchaudhuri, "Cooperative control of multiple uninhabited aerial vehicles for monitoring and fighting wildfires," *Journal of Aerospace Computing, Information, and Communication*, vol. 8, no. 1, pp. 1–16, 2011.

[23] H. X. Pham, H. M. La, D. Feil-Seifer, and M. C. Deans, "A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2018.

[24] P. B. Sujit, D. Kingston, and R. Beard, "Cooperative forest fire monitoring using multiple UAVs," in *2007 46th IEEE Conference on Decision and Control*, Dec 2007, pp. 4875–4880.

[25] K. A. Ghamry, M. A. Kamel, and Y. Zhang, "Cooperative forest monitoring and fire detection using a team of UAVs-UGVs," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 1206–1211.

[26] P. Stone and M. Veloso, "Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork," *Artificial Intelligence*, vol. 110, no. 2, pp. 241 – 273, 1999.

[27] N. Roy and G. Dudek, "Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations," *Autonomous Robots*, vol. 11, no. 2, pp. 117–136, Sep 2001.

[28] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain, "Cooperative forest fire surveillance using a team of small unmanned air vehicles," *International Journal of Systems Science*, vol. 37, no. 6, pp. 351–360, 2006.

[29] M. Lauri and R. Ritala, "Planning for robotic exploration based on forward simulation," *Robotics and Autonomous Systems*, vol. 83, pp. 15 – 31, 2016.

[30] A. Somanath, S. Karaman, and K. Youcef-Toumi, "Controlling stochastic growth processes on lattices: Wildfire management with robotic fire extinguishers," in *53rd IEEE Conference on Decision and Control (CDC)*, Dec 2014, pp. 1432–1437.

[31] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[32] T. P. Hill, "Conflations of probability distributions," *Transactions of the American Mathematical Society*, vol. 363, no. 6, pp. 3351–3372, 2011.

[33] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering problem: A survey of recent variants, solution approaches and applications," *European Journal of Operational Research*, vol. 255, no. 2, pp. 315 – 332, 2016.