Coordinating Human-UAV Teams in Disaster Response

Feng Wu[†] **Sarvapali D. Ramchurn**[‡] **Xiaoping Chen**[†]
[†]Computer Science and Technology, University of Science and Technology of China, Hefei, China
[‡]Electronics and Computer Science, University of Southampton, Southampton, UK
wufeng02@ustc.edu.cn, sdr1@soton.ac.uk, xpchen@ustc.edu.cn

Abstract

We consider a disaster response scenario where emergency responders have to complete rescue tasks in dynamic and uncertain environment with the assistance of multiple UAVs to collect information about the disaster space. To capture the uncertainty and partial observability of the domain, we model this problem as a POMDP. However, the resulting model is computationally intractable and cannot be solved by most existing POMDP solvers due to the large state and action spaces. By exploiting the problem structure we propose a novel online planning algorithm to solve this model. Specifically, we generate plans for the responders based on Monte-Carlo simulations and compute actions for the UAVs according to the value of information. Our empirical results confirm that our algorithm significantly outperforms the state-of-the-art both in time and solution quality.

1 Introduction

First Responders (FRs) face major challenges in planning search and rescue missions in the aftermath of major disasters. In such domains, FRs have to perform rescue tasks (e.g., extinguishing a fire or providing first aid) to minimize loss of life and costs (e.g., time or fuel costs). Thus, they have to plan their path to the tasks (as these may be distributed in space) and complete them, taking into account the status of the current tasks (e.g., health of victims or building fire) and the environment (e.g., if a fire or radioactive cloud is spreading). Uncertainty in the environment (e.g., fires spreading or riots erupting) or in the responders' abilities to complete tasks (e.g., some may be tired or get hurt) means that plans are likely to change continually to reflect the prevailing assessment of the situation. However, information about the tasks and the environment is often very limited [Fiedrich and Burghardt, 2007; Wu and Jennings, 2014]. Hence, it is crucial for the FRs to collect as much information as possible while performing their tasks.

To support the work of FRs, Unmanned Aerial Vehicles (UAVs) have been widely deployed for information gathering in disaster response [Adams and Friedland, 2011]. They are fast and can collect information with onboard sensors (e.g., photos with cameras). Crucially, they can be deployed in hazardous environments (e.g., due to fire or radioactive cloud) and thus help gather information without putting people in harm's way. To date, UAVs have not been well integrated with planning systems that explicitly model the uncertainty in the environment and suggests courses of actions both for FRs and UAVs [Adams and Friedland, 2011]. This is challenging because, on one hand, FRs need up-to-date information from UAVs about the disaster site in order to better plan their actions, while on the other hand, UAVs must predict where FRs may move to next so they can concentrate on these locations for information gathering. Otherwise, UAVs may fail to collect information critically required by FRs. It is therefore essential to develop novel solutions that coordinate human-UAV teams in disaster response.

Against this background, we develop a novel approach for such *human-agent collectives* [Jennings *et al.*, 2014] in disaster response where a planner agent gathers information with the assistance of UAVs and plans actions on behalf of FRs. In more detail, we consider a scenario involving rescue tasks distributed in a physical space over which a radioactive cloud is spreading. Tasks need to be completed by the FRs before the area is completely covered by the cloud. As FRs will die from radiation exposure, they need to know the radioactivity levels at the task sites and along the paths to the tasks. Crucially, such information should be collected by UAVs before FRs decide their next actions. We model this problem as a Partially Observable Markov Decision Process (POMDP) given that the environment is uncertain and only partial information about the radioactive cloud is observable to the UAVs. Although a POMDP provides a nice and coherent mathematical model that optimizes both actions for the UAVs and the FR in a single framework, solving large POMDPs is usually computationally demanding. As a result, our problem is intractable for most existing methods. Therefore, we propose a novel algorithm that combines planning and sensing based on *Monte-Carlo simulation* and *value of information* to compute policies both for FRs and UAVs. By so doing, we closely couple online planning (for FRs) with active sensing (for UAVs) to achieve more effective rescue missions. Furthermore, our algorithm is *anytime* and *scalable* for large domains. We empirically evaluated our algorithm using a benchmark simulator for disaster response and show that our algorithm outperforms the leading POMDP solver (i.e., POMCP) in the bench-

mark domain with faster runtime and better solution quality.

2 Related Work

Previous approaches for planning in disaster response typically assume the current state is fully observable to the planner agent [Musliner *et al.*, 2006; Wu *et al.*, 2015a; Ramchurn *et al.*, 2015c; 2015a]. However, this assumption is usually unrealistic for real-world applications. In early work for disaster response, POMDPs have been used to make token coordination decisions for a team of agents [Xu *et al.*, 2005] but not for task planning and information gathering. Most recently, [Wu *et al.*, 2015b] proposed to boost the performance for RoboCup Rescue simulation using POMDPs but this approach requires predefined macro-actions. There are related work using POMDPs for UAV based applications such as target search / detection / tracking [Geyer, 2008; Chanel *et al.*, 2013; Ramchurn *et al.*, 2015b; Bernardini *et al.*, 2015], collision avoidance [Bai *et al.*, 2012b] and path planning [Ragi and Chong, 2012]. In disaster response, POMDPs were considered as a useful model for search and rescue operations using UAVs [Waharte and Trigoni, 2010]. However, none of them integrate information gathering with task planning to guide both FRs and UAVs in disaster response.

In the past decades, many approaches have been proposed to solve POMDPs. Our work relates to the online planning literature where agents interleave planning with execution in each decision step [Silver and Veness, 2010; Wu *et al.*, 2010; Somani *et al.*, 2013] . Similar to our method, problem structures are often exploited to efficiently solve large POMDPs [Ong *et al.*, 2010; Bai *et al.*, 2012a]. However, it is worth noting that our goal in this paper is not to propose a better method for general POMDPs. Instead, *we aim to introduce a subclass of POMDPs useful for common disaster response scenarios and propose a novel algorithm that is faster than the state-of-the-art to solve such problems*. In what follows, we will present our scenario and model and discuss why this model is nontrivial and cannot be easily tackled by existing solvers.

3 Disaster Scenario

We draw upon previous work for the disaster scenario in which a satellite leaks radioactive particles after having crashed in a sub-urban area, causing damage to infrastructure and injuring civilians as a result [Ramchurn *et al.*, 2015c]. These particles are gradually spreading over the area, threatening to contaminate food reserves and people. Hence, emergency services are deployed to evacuate the casualties and key assets before they are engulfed by the radioactive cloud.

Let G denote a grid overlaid on top of the disaster space, and assume the satellite debris, casualties, assets, and actors are located at various coordinates $(x,y) \in G$ in this grid. The radioactive cloud induces a radioactivity level $l \in [0, l_{\max}]$ at every point that it covers. Since an FR in the disaster space may receive radiation doses that may, at worst, be life-threatening, we assign her a health level $h \in [0, h_{\max}]$ that decreases based on the radiation dose that she received from the space. Given the invisibility of radiation, the information about the radiation levels has to be collected by sensors placed in the disaster space. As the radioactive cloud are also likely to shift across the disaster space due to wind direction and speed, UAVs are deployed, equipped with geiger counters and other sensors in order to loiter and monitor the state of the cloud. Note that the information collected by UAVs is uncertain due to the poor positioning of the sensors and the variations in wind speed and direction.

In our settings, rescue tasks are performed by an FR working in the disaster space, guided by a planner agent. While FRs execute tasks, the agent collects information about the environment using UAVs, generates plans and communicates its instructions directly to the FR. To capture the uncertainty (e.g., the spread of the radioactive cloud) and partial observability (UAVs can only collect the information near their current location) of the domain, we model this problem as a POMDP which we present in the following section.

4 The POMDP Model

We model the aforementioned disaster response problem[1] using *Partially Observable Markov Decision Process* (POMDP) defined as tuple $\mathscr{M} = \langle S, A, P, R, \Omega, O, T \rangle$, where:

- $S = S_{FR} \times S_{RC} \times S_{UAV}$ is the state space where: $S_{FR} := \{(x,y), h | (x,y) \in G, h \in [0, h_{\max}]\}$ are the variables for the FR with her coordinates (x,y) in the grid G and her health level h; $S_{RC} := \{l_{x,y} | l_{x,y} \in [0, l_{\max}], (x,y) \in G\}$ are the variables for the radioactive cloud with the radioactive level $l_{x,y}$ for every cell (x,y) in G; and $S_{UAV} := \{(x,y)_i | (x,y)_i \in G, i \in 1..n\}$ are the variables for n UAVs with the $(x,y)_i$ coordinate for UAV i.
- $A = A_{FR} \times A_{UAV}$ is the action space where: A_{FR} and A_{UAV} are the action variables for the FR and the UAVs respectively to move in the grid G.
- $P(s'|s,a) = P(s'_{fr}|s_{fr}, s_{rc}, a_{fr}) P(s'_{rc}|s_{rc}) P(s'_{uav}|s_{uav}, a_{uav})$ is the state transition function where: $P(s'_{fr}|s_{fr}, s_{rc}, a_{fr})$ is the transition function for the FR as her next state $s'_{fr} \in S_{FR}$ depends on her current state $s_{fr} \in S_{FR}$, the current state of the radioactive cloud $s_{rc} \in S_{RC}$, and the action taken by her $a_{fr} \in A_{FR}$; $P(s'_{rc}|s_{rc})$ is the transition function for the radioactive cloud as the next state $s'_{rc} \in S_{RC}$ only depends on the current state $s_{rc} \in S_{RC}$, not the actions of the FR and the UAVs; and $P(s'_{uav}|s_{uav}, a_{uav})$ is the transition function for the UAVs as their next state $s'_{uav} \in S_{UAV}$ depends on their current state $s_{uav} \in S_{UAV}$ and their actions $a_{uav} \in A_{UAV}$.
- $R(s,a) = R(s_{fr}, a_{fr}) + R(s_{uav}, a_{uav})$ is the reward function where: $R(s_{fr}, a_{fr})$ is the reward function for the FR and $R(s_{uav}, a_{uav})$ is the reward function for the UAVs.
- $\Omega := \{l_i | l_i \in [0, l_{\max}], i \in 1..n\}$ is the observation space where l_i is the sensor reading of the radioactive level recorded by UAV i at its current location.
- $O(o|a,s') = O(o|s'_{uav}, s'_{rc})$ is the observation function modeling the probability of the UAVs observing $o \in \Omega$ (i.e., the sensor reading) given that the UAVs' next state is s'_{uav} and the next state of radioactive cloud is s'_{rc}.

[1] Our model generalizes to other disaster scenarios with the radioactive cloud being replaced by other uncertain event (e.g., fire).

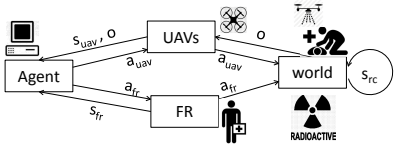


Figure 1: Interactions between FRs, UAVs, and the world.

- T is the horizon, i.e., the deadline of the tasks.

We assume that the FR and UAVs know their current locations using GPS and the FR can directly report her health level when doing the tasks. Hence the states of the FR and UAVs s_{fr}, s_{uav} are fully observable[2] to the planner agent. However, the state of the radioactive cloud s_{rc} is hidden to the agent and therefore UAVs are deployed to collect information about it. In our setting, each UAV can record the reading of the radioactive level at its current location and hence the state of the radioactive cloud is partially observable to the planner agent.

At each time step, as shown in Figure 1, the planner agent first selects an action a_{fr} for the FR and an action a_{uav} for the UAVs. As received from the agent, the actions are executed by the FR and the UAVs. This results a state transition in the world and an observation o about the state of the radioactive cloud s_{rc} is received by the UAVs. After that, the UAVs send back their current state and observation s_{uav}, o to the agent and the FR also reports her current state s_{fr}. Then, the agent proceeds to the next time step and the process repeats until the horizon T is reached. Thus, the goal is to find actions that ensure that UAVs gather as much information as possible to maximize the number of tasks that the FR is able to complete without being harmed by the cloud.

Given the POMDP, we must solve it and compute a policy that maps *belief states* (i.e., distributions over states) to actions, i.e., $\pi : \Delta(S) \rightarrow A$. In POMDPs, the quality of a policy π is evaluated based on the *value function* $V^{\pi}(\cdot)$. The objective of solving a POMDP is to find the optimal policy π^* in the policy space that maximizes the expected value given the initial belief state b^0: $\pi^* = \text{argmax}_{\pi \in \Pi} V^{\pi}(b^0)$.

Although many techniques exist for solving POMDPs, there are several challenges inherent in our problem that make it nontrivial to be solved by them:

1. **The state space of our problem is very large**. Suppose that the size of grid G is $|G|$, the number of health levels is N_h and the number of radioactive levels is N_l. The size of the overall state space is $\mathcal{O}(|G|N_h \cdot |G|^n \cdot N_l^{|G|})$. Due to the huge state space of our problem, it is computationally intractable for most existing POMDP solvers. This is also known as "the curse of dimensionality".
2. **The action and observation spaces of our problem are also very large**. Both grow *exponentially* with the number of UAVs. This makes existing approximate POMDP algorithms inefficient because the policy space becomes huge given large number of actions and observations. This is also known as "the curse of history" in the POMDP literature [Kaelbling *et al.*, 1998].

[2]Information about the FR and UAVs is not the focus of the sensing tasks in this paper but will be considered in future work.

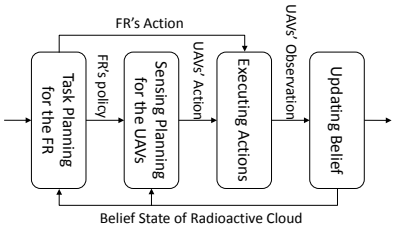


Figure 2: Online planning with active sensing.

3. **The transition function cannot be represented explicitly with probability values**. Notice that the spreading of the radioactive cloud involves a very complex physical process. Therefore, given the current state of the radioactive cloud s_{rc} and a next state s'_{rc}, it is very difficult to compute the transition probability $P(s'_{rc}|s_{rc})$ especially when the state space of the radioactive cloud is huge. Fortunately, we can stochastically generate a next state s'_{rc} given s_{rc} based on some physical rules. Thus, approaches that require the use of transition probabilities cannot be applied to our problem.

In the next section, we present our planning algorithm for our POMDP formulation of the disaster response problem.

5 Online Planning Algorithm

To address the aforementioned challenges, we adopt an online planning approach so that we only need to compute the best action for the current belief state at each time step, instead of the entire belief space. The key observation we make is that the information gathering and task execution are two processes in our problem, where the former one is performed by the UAVs and the later one is accomplished by the FR. Based on this, we divide the whole algorithm into two procedures, one for task planning and the other for information gathering. However, these two procedures cannot be executed completely separately because the FR needs the information about the radioactive cloud in order to effectively plan her action, while the UAVs must know which tasks will be taken by the FR to actively collect information about those tasks.

To this end, we propose the OPAS algorithm — *Online Planning with Active Sensing*. As shown in Figure 2, in each time step, the FR computes a policy and specifics her action and the UAVs use this policy to update their sensing action. Notice that the UAVs' actions are dependent on what policy is chosen for the FR. In turn, for the FR, she can update and enhance the policy she computes based on information reported by UAVs as they fly over the areas chosen. Here, the key procedures are: 1) How to compute the FR's policy; 2) How to choose the best sensing action for the UAVs; 3) How to update the belief state of the radioactive cloud given the observation from the UAVs. In the following sections, we will describe these procedures in detail.

5.1 Simulation-Based Task Planning

To compute the policy for the FR, we first get her current state, which is fully observable by the agent. Starting from

Algorithm 1: Simulation-Based Task Planning

Input: The belief state of radioactive cloud $b_{rc}(\cdot)$

$s^*_{fr} \leftarrow$ the current state of the FR, $\tau \leftarrow$ the current time step
repeat
  $b^{\tau}_{rc} \leftarrow b_{rc}, s_{fr} \leftarrow s^*_{fr}$
  for $t \leftarrow \tau$ **to** T **do**
    foreach $a_{fr} \in A_{FR}$ **do**
      `# Compute the Q value`
      $Q(s_{fr}, b^t_{rc}, a_{fr}) \leftarrow R(s_{fr}, a_{fr}) +$
        $\sum_{s'_{fr}} Pr(s'_{fr}|s_{fr}, b^t_{rc}, a_{fr}) V(s'_{fr}, b^{t+1}_{rc})$
    $a^{best}_{fr} \leftarrow \text{argmax}_{a_{fr}} Q(s_{fr}, b^t_{rc}, a_{fr})$
    `# Update the value function`
    $V(s_{fr}, b^t_{rc}) \leftarrow Q(s_{fr}, b^t_{rc}, a^{best}_{fr})$
    if s_{fr}: tasks are completed **or** FR is dead **then**
      Terminate the simulation
    $b^{t+1}_{rc} \leftarrow$ update the belief state based on b^t_{rc}
    $s_{fr} \sim P(\cdot|s_{fr}, s_{rc}, a^{best}_{fr})$
until it is timeout.
`# Choose the best action for the FR`
return $a^*_{fr} \leftarrow \text{argmax}_{a_{fr}} Q(s^*_{fr}, b_{rc}, a_{fr})$

the FR's current state, several simulations are run for her during planning. Specifically, for each possible action of the FR, we compute the Q-values. After that, the best action for the FR is selected and the value function is updated. This process advances to the next time step with the new states drawing from the transition model until the planning horizon is reached or the process is terminated. We repeat the simulation multiple times until it runs out of time for planning. Finally, the best action for the FR is computed based on the Q-values. The main process is outlined in Algorithm 1. Indeed, this algorithm is analogy to RTDP-Bel [Geffner and Bonet, 1998] for POMDPs, which converges to the optimal policy given sufficient number of simulations. Note that the state of the radioactive cloud is independent of the actions of the FR and the UAVs. Hence the belief state is updated based on a Markov chain for the spreading process. By so doing, the FR makes her decision based on the knowledge about the radioactive cloud that is current available.

One remaining question is how to efficiently compute the Q-values in the algorithm. Typically, this is done using the Bellman equation as:

$$Q(s_{fr}, b_{rc}, a_{fr}) = R(s_{fr}, a_{fr}) + \sum_{s'_{fr} \in S_{FR}} Pr(s'_{fr}|s_{fr}, b_{rc}, a_{fr}) \cdot V(s'_{fr}, b'_{rc}) \quad (1)$$

However, this equation requires to enumerate all possible states of the FR and the radioactive cloud. Given that our state space is very large, the computation of the exact Q-values will be intractable. To address this, we approximate the Q-values with Monte-Carlo simulations. The basic idea is to repeatedly draw K state samples based on the transition model given the current states and the action. Then, the expected value is estimated by the mean value of the samples as:

$$Q(s_{fr}, b_{rc}, a_{fr}) \approx R(s_{fr}, a_{fr}) + \frac{1}{K} \sum_{k=1}^{K} V(s^k_{fr}, s^k_{rc}) \quad (2)$$

where s^k_{fr}, s^k_{rc} are the k-th sample drawn from the transition function: $(s^k_{fr}, s^k_{rc}) \sim P(\cdot, \cdot|s_{fr}, s_{rc}, a_{fr})$ where $s_{rc} \sim b_{rc}(\cdot)$.

Algorithm 2: Information-Based Active Sensing

Input: The belief state $b_{rc}(\cdot)$, the value function $V(s_{fr}, \cdot)$

$s_{fr}, s_{uav} \leftarrow$ the current states of the FR and the UAVs
foreach a_{uav} **do**
  $Q(b_{rc}, s_{uav}, a_{uav}) \leftarrow 0$
  for $k = 1$ **to** K **do**
    `# Simulate the action execution`
    $s_{rc} \sim b_{rc}(\cdot), s'_{rc} \sim P(\cdot|s_{rc})$
    $s'_{uav} \sim P(\cdot|s_{uav}, a_{uav}), o \sim O(\cdot|s'_{uav}, s'_{rc})$
    $b'_{rc} \leftarrow$ update the belief state b_{rc} with o
    `# Compute the value of information`
    $V_{info} \leftarrow \|V(s_{fr}, b'_{rc}) - V(s_{fr}, b_{rc})\|$
    $Q(b_{rc}, s_{uav}, a_{uav}) \leftarrow Q(b_{rc}, s_{uav}, a_{uav}) + V_{info}$
  $Q(b_{rc}, s_{uav}, a_{uav}) \leftarrow R(s_{uav}, a_{uav}) + Q(b_{rc}, s_{uav}, a_{uav})/K$
`# Choose the best action for the UAVs`
return $a^*_{uav} \leftarrow \text{argmax}_{a_{uav}} Q(b_{rc}, s_{uav}, a_{uav})$

5.2 Information-Based Active Sensing

We choose the sensing action for the UAVs based on the Value of Information (VoI) [Howard, 1966] — a well-known concept for information theory. Specifically, given that the FR's state is s_{fr}, we define the VoI as the difference in value for the belief states before and after some sensing action:

$$V^{s_{fr}}_{info}(b_{rc}, b'_{rc}) := \|V(s_{fr}, b'_{rc}) - V(s_{fr}, b_{rc})\| \quad (3)$$

where b_{rc} is the current belief state about the radioactive cloud and b'_{rc} is the belief state after the UAVs taking some sensing action. Intuitively, if the likelihood of some states of the radioactive cloud changes significantly and the expected values of the FR regrading of these states are large, the sensing action of the UAVs is considerably informative. If the sensing action makes no difference to the FR's expected value, then this action is not informative because the information collected by the UAVs is not useful for the FR's decision.

Given an action for the UAVs, there will be many possible observations obtained by them and each observation corresponds to a new belief state. Therefore, the VoI of the UAVs taking action a_{uav} in state s_{uav} is computed as:

$$V^{s_{fr}}_{info}(b_{rc}, s_{uav}, a_{uav}) = \sum_{o \in \Omega} Pr(o|b_{rc}, s_{uav}, a_{uav}) V^{s_{fr}}_{info}(b_{rc}, b^o_{rc})$$

where b^o_{rc} is the new belief state given b_{rc}, a_{uav}, o and

$$Pr(o|b_{rc}, s_{uav}, a_{uav}) = \sum_{s'_{uav} \in S_{UAV}} \sum_{s'_{rc} \in S_{RC}} O(o|s'_{uav}, s'_{rc}) \cdot \sum_{s_{rc} \in S_{RC}} P(s'_{rc}|s_{rc}) P(s'_{uav}|s_{uav}, a_{uav}) b_{rc}(s_{rc}) \quad (4)$$

Proposition 1. *Given the optimal value function for the FR, the sensing action that maximizes the VoI is the optimal action for the UAVs in the POMDP.*

Algorithm 3: Alternative Maximization for Sensing

```
a*_uav ← a random action in A_UAV, Q*_uav ← −∞
repeat
    ε ← 0, θ ← sample a set of waypoints based on Q
    foreach UAV i ∈ 1..n in a random order do
        a_uav ← a*_uav # The current best action
        foreach (x,y) ∈ θ and (x,y) is reachable by UAV i do
            a_uav[i] ← set the target of UAV i as (x,y)
            Q_uav ← compute the expected value of a_uav
            if Q_uav > Q*_uav then
                ε ← ε + (Q_uav − Q*_uav) # Residual
                a*_uav ← a_uav, Q*_uav ← Q_uav
until ε is sufficiently small or it is timeout.
return a*_uav # The best action for the UAVs
```

The proof is omitted due to limit of space but can be done by showing that the sensing action that maximizes the VoI is also the action that maximizes the Bellman equation. Thus, it is the optimal action for the UAVs in the POMDP. Intuitively, the most informative action is also the optimal one for the UAVs because the their only task is to sense the environment and gather the information that is most useful for the FR.

In order to compute the VoI of each sensing action and select the most informative one, we propose an active sensing algorithm shown in Algorithm 2. Firstly, we read the current state of the FR and the UAVs. Then, we estimate the VoI for each sensing action. As aforementioned, the state space about the radioactive cloud is very large. Therefore, we estimate the expected value using simulations. Specifically, we draw a sample of the state from the current belief state and run the simulation with the sensing action chosen. After the simulation, an observation given the states and the sensing action is obtained. Then, we use the observation to update the belief state and compute the new one. Given the two belief states, the VoI is computed using Equation 3. We average the VoI over several simulation samples and add it with the immediate reward of the sensing action:

$$Q^{s_{fr}}(b_{rc}, s_{uav}, a_{uav}) \approx R(s_{uav}, a_{uav}) + \frac{1}{K}\sum_{k=1}^{K} V_{info}^{s_{fr}}(b_{rc}, b_{rc}^{o^k}) \quad (5)$$

where $b_{rc}^{o^k}$ is the k-th belief state given o^k drawn from the probabilities: $o^k \sim Pr(\cdot|b_{rc}, s_{uav}, a_{uav})$. At this point, we obtain an approximate value of the sensing action and accordingly the sensing action with the maximum value is selected.

However, it is costly to consider the value of every sensing action because the action space for the UAVs is very large. To address this, we propose a local search algorithm to find the best sensing action using alternative maximization. The main procedure is outlined in Algorithm 3. Starting with a random sensing action, we sample a set of waypoints along the possible path of the FR as shown in Figure 3. Then, for each UAV in a random order, we generate a sensing action by assigning its location to each waypoint. After that, we compute the expected value of this sensing action and choose it as the best action if it has larger expected value than the previous one. To sample a set of waypoints, we simulate the movement of the FR by selecting the best action based on the Q-value computed by Algorithm 1 and collect the waypoints of the FR in the simulation. We use the same method to compute the expected value for each sensing action as in Algorithm 2. We next consider how to represent the beliefs of the planner.

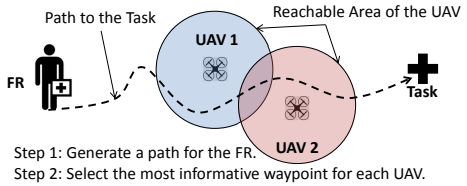


Figure 3: UAV sensing based on FR's waypoints

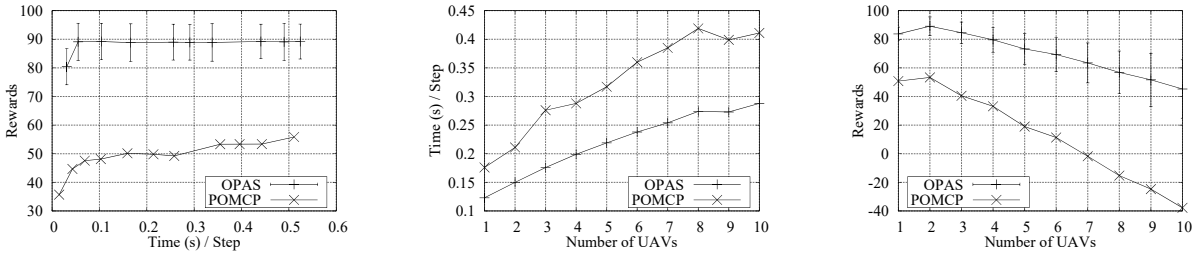
5.3 Particle-Based Belief Updating

Now, a typical method to represent the belief state is to define a probability distribution over all possible states and update it according to the Bayesian rule. Unfortunately, the state space of the radioactive cloud is very large. Therefore, we use a set of weighted state particles to represent the belief state: $b_{rc} \approx \{(s_{rc}^1, w^1), \cdots, (s_{rc}^n, w^n)\}$ s.t. $\sum_{k=1}^{n} w^k = 1$ where the weights sum to one. Notice that the particle-based belief representation has been widely used in many state-of-the-art POMDP solvers to tackle problems with large state space [Thrun, 1999; Silver and Veness, 2010]. When sampling a state from the belief state, it selects a state particle from the particle set with the probability of its weight.

To update the belief state given an sensing action and an observation from the UAVs, we first draw a state of the radioactive cloud from the current belief state. Next, we simulate the process and obtain the next state of the radioactive cloud and the UAVs. Then, we add the new state of the radioactive cloud to the new particle set representing the new belief state with the weight based on the observation function conditioning on the observation. When a sufficient number of new particles have been collected, we normalize the weights of the new particles and return the set as the new belief state.

6 Empirical Evaluation

To evaluate our algorithm, we extended an existing benchmark simulator used to develop prototypes for real-world studies [Ramchurn *et al.*, 2015c]. Specifically, the disaster space is overlaid by a 55×50 grid with obstacles (e.g., buildings, walls, water pools, etc.) and the radioactive clouds. We consider a single FR and a group of UAVs for the response tasks, where the FR must move to the task locations and complete the tasks before the resources are engulfed by the radioactive cloud. Here, a system state consists of: the FR's (x,y)-coordinate in the grid and her health level, the radioactive level for each cell of the grid, and the UAVs' coordinates. For the actions, the FR can move either one of the 8 neighboring cells (unless the cell is occupied by an obstacle) or just stay in her current cell and each UAV can move to any cell in the 10×10 sub-grid centered with its current location. For the transition function and the reward function, we adopt the same settings as in [Ramchurn *et al.*, 2015c]. An observation is defined as a set of sensory readings from the UAVs about



(a) Time vs. Rewards (b) Number of UAVs vs. Time (c) Number of UAVs vs. Rewards

Figure 4: Experimental results on benchmark domains.

the radioactive levels at their current coordinates. To simulate the noisy sensors, random Gaussian noises are added to the sensory readings.

Due to the large state and action space, there are few existing algorithms that can solve our problem. Among them, POMCP [Silver and Veness, 2010] is currently the leading solver for large POMDPs. Particularly, POMCP only requires a black-box simulator, which fits our domains. Therefore, we compared our approach with POMCP and used the same problem instances generated by the simulator. To simplify the tests, we fixed the locations of the radioactive sources but randomized the start location of the FR and the task locations to generate different problem instances. We used the same random seeds so the problem instances for each algorithm were identical. Due to the stochastic nature of our problem, we ran each algorithm 1000 times until the results are statistically meaningful and recorded the mean and standard deviation of the runtime per step and the accumulated rewards obtained from the tests. A machine with a 3.50GHz Intel Core i7 CPU and 8GB RAM was used to produce the results.

Figure 4 summarizes our experimental results. As aforementioned, both OPAS and POMCP are online planning algorithms. Therefore, they must be able to complete the planning process and compute the action with limited amount of time in each decision step to vary the number of simulations allowed. In Figure 4(a), we show the performance of OPAS and POMCP given different time per decision step (x-axis). These results were obtained by varying the number of simulations allowed in each algorithm. The performance is evaluated based on the mean of the values (y-axis) in multiple runs. As we can see from the figure, OPAS converged much faster than POMCP. Moreover, it achieved much larger value than POMCP given the same amount of time per step. In our tests, we observed that POMCP got stuck in a local optima with the value of about 50. Note that the runtime of both OPAS and POMCP grows linearly with the number of simulations.

For reasons of presentation, the standard deviations for POMCP are omitted in Figure 4(a) because they are large. This is not surprising because the FR guided by POMCP was killed by the radioactive cloud in the runs of some challenging problem instances. In such cases, the runs terminated with a penalty of -100. In contrast, the performance of OPAS was very stable (with the standard deviation about 6) even for those challenging problem instances. Note that we tested both methods with the same set of problem instances. We observed that the FR guided by OPAS kept alive across all runs in the experiments, which is crucial for disaster response tasks. As in Algorithm 1, we use a conservative strategy where the FR's policy is computed with the information currently available.

Figures 4(b) and 4(c) illustrate the performance of OPAS and POMCP in runtime and value respectively with different numbers of UAVs. As shown in Figure 4(b), the runtime of both algorithms grows linearly with the number of UAVs and OPAS ran faster than POMCP with the same number of UAVs. This confirms the scalability of OPAS for problems with many UAVs. It is interesting to see from Figure 4(c) that the values of both algorithms decreased with more UAVs. This indicates that there may be redundancy in the information collected by the UAVs. With more UAVs, there are more costs to run them but the benefit in the sense of information value is limited. Generally, the best number of UAVs depends on the specific problem at hand. In the problem instances we chose, 2 UAVs seem sufficient for collecting the information about the radioactive cloud. It is worth noting that, with different numbers of UAVs, OPAS produced better values than POMCP though they shared similar trend. With more UAVs, the standard deviation of OPAS grows because there is more diversity over the runs in the choice of the UAVs' actions with larger action space. Again, the standard deviations of POMCP were omitted for reasons of presentation.

7 Conclusion

In this paper, we proposed a novel approach that integrated online planning with action sensing for guiding an FR, assisted by UAVs in disaster response. In more detail, we modeled the problem as a POMDP and proposed a novel algorithm to efficiently solve this model. In particular, we presented a novel model based on POMDPs for planning and sensing in disaster response capturing both uncertainty and partial observability of that domain. To solve the model, we proposed an online planning algorithm that is *anytime* and *scales well* based on *Monte-Carlo simulation* and the concept of *value of information*. Results from our experiments confirmed the scalability and advantage of our algorithm in the benchmark domain comparing to the leading POMDP solver (i.e., POMCP). Future work will look at conducting field-trials of our approach using human FRs and physical UAVs.

Acknowledgments

We thank all the anonymous reviewers for their constructive comments. This work is supported in part by the Fundamental Research Funds for the Central Universities (No. WK0110000045) and the Youth Innovation Promotion Association, CAS (No. CX0110000012).

References

[Adams and Friedland, 2011] Stuart M Adams and Carol J Friedland. A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management. In *Workshop on Remote Sensing for Disaster Response*, 2011.

[Bai *et al.*, 2012a] Aijun Bai, Feng Wu, and Xiaoping Chen. Towards a principled solution to simulated robot soccer. In *Proc. of RoboCup*, pages 141–153, 2012.

[Bai *et al.*, 2012b] Haoyu Bai, David Hsu, Mykel J Kochenderfer, and Wee Sun Lee. Unmanned aircraft collision avoidance using continuous-state POMDPs. *Robotics: Science and Systems VII*, 1, 2012.

[Bernardini *et al.*, 2015] Sara Bernardini, Maria Fox, and Derek Long. Combining temporal planning with probabilistic reasoning for autonomous surveillance missions. *Autonomous Robots*, pages 1–23, 2015.

[Chanel *et al.*, 2013] Caroline Ponzoni Carvalho Chanel, Florent Teichteil-Königsbuch, and Charles Lesire. Multi-target detection and recognition by UAVs using online POMDPs. In *Proc. of AAAI*, 2013.

[Fiedrich and Burghardt, 2007] Frank Fiedrich and Paul Burghardt. Agent-based systems for disaster management. *Communications of the ACM*, 50(3):41–42, 2007.

[Geffner and Bonet, 1998] Hector Geffner and Blai Bonet. Solving large POMDPs using real time dynamic programming. In *Proc. of AAAI Fall Symp. on POMDPs*, 1998.

[Geyer, 2008] Christopher Geyer. Active target search from UAVs in urban environments. In *Proc. of ICRA*, pages 2366–2371, 2008.

[Howard, 1966] Ronald Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26, 1966.

[Jennings *et al.*, 2014] Nicholas R Jennings, Luc Moreau, David Nicholson, Sarvapali Ramchurn, Stephen Roberts, Tom Rodden, and Alex Rogers. Human-agent collectives. *Communications of the ACM*, 57(12):80–88, 2014.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[Musliner *et al.*, 2006] David J Musliner, Edmund H Durfee, Jianhui Wu, Dmitri A Dolgov, Robert P Goldman, and Mark S Boddy. Coordinated plan management using multiagent MDPs. In *AAAI spring symposium: Distributed plan and schedule management*, pages 73–80, 2006.

[Ong *et al.*, 2010] Sylvie CW Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under uncertainty for robotic tasks with mixed observability. *International Journal of Robotics Research*, 29(8):1053–1068, 2010.

[Ragi and Chong, 2012] Shankarachary Ragi and Edwin KP Chong. Dynamic UAV path planning for multitarget tracking. In *Proc. of ACC*, pages 3845–3850, 2012.

[Ramchurn *et al.*, 2015a] Sarvapali Ramchurn, Trung Dong Huynh, Yuki Ikuno, Jack Flann, Feng Wu, Luc Moreau, Nicholas R Jennings, Joel E Fischer, Wenchao Jiang, Tom Rodden, et al. HAC-ER: a disaster response system based on human-agent collectives. In *Proc. of AAMAS*, pages 533–541, 2015.

[Ramchurn *et al.*, 2015b] Sarvapali D Ramchurn, Joel E Fischer, Yuki Ikuno, Feng Wu, Jack Flann, and Antony Waldock. A study of human-agent collaboration for multi-UAV task allocation in dynamic environments. In *Proc. of IJCAI*, pages 1184–1192, 2015.

[Ramchurn *et al.*, 2015c] Sarvapali D Ramchurn, Feng Wu, Wenchao Jiang, Joel E Fischer, Steve Reece, Stephen Roberts, Tom Rodden, Chris Greenhalgh, and Nicholas R Jennings. Human-agent collaboration for disaster response. *Autonomous Agents and Multi-Agent Systems*, pages 1–30, 2015.

[Silver and Veness, 2010] David Silver and Joel Veness. Monte-Carlo planning in large POMDPs. In *Proc. of NIPS*, pages 2164–2172, 2010.

[Somani *et al.*, 2013] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. DESPOT: Online POMDP planning with regularization. In *Proc. of NIPS*, 2013.

[Thrun, 1999] Sebastian Thrun. Monte carlo POMDPs. In *Proc. of NIPS*, volume 12, pages 1064–1070, 1999.

[Waharte and Trigoni, 2010] Sonia Waharte and Niki Trigoni. Supporting search and rescue operations with UAVs. In *Proc. of EST*, pages 142–147, 2010.

[Wu and Jennings, 2014] Feng Wu and Nicholas R. Jennings. Regret-based multi-agent coordination with uncertain task rewards. In *Proc. of AAAI*, 2014.

[Wu *et al.*, 2010] Feng Wu, Shlomo Zilberstein, and Xiaoping Chen. Trial-based dynamic programming for multi-agent planning. In *Proc. of AAAI*, pages 908–914, 2010.

[Wu *et al.*, 2015a] Feng Wu, Sarvapali D. Ramchurn, Wenchao Jiang, Jeol E. Fischer, Tom Rodden, and Nicholas R. Jennings. Agile planning for real-world disaster response. In *Proc. of IJCAI*, pages 132–138, 2015.

[Wu *et al.*, 2015b] Kegui Wu, Wee Sun Lee, and David Hsu. POMDP to the rescue: Boosting performance for RoboCup rescue. In *Proc. of IROS*, 2015.

[Xu *et al.*, 2005] Yang Xu, Paul Scerri, Bin Yu, Steven Okamoto, Michael Lewis, and Katia Sycara. An integrated token-based algorithm for scalable coordination. In *Proc. of AAMAS*, pages 407–414, 2005.