To Ask, Sense, or Share: Ad Hoc Information Gathering

Adam Eck and Leen-Kiat Soh
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, Nebraska, USA
{aeck, lksoh}@cse.unl.edu

ABSTRACT

Agents operating in complex (e.g., dynamic, uncertain, partially observable) environments must gather information from various sources to inform their incomplete knowledge. Two popular types of sources include: (1) directly sensing the environment using the agent's sensors, and (2) sharing information between networked agents occupying the same environment. In this paper, we address agent reasoning for appropriately selecting between such types of sources to update agent knowledge over time. In particular, we consider ad hoc environments where agents cannot collaborate in advance to predetermine joint solutions for when to share vs. when to sense. Instead, we propose a solution where agents individually learn the benefits of relying on each type of source to maximize knowledge improvement. We empirically evaluate our learning-based solution in different environment configurations to demonstrate its advantages over other strategies.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *intelligent agents, multiagent systems*

General Terms

Performance, Design, Experimentation

Keywords

Information Gathering; Information Sharing; Ad Hoc

1. INTRODUCTION

One of the most fundamental responsibilities of intelligent agents is *understanding their complex (e.g., dynamic, uncertain, partially observable) environments*, which guides agent reasoning, actuation, and goal accomplishment. Often, agents lack complete knowledge of their environment *a priori* and must update their understanding over time. These updates are informed by incorporating information gathered whilst operating in the environment. Two popular types of sources of information include (1) an agent independently *sensing* its environment, gathering direct observations as a result of the agent's actions and sensors, and (2) receiving *shared* information from other agents operating in the same environment (either cooperatively for the sake of the system or for individual profit by self-interested agents).

Depending on the application, these two types of sources might have different benefits (e.g., types of information provided, information quantity and quality) and costs (e.g., resource and time expenses). Sensing can be performed on demand, gathering information as soon as the agent needs, and the agent can do so in a timely fashion without taking away from other agents' activities. Information sharing, on the other hand, can propagate information through the entire system potentially faster and with less cost (not waiting for each agent to individually sense the same information). However, relying on sharing also means waiting for another agent to possess the desired information, and sharing takes time and resources away from other agent activities that could instead further the sharing agent's individual goals.

Because of these differences, agents in applications where both sources coexist face an interesting question: *when should I use sensing to update my understanding vs. when should I request information from other agents and rely on shared information?* Answering this question leads to a challenging tradeoff between using the two types of information sources that when properly balanced could lead to improved agent behavior and goal accomplishment (e.g., through lower cumulative cost and higher quality knowledge).

Traditionally, agents in a shared environment would pre-coordinate when they might be willing and able to share information so that each agent could plan appropriately to know when to sense vs. when to rely on shared information. However, in many applications, this pre-coordination might not be possible. Specifically, in **ad hoc environments** where pre-coordination is impossible and agents might not know the behaviors or capabilities of their peers in advance [24], agents cannot determine *a priori* the value of relying on shared information against the value of sensing alone. This is especially true in many types of ad hoc environments that are also *open environments*, where agents can join and leave the environment over time. Agent openness is especially problematic to information sharing because the availability of shared information changes over time and knowledge about the environment disappears with departing agents (who knew more than newly joining agents). Thus, determining when to sense vs. when to rely on shared information is especially difficult in ad hoc environments. In this paper, we study how agents should balance the sensing vs. sharing tradeoff in ad hoc environments, henceforth referred to as the ad hoc information gathering (AHIG) problem.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

In order to solve the AHIG, we propose a learning-based solution where agents individually learn over time how different types of information gathering actions (independently sensing vs. requesting shared information) improve their knowledge about the environment. Through learning, agents can find good information gathering strategies without relying on pre-coordination in ad hoc environments, instead treating other agents as part of the environment affecting the quality of their information gathering. Moreover, learning enables each agent to adapt its behavior as it interacts with different agents, which is valuable in open environments where agents join and leave over time. Thus, through learning, agents can individually adapt their behavior to maximize their own knowledge improvement by learning the benefits of using different types of information sources without requiring coordination between agents.

However, because agents are operating in complex environments with incomplete information, learning is generally a computationally complex problem: learning in partially observable environments is much harder than learning in fully observable environments. To simplify the agents' learning process, we show how the agents' general problem of *understanding the current state of the complex environment* can be transformed to a simpler problem of *improving agent knowledge over time*, in a transformation we term the **Knowledge State MDP** exploiting full observability of current measures of agent knowledge as intermediate states for guiding agent decision making. As a result, an agent can learn faster how to gather information in the environment to best refine knowledge. Moreover, this transformation is potentially useful in more general information gathering problems (beyond the AHIG).

To demonstrate the effectiveness of our transformation and learning-based solution, we empirically evaluate using different experimental environment configurations how well agents learn to select between different information sources over time to improve their knowledge. We discover that our solution outperforms baseline approaches maximizing either sensed or shared information, and does so by appropriately selecting between different information sources at different times to best refine agent knowledge. Furthermore, our results indicate that learning about how to gather information is most beneficial when information is most scarce (and careful information gathering is most necessary).

2. PROBLEM

The AHIG problem occurs whenever a set of agents observe the same environment and can share information but cannot coordinate in advance to determine when agents might share or what quality of information they might provide. This includes real world examples such as (1) intelligent ad hoc sensor networks, where agents are deployed on wireless sensors that are randomly dropped to monitor an open space, (2) robotic search and rescue operations, where different organizations might bring their own robots to explore the same disaster area, and (3) ad hoc traffic information networks, where intelligent agents on cars communicate with a road infrastructure system as they navigate through town to report and understand traffic conditions.

2.1 AHIG Formulation

We formalize the AHIG problem as follows. A set of agents $Ag = \{i\}$ exist in a shared environment and are connected by a bidirectional communication network. Because communication costs grow as the network becomes larger, each agent's local neighborhood $N(i)$ is relatively small compared to the size of the entire network. Occasionally, due to openness, some agents will leave the network and others will join. Thus, we represent the current set of agents at time t with Ag_t, and likewise for an agent i's neighborhood $N_t(i)$.

Also in the shared environment are a finite set of phenomena $P = \{j\}$ that represent objects, entities, or properties of the environment that the agents need to understand. Each phenomenon j can take states from a finite set $PS_j = \{ps\}$, and the current state of each phenomenon in the dynamic environment changes with probability cp each time step. In AHIG, the agents are tasked with always understanding the current state of each phenomenon, which requires forming correct knowledge about each phenomenon over time that is refined through gathering information.

To gather information about a particular phenomenon, agents can perform different actions that use different types of sources for information. In particular, each agent can (1) *sense* each phenomenon directly using its sensors, or the agent can (2) *request* that its neighbors $N_t(i)$ share their beliefs about a phenomenon. We assume that the agent's sensors are noisy and imperfect, returning correct observations about the sensed phenomenon's current state with accuracy acc (and an incorrect observation with probability $1 - acc$). Agents can also perform a third type of action: (3) agents can respond to requests from neighbors with a *share* action communicating the agent's uncertain current knowledge about the state of the phenomenon in question.

The goal of each agent is to form accurate knowledge about each phenomenon, representing good knowledge about the current state of the environment, while minimizing costs incurred in sensing. Agents are awarded a reward for each time point during which they have relatively certain knowledge about a phenomenon, whereas sensing actions and requests for information incur costs to the agent. To encourage self-interested agents to collaborate, the agents are also awarded a small reward for sharing information with their neighbors, but only when requested (to avoid unnecessarily consuming the communication resources) and when they are confident about the current state of the requested phenomenon (to avoid sharing unfruitful information). Otherwise, agents receive a penalty for sharing information.

To illustrate, consider a search and rescue (S&R) robotics example, where robot agents Ag_t explore a damaged building after a natural disaster. Here, the phenomena P represent different locations where victims might be trapped, and the phenomenon states PS_j indicate whether victims exist at location j. A robot i can either directly observe the environment with a noisy camera sensor (that consumes limited energy), or the agent can communicate with nearby robots $N_t(i)$ using line-of-sight communications. The goal of each robot is to determine with certainty whether victims exist in each location so that they can be rescued by human first responders, all-the-while minimizing energy and time costs.

Of final note: how agents represent their knowledge about the phenomena in the environment, as well as how they choose actions to refine their knowledge are not specified in the general AHIG formalization. Different domains, applications, and solutions might require different approaches to these features (knowledge and decision making) that are internal to the agent. Indeed, in real-world ad hoc envi-

ronments, different agents produced by different developers might even use different approaches to these features in the same environment. However, agents must have some shared language that is consistent between agents for communicating shared information. In this paper, we choose the knowledge representation and decision making process as part of our solution, described in Section 4.

2.2 Related Work

The AHIG problem is closely related to several other problems in the multiagent systems literature. First, the Large Team Information Sharing (LTIS) problem (e.g., [11, 12, 19]) also considers a team of agents working together to observe at least one phenomenon in the environment, where agents both sense the current state of the phenomenon individually, as well as share information through the team's network. Prior research on LTIS has focused primarily on producing analytic models for the flow of information through the team of agents [11, 12], as well as developing distributed solutions for adapting information flow to achieve accurate, consistent, shared beliefs [11, 19]. However, LTIS differs from the AHIG in several key ways. First, in LTIS, the team of agents is constant over time (i.e., there is no agent openness), and agents follow a pre-coordinated strategy of when to share information. Second and most importantly, in LTIS agents do not choose between sensing, requesting, or sharing information. Instead, agents with sensors (which might not be all agents in the team) always receive observations from their sensors at every time point. Additionally, agents never request information; instead, they automatically share information with their neighbors whenever (and only when) they reach new highly certain knowledge about a phenomenon. Thus, LTIS does not consider the tradeoff between relying on different types of information as in the AHIG.

Another closely related problem studied in the multiagent systems literature is trust and reputation systems (e.g., [21, 22, 26]). In such systems, agents can also request and share information with one another to provide additional information to refine agent knowledge over time. The primary focus in trust and reputation systems is to determine how to incorporate such shared information: should the sharing agent be highly trusted and should their information heavily influence the receiving agent's knowledge, or should an agent be cautious when receiving information from another agent with which it has limited experience interacting? Like LTIS, this research does not focus on balancing information from other agents with the agent's own sensing, and thus does not solve the AHIG problem, but it is complementary in that reasoning about the trustworthiness and reputation of neighboring agents as information sources could be used to improve an AHIG agent's decision making process (which we intend to pursue as future work).

Finally, previous research in ad hoc environments has focused on problems such as how to lead teams of agents without communication [1, 10], as well as how to learn to interact with a single Markovian agent [6]. Since information sharing inherently requires communication, our research differs from the former (although in our work, agents still cannot pre-coordinate how they will interact, under the broad definition of ad hoc environments [24]). Similar to the latter, we also use reinforcement learning to determine how to interact with other agents, although our approach considers an agent working with multiple other agents in the environment.

3. POMDP FORMULATION

In order to solve the AHIG and gather the necessary information to understand the environment, each agent faces a sequential decision making problem of planning a sequence of actions to perform that refine its incomplete knowledge while minimizing costs incurred for gathering such information. In most partially observable environments (which includes AHIG since sensing phenomena returns noisy, imperfect observations), sequential decision making problems are generally solved by some variant of partially observable Markov decision processes (POMDPs) [15]. This is especially true of applications of single agent control of environment monitoring (e.g., [2, 4, 7, 8, 23], similar to our S&R robot example), which we extend in this paper to multiagent information gathering in ad hoc environments.

To setup our solution, in the following we first introduce background on the MDP and POMDP frameworks, then we provide a description of both how the AHIG problem could be cast as a POMDP and the problems with this formulation. Then, in Section 4, we will introduce our Knowledge State MDP transformation of the POMDP for sequential decision making for information gathering problems.

3.1 Background

Markov decision processes (both fully and partially observable) are mathematical descriptions of complex environment properties (e.g., dynamics, incomplete information, and uncertainty) that enable an agent to plan action sequences that maximize rewards (and/or minimize costs) in order to accomplish its goals [15].

Formally, a (discounted, finite state) MDP is represented by a tuple $\langle S, A, T, R, \gamma \rangle$ where $S = \{s\}$ is a set of states in which the agent needs to choose actions from the set $A = \{a\}$. Because the environment is dynamic, actions can change the current state according to a stochastic transition function $T : S \times A \times S \rightarrow [0, 1]$ modeling the probability $T(s_t, a, s_{t+1}) = P(s_{t+1}|s_t, a)$ that the state changes from s_t to s_{t+1} after action a. To guide agent decision making, the agent receives rewards (or costs) according to a reward function $R : S \times A \rightarrow \mathbb{R}$ dependent on the current state and the action chosen. The agent's goal is to determine a plan of actions called a policy $\pi : S \rightarrow A$ that tells the agent what actions to perform (dependent on state) to maximize cumulative, discounted rewards, with $\gamma \in (0, 1)$ a discount factor weighting uncertain future rewards:

$$\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \tag{1}$$

Because the current state is hidden in partially observable environments, a POMDP model is often used instead. Formally, a (discounted, finite state) POMDP is a tuple $\langle S, A, Z, T, O, R, \gamma, b_0 \rangle$ with S, A, T, R, γ as in a MDP. $Z = \{z\}$ represents a set of observations the agent receives (instead of perfect information about s), which are generated according to a stochastic observation function $O : S \times A \times Z \rightarrow [0, 1]$ modeling the probability $O(s_{t+1}, a, z) = P(z|s_{t+1}, a)$ that the environment produces observation z after action a transitions the state to s_{t+1}. To handle uncertainty in the current state s_t, the agent maintains a belief state $b : S \rightarrow [0, 1]$ representing its probabilistic knowledge in the current state. Using observations gathered from the environment, the agent updates its knowledge through belief updates:

$$b_{t+1}(s_{t+1}) = \frac{1}{\eta} O(s_{t+1}, a, z) \sum_{s_t \in S} T(s_t, a, s_{t+1}) b_t(s_t) \quad (2)$$

where η normalizes b_{t+1}. b_0 is the agent's initial belief state. Note, R can be chosen to provide rewards based on properties of the belief state b, instead of just individual environment states [2]. Likewise, policy π maps beliefs to actions.

3.2 AHIG as a POMDP

Since the AHIG is a sequential decision making problem in a partially observable environment (i.e., phenomenon states are partially observable), casting the AHIG as a POMDP is a natural starting point for a potential solution. In particular, we consider the POMDP formulation for the AHIG $\langle S, A, T, O, R, \gamma, b_0 \rangle$ summarized in Table 1.

In this POMDP, the state space S contains variables representing different information about situations faced by the agent: partially observable PS_j represent the different states each phenomenon can take (e.g., the presence of victims in different locations in our S&R example), and fully observable S_{Req} and S_{Rec} represent counts per phenomenon of how long it has been since the agent last requested that its neighbors share information or received a neighbor's request, respectively. These count variables are useful for tracking (1) whether the agent recently requested information, so that it doesn't request too frequently and disrupt other agents, and (2) whether a neighbor requested information so that the agent knows if it is appropriate to share its own knowledge. Given this S, the belief state b represents the agent's uncertain knowledge about each phenomenon's hidden state. This knowledge is refined using information Z collected from actions A using Eq. 2. Beliefs start with pure uncertainty (a uniform distribution over phenomenon states, e.g., a location is equally likely to contain a victim or not).

Since the environment is dynamic, the transition function T encodes the probability that phenomena change states at each time point (to a new state with probability cp, else phenomenon states stay the same with probability $1 - cp$, c.f., Section 2.1) (e.g., whether a previously safe location collapses and traps new victims, or trapped victims are rescued). The fully observable states transition deterministically each time step: the count for each phenomenon j in S_{Req} is incremented by one (up to k) unless the agent requests new information about j, and the count for each phenomenon j in S_{Rec} is incremented by one (up to k) unless the agent shares information (in which case it reverts to k to indicate no request from a neighbor is pending).

The observation function O, on the other hand, encodes the probability that the agent receives information about a particular phenomenon depending on the action taken. For $Sense_j$ actions, O encodes that the agent observes the correct state with probability acc (the agents' sensor's accuracy, c.f., Section 2.1) and a wrong state with probability $1 - acc$ (e.g., whether or not the robot's camera correctly identifies a victim in a room). Other actions return a null observation since they do not directly gather information about the state of any phenomenon in the environment.

The reward function R encodes (1) the rewards for having high certainty beliefs or sharing information when requested, and (2) the costs for information gathering actions or penalties for sharing unrequested or uncertain information as described in Section 2.1. Maximizing cumulative rewards leads the agent to highly certain knowledge (for which it receives a reward) while minimizing costs used to refine its knowledge.

However, a few problems exist in this solution formulation. First, the observation set Z only considers observations from the $Sense_j$ actions and does not handle shared information from neighbors, which would occur some delayed amount of time after a $Request_j$ action. Although Z could be modified to include additional variables for received information, this limits the types of shared information neighbors can provide to discrete quantifications of the neighbor's beliefs (e.g., the locations most likely to contain victims), which loses information about the neighbor's uncertainty (e.g., the probabilities of victims in each location). Otherwise, the observation space would be continuous (and thus very difficult to work with) if neighbors shared their full belief states.

Second, even if Z were extended to include shared information, there is no way for the observation function in a single agent POMDP to encode a probability that a neighbor shares a phenomenon state from its own beliefs (in response to a request) without some pre-coordination and agreement between agents. That is, agents must understand the likelihoods that a neighbor both (1) shares a particular piece of information (dependent on the neighbor's beliefs that change over time) and (2) any information at all (e.g., a robot might be busy and unwilling to share information at the current time). Without this information, an agent cannot calculate the overall probability that it receives any particular information from a neighbor at any point in time, necessary for updating its beliefs with Eq. 2 with shared information, nor plan what information it might receive over time. Therefore, a single agent POMDP formulation of the AHIG will not directly work in ad hoc environments.

Of note, traditional multiagent variants of POMDPs (e.g., DEC-POMDPs, Distributed POMDPs, and I-POMDPs [3, 13, 18]) provide some methods for handling both of the aforementioned problems; however, these types of POMDPs require pre-coordination so are inappropriate for ad hoc environments and do not scale well with the number of agents.

To resolve these problems inherent in a POMDP-based AHIG model, we need to add some method to incorporate shared information (which is inherently multiagent in nature) outside of the (single agent) POMDP framework's belief updates. Then, the agent should still make decisions based on its current knowledge, but it also needs a way to plan how its beliefs will change to form an action policy.

4. KNOWLEDGE STATE MDP

In this section, we first describe how we propose to incorporate shared information from other agents, building on the aforementioned POMDP formulation. Then, we describe a *transformation* of the POMDP into a MDP that looks at solving the AHIG from a metareasoning perspective, decoupled from how the agent refines its knowledge when it receives new information. Finally, we introduce a *learning process* for the MDP that enables an agent to learn how to choose actions to take to refine its knowledge in ad hoc environments without requiring pre-coordination about how and when other agents will share information.

4.1 Incorporating Shared Information

For agent knowledge about phenomenon states, we consider probability distributions over all possible phenomenon states very similar to belief states described in Section 3.2.

Table 1: POMDP Formulation of the AHIG Problem

POMDP	Values	AHIG Description
State Variables S	$S_{Req} \times S_{Rec} \times_{j \in P} PS_j$ $S_{Req} = \{0, \ldots, k\}^{\lvert P \rvert}$ $S_{Rec} = \{0, \ldots, k\}^{\lvert P \rvert}$	Counts of the number of time steps since the agent last requested (S_{Req}) or received a request for information (S_{Rec}), up to a maximum count k, and the phenomenon states (PS_j)
Actions A	$\bigcup_{j \in P} \{Sense_j, Request_j, Share_j\}$	Actions (1) sensing a particular phenomenon j, (2) requesting information from neighbors about phenomenon j, and (3) sharing information to neighbors about phenomenon j (for each $j \in P$)
Observations Z	$\{null\} \cup PS_j$	Observations about the phenomenon state of a particular phenomenon, or receiving no observation at all
Transition Function T	$[0, 1]$	Probability that request counts change (deterministically) and phenomenon states change (stochastically) after each action
Observation Function O	$[0, 1]$	Probability that the agent receives observations about partially observable phenomenon states from its actions
Reward Function R	$\mathbb{R}$	Rewards received for taking different actions based on the current state of the environment and the agent's knowledge.
Discount Factor γ	$(0, 1)$	A discount factor to use for weighting future, uncertain rewards
Initial Belief State b_0	$[0, 1]$	The probability ascribed to each phenomenon state being the correct initial state of each phenomenon (a uniform distribution)

We reuse notation with $b_t(j, ps)$ the probability that the agent believes phenomenon $j \in P$ is currently $ps \in PS_j$. For $Sense_j$ actions, beliefs update from b to b' after receiving observation z about phenomenon j using Bayes' rule:

$$b'(j, ps) = \frac{p}{\eta}\Big[(1-cp)b_t(j, ps) + \sum_{ps' \in PS_j / ps} \Big(\frac{cp}{|PS_j| - 1}\Big) b(j, ps')\Big] \tag{3}$$

where $p = acc$ when $z = ps$, else $p = 1 - acc$. This is equivalent to the belief updates performed with Eq. 2 using the POMDP formulation described in Section 3.2.

With respect to shared information, we assume[1] that agents share the full information about their beliefs: the probabilities ascribed to each phenomenon state for the particular phenomenon for which a neighbor sent a request. Then, the corresponding belief update for shared information b_{Sh} is:

$$b'(j, ps) = \frac{b(j, ps)\big[w \cdot b_{Sh}(j, ps) + (1 - w)(1 - b_{Sh}(j, ps))\big]}{\sum_{ps' \in PS_j} b(j, ps')\big[w \cdot b_{Sh}(j, ps') + (1 - w)(1 - b_{Sh}(j, ps'))\big]} \tag{4}$$

where constant weight[2] w dampens shared information so that uncertain shared beliefs do not cause agents to become certain too quickly from little gathered information.

Using these two rules, agents can incorporate information from both from (1) directly observing a phenomenon with its sensors, and (2) its neighbors sharing their knowledge.

4.2 Knowledge State MDP Transformation

At the core of AHIG, the agent's behavior does not necessarily depend on which *particular* phenomenon state is currently correct for each phenomenon, but instead the problem is really about how the agent should choose actions to improve its *knowledge* (noting that actions to improve knowledge could be equivalent for each actual phenomenon state). After all, the agents' goal is to form highly certain knowledge about each phenomenon using the information available in the environment. For instance, in our S&R example, a robot will base its information gathering on *how certain its knowledge is* about a location (looking to resolve its uncertainty so that it knows where all victims are as quickly as possible), which is *internal* to the agent and independent of whether or not an *external* unknown location actually contains victims. The robot isn't necessarily responsible for using the refined knowledge for a separate task (that is done by human first responders), but the goal of the agent in the AHIG is to develop high quality knowledge that could subsequently be used for other purposes, depending on the application.

Given this insight, we transform the above POMDP into what we call the **Knowledge State MDP**–an alternative formulation of the problem directly enabling an agent to make decisions of how to gather information based on considering the current state of its *knowledge*, as opposed to the state of the *environment* (including the states of phenomena under observation). This provides a ***metareasoning solution*** enabling the agent to choose how to gather information based on reflecting about the quality of its knowledge without worrying about the domain-specific contents of that knowledge. As a result, the agent's decision making (at a metareasoning level) is decoupled from its knowledge refinement (at a standard reasoning level), as desired.

The Knowledge State MDP can be mathematically described as a MDP $\langle S_{Req} \times S_{Rec} \times K, A, T, R \rangle$, summarized in Table 2. Here, the partially observable part of the state space is replaced with the different knowledge states K of the agent's knowledge (which are fully observable when reflecting on the agent's knowledge) as it gathers information to understand its environment. K is combined with the S_{Req} and S_{Rec} state variables representing counts of time since requests were sent or received, described in Section 3.2.

Recall that in the AHIG, the primary concern of the agent is to form highly certain beliefs, so the state of agent knowledge should reflect how much certainty exists in the agent's knowledge. Then, the agent can take actions that improve its certainty and result in better knowledge states (closer

[1]Other types of information might instead be shared, based on the domain, which we leave to consider as future work.

[2]Such weights are common in the information sharing literature (e.g., [11, 19]) and could be learned as in trust and reputation systems to further refine our solution, which we intend to explore in the future. Please see [11] for a more elaborate discussion of the impact of weight w.

Table 2: Knowledge State MDP Formulation

MDP	Values	AHIG Description
State Variables S	$S_{Req} \times S_{Rec} \times K$ $S_{Req} = \{0, \ldots, k\}^{\lvert P \rvert}$ $S_{Rec} = \{0, \ldots, k\}^{\lvert P \rvert}$ $K : H(b, j)$ in k bins	Counts of the number of time steps since the agent last requested information (S_{Req}) or received a request for information (S_{Rec}), up to a maximum count k, and the agents current certainty (K) in the current state of each phenomenon j
Actions A	$\bigcup_{j \in P} \{Sense_j, Request_j, Share_j\}$	Actions (1) sensing a particular phenomenon j, (2) requesting information from neighbors about phenomenon j, and (3) sharing information to neighbors about phenomenon j (for each $j \in P$)
Transition Function T	$T_{S_{Rec}, S_{Req}} \cdot T_K \in [0, 1]$	Probability of state changes, as the product of request state variable transitions and knowledge state transitions T_K
Reward Function R	$\mathbb{R}$	Rewards received for taking different actions based on the agent's knowledge.
Discount Factor γ	$(0, 1)$	A discount factor to use for weighting future, uncertain rewards

to full certainty). Given that the knowledge representation b described in Section 4.1 is a probability distribution over possible phenomenon states for each phenomenon, an appropriate measure of certainty in each phenomenon j's state (independent of application) is the entropy $H(b, j) \in [0, 1]$ of the probability distribution representing its knowledge [2]:

$$H(b, j) = 1 + \frac{1}{\log |PS_j|} \sum_{ps \in PS_j} b(j, ps) \log b(j, ps) \qquad (5)$$

To create a set of finite knowledge states K using $H(b, j)$ so that the MDP is a discrete state MDP, and thus is much more tractable, we suggest discretizing the certainty values into equal sized bins so that there exist a desired number of states $|K|$. Note that a larger $|K|$ creates a finer grained separation between different knowledge states, potentially enabling better planning, whereas a smaller $|K|$ make the MDP faster to solve (and has implications on the learning process described in Section 4.3).

Given the rewards in the AHIG described in Section 2.1, it is important to note that the same reward encoding works for the Knowledge State MDP as well: knowledge states identifying high certainty earn a reward, and action-based costs, rewards, and penalties stay the same.

4.3 Learning Knowledge State Dynamics

Now, within the Knowledge State MDP, the key to guiding appropriate action selection is the dynamics of how knowledge states change based on each action $a \in A$. That is, how actions lead the agent to improve its certainty over time. This information is encoded in the knowledge state transition function T_K. Unfortunately, due to a lack of pre-coordination to determine how and when agents will share information, this function is undetermined initially. However, whereas this was a problem in our suggested POMDP-based solution in Section 3.2, the transformation into an MDP makes it feasible to perform model-based reinforcement learning[3] (MB-RL) [16] to learn this transition function through interactions with the environment and other agents (and adjust it over time as agent openness causes the environment to change), instead of having to rely on pre-coordination.

[3]Although MB-RL algorithms also exist for POMDPs (e.g., [20]), such algorithms have high complexity and are not generally applicable in practice for POMDPs of moderate to large state spaces (which grows quickly with phenomena P and their states PS_j for Section 3.2's POMDP).

In general, any MB-RL algorithm should be sufficient to learn the knowledge state transition function T_K. For our experimental setup in this paper, we use a learning approach for the transition function similar to recent variants [14, 25] of one of the most popular MB-RL algorithms: R-max [5]. In particular, this algorithm uses frequentist counting by maintaining a table counting the number of times $n(s_t, a, s_{t+1})$ that the agent observes a transition from state s_t to s_{t+1} after taking action a, then the algorithm updates the transition table to

$$T(s_t, a, s_{t+1}) = \frac{n(s_t, a, s_{t+1})}{n(s_t, a)} \qquad (6)$$

whenever the total count of observed transitions for a state-action pair $n(s_t, a) = \sum_{s_{t+1} \in S} n(s_t, a, s_{t+1})$ equals a parameter m, after which the learning counts for the state-action pair are reset to 0. A smaller m enables *faster updates* to the transition table, whereas a larger m ensures *more precise updates* (by relying on more observed transitions before updating). Of note, smaller $|K|$ are also beneficial here, causing the same knowledge state to be encountered more frequently, and thus more frequent learning updates.

Considering the Knowledge State MDP, learning T_K amounts to learning exactly how the certainty in the agent's knowledge changes based on (1) each information gathering action, and (2) how long it has been since the agent requested information (since this alerts the agent both how timely neighbors respond, as well as whether or not they respond at all). Understanding such changes to agent knowledge is exactly the information the agent needs to determine which information gathering actions to perform in order to reach highly certain knowledge and achieve its primary goal–actions that are more likely to lead to high certainty knowledge states from the current knowledge state are actions that most improve the agent's knowledge, as desired.

This learning process only requires feedback from the agent's knowledge updates (using sensed or shared information) to observe exactly which knowledge state (i.e., certainty) transitions occur after taking each action. Thus, the agent can learn over time how its knowledge changes when it senses, as well as when it requests shared information (including how long such information takes to arrive), without having to know in advance when or how other agents will choose to share information. Therefore, this learning process bypasses the problems of other solutions in ad hoc environments without requiring pre-coordination to understand the behaviors

of neighboring agents and their impact on knowledge refinement. Moreover, the agent also adapts its understanding of knowledge state transition changes over time, which is important for open environments where information sharing can become more or less prevalent over time, in which case a smaller m might be useful for more frequent learning and faster adaptation to the changing environment.

By planning with the reward function R, the agent plans to reach certainty as fast as possible (by maximizing rewards for certain knowledge) while also minimizing costs required for gathering information, making the agent both effective and efficient at its task. Thus, our Knowledge State MDP coupled with MB-RL is an appropriate solution for the AHIG.

It is important to note that this Knowledge State MDP transformation is closely related to a similar metareasoning framework in the literature: the Observer Effect POMDP [8], which combines fully observable knowledge states with partially observable environment states to guide agents to perform actions that refine knowledge over time. Our solution differs in that (1) it learns the transitions in knowledge over time, as opposed to the domain-specific value of information, and (2) extends this type of approach to a multiagent setting where learning enables the agent to reason about the affects of other agents on its own knowledge.

5. EXPERIMENTAL SETUP

To better understand our approach and investigate its performance in different AHIG settings, we conducted experiments empirically evaluating how well our Knowledge State MDP and MB-RL process guide agent information gathering using different information sources, including information sharing, without requiring pre-coordination. In particular, we considered a range of network configurations that might reflect different types of environments and applications.

That is, we varied the average neighborhood size $N_t(i)$, where larger neighborhoods made shared information more prevalent, whereas smaller neighborhoods represent more communication-constrained environments (e.g., our S&R robot example where only a few robots might be within line-of-sight of one another). The networks were randomly generated using an Erdos-Renyi model [9]. Since the environment was ad hoc, agents knew nothing about their neighbors in advance. Moreover, we made the environment open, where a predetermined percentage (10%) of the agents left periodically (every 100 time steps) and new agents joined. This agent openness also reduced the availability of information over time, making information sharing more or less valuable at different points in time. Within a neighborhood (and throughout the set of agents), agents differed in their capabilities: different agents had different sensing accuracies, making them better or worse at quickly gathering good information from the environment to share with their neighbors upon request. This follows in the tradition of other ad hoc environments (e.g., [6, 24]), where agents must work with agents with different capabilities than themselves.

The different opponents in our experiments included: (1) **KSMDP+MB-RL**: our Knowledge State MDP solution with MB-RL, using the UCT algorithm [17] to plan each time step using the learned MDP, (2) **KSMDP**: our Knowledge State MDP solution without MB-RL (also using UCT for planning, but only using the initial, uninformed K_T function where knowledge states only transition to the closest states), and two baselines: (3) **AlwaysSense**: where agents *maximized sensing* for information gathering and did not plan for information sharing since pre-coordination was not possible (which serves as a lower bound on acceptable agent performance), and (4) **RequestThenSense**: where agents requested information about each phenomenon every k steps to *maximize information sharing*, then either sensing the rest of the time to further inform agent knowledge or sharing if the agent had certain knowledge to help its neighbors.

We evaluated agent performance using three measures averaged per time step: (1) average belief certainty across all agents, (2) average proportion of agents with correct, highly certain knowledge, and (3) average total rewards earned by all agents. Each agent earned rewards: +10 whenever its b was sufficiently certain (i.e., $H(b) \geqslant 0.8$), -1 for every $Sense_i$ action, -1 for each $Request_j$ action (or -5 if $S^j_{Req} < k$), and +1 for each $Share_j$ action (whenever $S^j_{Rec} < k$, else -5). The other parameters were set: $|Ag| = 100$ (which is too large for multiagent POMDP solutions as a baseline), average $N_t(i) \in \{2, 4, 6, 8, 10\}$, $|P| = 1$, $|PS_j| = 3$, $cp = 1\%$, $acc \sim (0.5, 0.8)$, $\gamma = 0.99$, $k = 6$, $|K| = 100$. Each configuration repeated 50 times for 1000 time steps.

6. RESULTS

We begin our results analysis by considering the agent's average belief certainty, presented in Fig. 1. From these results, we first observe that our Knowledge State MDP solution with and without MB-RL (respectively KSMDP+MB-RL, KSMDP) achieved *higher amounts of belief certainty* than either of the baselines. This implies that, instead of trying to maximize either type of information gathering, our KSMDP formulation enabled agents to *appropriately select between information gathering actions using different sources to best refine their knowledge*, as opposed to either (1) requesting shared information as often as possible (RequestThenSense), or (2) independently relying only on sensed information (AlwaysSense).

Comparing across average neighborhood sizes, we observe that as neighborhood size increased and information became more available through sharing (due to each agent being connected to more potential information sources), the average certainty of the agents increased. Most notably, *certainty increased fastest for our KSMDP solutions*, implying that they became better at controlling information gathering as information became more readily available (although they also achieved the best performances when the neighborhoods were smallest and information was most limited).

Further comparing between the two variants of our solutions, we note that although adding MB-RL did not improve belief certainty very much, it did so at a 0.05 statistically significant level for the smaller average neighborhood sizes (2-6). This was when information was least available (due to fewer neighbors as sources) and thus more care was necessary for controlling information gathering. Therefore, *adding MB-RL to our Knowledge State MDP was most beneficial when information gathering most needed help.*

Next, we consider the average proportion of agents holding correct and highly certain beliefs, presented in Fig. 2. Maximizing this performance measure was the desired emergent behavior of solving the AHIG. From these results, we additionally observe that not only did our Knowledge State MDP-based solutions (KSMDP+MB-RL and KSMDP) lead

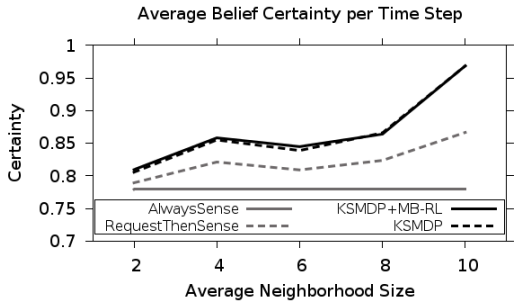

Figure 1: Average Belief Certainty

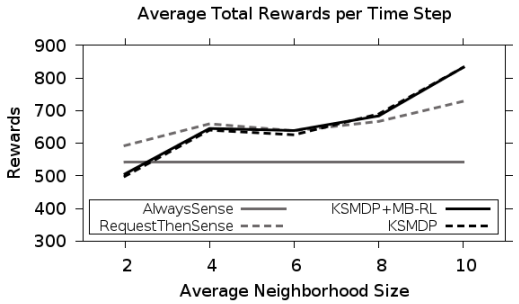

Figure 3: Average Total Reward

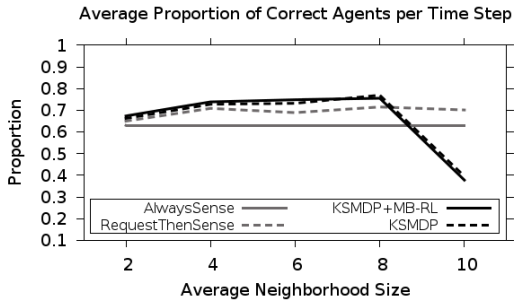

Figure 2: Average Proportion of Correct Agents Note: in all figs., 95% CI are too small to display

to more certainty in the agents' beliefs, but those beliefs were also correct. Thus, agents were *gathering the right information to understand their environments* over time. Additionally, we again find evidence of the benefits of using MB-RL to learn how agent knowledge changes based on different information gathering actions using different sources: the improvement over KSMDP (without MB-RL) for KSMDP+MB-RL *was more pronounced when information was most constrained* (i.e., at lowest neighborhood sizes).

Interestingly, we also observe that for the largest neighborhood size (10) considered in our experiments, our KSMDP solutions actually achieved very few correct agents compared to the baselines, which is in sharp contrast to the other neighborhood sizes. Upon further inspection, what happened is the agents fell victim to **institutional memory**: they converged to highly certain beliefs (as indicated in Fig. 1) because of the prevalence of shared information (favoring requesting information over continually sensing the environment). This caused the agents to become stuck with outdated beliefs that didn't adapt as the phenomenon changed over time since very few agents continued sensing the phenomenon directly. In the future, we intend to explore how we can adapt our solution to learn to avoid this problem.

Finally, we consider the average total rewards earned by all agents per time step, presented in Fig. 3. We observe that for all but the lowest neighborhood sizes, our KSMDP approaches–that directly maximized rewards to plan information gathering actions–earned the highest cumulative rewards due to achieving high certainty while trying to minimize costs. Of note, for the lowest neighborhood sizes (2-4) when information was most scarce, the KSMDP approaches were *willing to accept more information gathering cost in order to achieve higher certainty and correctness*, as displayed in Figs. 1-2, ultimately attaining the agents' primary goal.

7. CONCLUSIONS

In summary, we introduced the ad hoc information gathering (AHIG) problem occurring when agents must balance relying on different types of information sources (knowing when to sense vs. when to rely on shared information from other agents) in order to understand their complex environment without pre-coordinating with one another. From the tradition of using POMDPs to guide agent decision making, we proposed a transformation called the Knowledge State MDP that enables agents to control information gathering by reflecting on (fully observable) changes to their knowledge. To address the inability of agents to pre-coordinate in ad hoc environments, we added a MB-RL process to the Knowledge State MDP that enables agents to learn how their knowledge changes when relying on different information sources. This includes learning how and when neighbors might be willing to share information to supplement an agent's own sensing of the environment. Using an experimental study, we investigated the performance of our Knowledge State MDP (with and without MB-RL) in a range of environment configurations (with varying number of information sources), and discovered: (1) our solution gathered better information and earned greater rewards than baseline strategies of trying to maximize the usefulness of either type of information source (sensing vs. shared information), and (2) adding MB-RL enabled agents to best guided their behavior when information availability was most limited (and high quality information gathering was most necessary).

In the future, we intend to: (1) combine our solution with trust and reputation systems to further learn not only *when to rely on different information sources*, but *how much weight to place in received information*, which could help overcome the institutional memory problem (where weight w could be adapted to avoid agents rapidly converging to certain beliefs when shared information is prevalent), and (2) study how to use the Knowledge State MDP to balance information gathering about different phenomena in the environment to avoid *imbalanced knowledge* potentially caused by favoring sources for one phenomenon over the others.

Acknowledgments

This material is based upon work partially supported by the NSF (grant SES-1132015) and was completed utilizing the Holland Computing Center of the University of Nebraska.

REFERENCES

[1] N. Agmon, S. Barrett, and P. Stone. Modeling uncertainty in leading ad hoc teams. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'14)*, pages 397–404, Paris, France, May 5-9, 2014.

[2] M. Araya-Lopez, O. Buffet, V. Thomas, and F. Charpillet. A POMDP extension with belief-dependent rewards. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS'10)*, pages 64–72, Vancouver, B.C., Canada, December 6-9, 2010.

[3] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

[4] C. Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI'02)*, pages 239–246, Edmonton, Alberta, Canada, July 28-August 1, 2002.

[5] R. I. Brafman and M. Tennenholtz. R-max – a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2002.

[6] D. Chakraborty and P. Stone. Cooperating with a Markovian ad hoc teammate. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*, pages 1085–1092, Saint Paul, MN, May 6-10, 2013.

[7] F. Doshi and N. Roy. The permutable POMDP: fast solutions to POMDPs for preference elicitation. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'08)*, pages 493–500, Estoril, Portugal, May 12-16, 2008.

[8] A. Eck and L.-K. Soh. Observer effect from stateful resources in agent sensing. *Autonomous Agents and Multiagent Systems*, 26(2):202–244, 2013.

[9] P. Erdos and A. Renyi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, pages 17–61, 1960.

[10] K. Genter, N. Agmon, and P. Stone. Ad hoc teamwork for leading a flock. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*, pages 531–538, Saint Paul, MN, May 6-10, 2013.

[11] R. Glinton, P. Scerri, and K. Sycara. Exploiting scale invariant dynamics for efficient information propagation in large teams. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'10)*, pages 21–28, Toronto, Canada, May 10-14, 2010.

[12] R. Glinton, P. Scerri, and K. Sycara. An investigation of the vulnerabilities of scale invariant dynamics in large teams. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'11)*, pages 21–28, Taipei, Taiwan, May 2-6, 2011.

[13] P. J. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, 24:49–79, 2005.

[14] P. Hernandez-Leal, E. M. de Cote, and L. E. Sucar. Exploration strategies to detect strategy switches. In *Proceedings of the 14th International Workshop on Adaptive and Learning Agents (ALA'2014)*, Paris, France, May 5-6, 2014.

[15] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[16] L. P. Kaelbling, M. L. Littman, and W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[17] L. Kocsis and C. Szepesvari. Bandit based Monte-Carlo planning. In *Proceedings of the 17th European Conference on Machine Learning (ECML'06)*, pages 282–293, Berlin, Germany, September 18-22, 2006.

[18] R. Nair, P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI'05)*, pages 133–139, Pittsburgh, PA, July 9-13, 2005.

[19] O. Pryymak, A. Rogers, and N. R. Jennings. Efficient opinion sharing in large decentralized teams. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'12)*, Valencia, Spain, June 4-8, 2012.

[20] S. Ross, B. Chaib-draa, and J. Pineau. Bayes-adaptive POMDPs. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS'07)*, Vancouver, B.C., Canada, December 3-6, 2007.

[21] J. Sabater and C. Sierra. Reputation and social network analysis in multi-agent systems. In *Proceedings of the 1st International Conference on Autonomous Agents and Multiagent Systems (AAMAS'02)*, pages 475–482, Bologna, Italy, July 15-19, 2002.

[22] M. Sensoy, A. Fokoue, J. Z. Pan, T. J. Norman, Y. Tang, N. Oren, and K. Sycara. Reasoning about uncertain information and conflict resolution through trust revision. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*, pages 837–844, Saint Paul, MN, May 6-10, 2013.

[23] M. T. J. Spaan, T. S. Veiga, and P. U. Lima. Active cooperative perception in networked robotic systems using POMDPs. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10)*, pages 4800–4805, Taipei, Taiwan, October 18-22, 2010.

[24] P. Stone, G. A. Kaminka, S. Kraus, and J. S. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI'10)*, pages 1504–1509, Atlanta, GA, July 11-15, 2010.

[25] I. Szita and C. Szepesvari. Model-based reinforcement learning with nearly tight exploration complexity

bounds. In *Proceedings of the 27th International Conference on Machine Learning (ICML'2010)*, pages 1031–1038, Haifa, Israel, June 21-24, 2010.

[26] W. T. Teacy, J. Patel, N. R. Jennings, and M. Luck. TRAVOS: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multiagent Systems*, 12(2):183–198, 2006.