To appear in *Autonomous Agents and Multiagent Systems* (AAMAS), Valencia, Spain, June 2012.

An Analysis Framework for Ad Hoc Teamwork Tasks

Samuel Barrett and Peter Stone
Dept. of Computer Science
The University of Texas at Austin
Austin, TX 78712 USA
{sbarrett, pstone}@cs.utexas.edu

ABSTRACT

In multiagent team settings, the agents are often given a protocol for coordinating their actions. When such a protocol is not available, agents must engage in ad hoc teamwork to effectively cooperate with one another. A fully general ad hoc team agent needs to be capable of collaborating with a wide range of potential teammates on a varying set of joint tasks. This paper presents a framework for analyzing ad hoc team problems that sheds light on the current state of research and suggests avenues for future research. In addition, this paper shows how previous theoretical results can aid ad hoc agents in a set of testbed domains.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligience**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Algorithms, Experimentation

Keywords

Ad Hoc Teams, Multiagent Systems, Teamwork

1. INTRODUCTION

As the number of autonomous agents in society grows, so does the need for them to interact with other agents effectively. Both robots and software agents are becoming more common, and they are becoming more durable and robust, remaining deployed for increasing durations. Most existing methods for handling the interactions of agents require prior coordination, in the form of protocols for either coordination or communication. However, as agents stay deployed for longer, new agents are likely to be introduced that may not share these protocols. Furthermore, a multitude of different agents are under development in different businesses and research laboratories. Unfortunately, it is unlikely that these agents will all share a common world view or communication protocol. Therefore, it is desirable for agents to be capable of adapting to new teammates and learning to cooperate with previously unseen agents as part of an *ad hoc* team.

For example, consider a disaster rescue scenario where many different robots developed by many different people converge on the location of a disaster to attempt to locate and rescue victims. It is desirable for the robots to cooperate as a team, but in such a scenario, there is no time to program the robots to cooperate on site. If the robots were not explicitly designed to cooperate with one another, they will not work together and may even hinder each other. On the other hand, if some of these robots are programmed to reason about ad hoc teamwork, they may be able to quickly adapt their behaviors to cooperate with the other robots to rescue victims.

In a recent AAAI challenge paper, Stone et al. [15] introduced the concept of an *ad hoc team setting*, specifying it as a problem in which team coordination strategies cannot be specified a priori. They further presented a framework for evaluating the performance of an ad hoc team agent with respect to a domain and a set of possible teammates. The authors argued that the ad hoc teamwork challenge is inherently an empirical problem, but noted that little empirical research has been done in this area so far. On the other hand, some recent papers provide interesting theoretical results for some specific, isolated ad hoc team scenarios [16, 17].

This paper introduces a framework for understanding the relationships among existing lines of research, specifying several dimensions that are especially relevant when reasoning about the difficulty of different ad hoc team problems. Furthermore, this paper investigates several empirical scenarios and shows how existing theoretical solutions can be applied to these problems.

The remainder of the paper is organized as follows. Section 2 gives a more complete definition of ad hoc teamwork and specifies the empirical evaluation framework used in this work, and then Section 3 introduces a classification framework for ad hoc team problems along three important dimensions. Section 4 studies four different variations of an experimental domain, each representing a different point within the classification framework, and identifies their different solutions. Next, Section 5 explores other ad hoc teamwork domains, classifies them with respect to these dimensions, and suggests avenues for future research. Section 6 situates our research in literature, and Section 7 concludes.

2. AD HOC TEAMS

In an ad hoc team, agents need to cooperate with previously unseen teammates. Rather than developing protocols for coordinating an entire team, ad hoc team research focuses on developing agents that cooperate with teammates in the absence of such explicit protocols. Therefore, we consider a single agent cooperating with teammates that may or may not adapt to its behavior. We assume that we can only develop algorithms for the ad hoc team agent, without having any direct control over the other teammates.

For this work, we adopt essentially the same evaluation framework proposed by Stone et al. [15]. This framework is specified in Algorithm 1. According to this framework, the performance of the ad hoc team agent a depends on the distribution of problem domains D and the distribution of possible teammates A that it will

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

cooperate with. For the team B cooperating to execute the task d, $s(B,d)$ is a scalar score representing their effectiveness, where higher scores indicate better performance. The algorithm takes a sampling approach to average the agent's performance across a range of possible tasks and teammates to capture the idea that a good ad hoc team player ought to be robust to a wide variety of teamwork scenarios. We use s_{min} as a minimum acceptable reward for the team to be evaluated, because the ad hoc team agent may be unable to accomplish a task if its teammates are too ineffective, regardless of its own abilities. It is mainly used to reduce the number of samples required to evaluate the ad hoc agents and reduces the noise in the comparisons. Metrics other than the sum of the rewards can be used depending on the domain, such as the worst-case performance.

Algorithm 1 Ad hoc agent evaluation

Evaluate(a, A, D):

- Initialize performance (reward) counter $r = 0$.
- Repeat:
  - Sample a task d from D.
  - Randomly draw a subset of agents B, from A such that $E[s(B,d)] \geq s_{min}$.
  - Randomly select one agent $b \in B$ to remove from the team to create the team B^-.
  - Increment r by $s(\{a\} \cup B^-, d)$
- If Evaluate$(a_0, A, D) >$ Evaluate(a_1, A, D) and the difference is significant,then we conclude that a_0 is a better ad hoc team player than a_1 in domain D over the set of possible teammates A.

3. DIMENSIONS OF AD HOC TEAM PROBLEMS

Section 2 specified the framework for evaluating ad hoc team agents, but this evaluation depends on the specific domain and teammates that the ad hoc agent may encounter. In this section, we identify three dimensions of ad hoc teamwork settings that we believe can be used to better understand these domains and teammates. There are many possible ways that ad hoc team domains can vary, such as the size of the task's state space and the stochasticity of the domain. But we find that for differentiating among the algorithms in the existing literature, the following three are most informative.

1. **Team Knowledge:** Does the ad hoc agent know what its teammates' actions will be for a given state, before interacting with them?
2. **Environment Knowledge:** Does the ad hoc agent know the transition and reward distribution given a joint action and state before interacting with the environment?
3. **Reactivity of teammates:** How much does the ad hoc agent's actions affect those of its teammates?

These dimensions affect the difficulty of planning in the domain in addition to how much an ad hoc agent needs to explore the environment and its teammates. When an ad hoc agent has good knowledge, it can plan without considering exploration, but when it has incomplete knowledge, it must reason about the cost and benefits of exploration. The exploration-exploitation problem has been studied previously, but adding in the need to explore the teammates' behaviors and the ability to affect them considerably alters this tradeoff. Sections 3.1–3.3 provide further details about each of these dimensions, how we measure them, and why they are important for ad hoc teamwork.

To better illustrate the dimensions, we introduce a simple domain to evaluate across each of the dimensions. We describe the domain here and revisit it in the discussion of each dimension.

MatchActions: *This domain is a typical coordination game with two agents, each of which has two actions. If they select the same action, both receive a reward of r_i, where r_i is randomly selected from $\{0.5, 0.75, 1.0\}$ for $i \in 1, 2$, but fixed for the episode. On the other hand, if both agents select different actions, they receive a reward of 0. In addition, both agents can observe their teammates' previous actions. The ad hoc agent knows that its teammate is following one of two behaviors:*

- ***FirstAction**: the teammate always chooses the first action*
- ***BestResponse**: the teammate chooses the same action as the ad hoc agent did previously*

Therefore, the state can be represented as the previous action taken by the ad hoc agent, called s_0 if the ad hoc agent chose the first action, and s_1 otherwise.

3.1 Team Knowledge

The ad hoc agent's knowledge about its teammates' behaviors gives insight into the difficulty of planning in the domain. The agent's knowledge can range from knowing the complete behaviors of its teammates to knowing nothing about them. Settings with partial information are especially relevant, because in many real world problems, the exact behavior of a teammate may not be known, but some reasonable guidelines of their behaviors exist. For example, when playing soccer, one can usually assume that a teammate will not intentionally pass to the other team or shoot at the wrong goal. If the behaviors are completely known, the agent can reason fully about the team's actions, while if the behaviors are unknown, the agent must learn about them and adapt to find a good behavior.

To estimate the ad hoc agent's knowledge about its teammates' behaviors, we compare the actions the ad hoc agent expects them to take and the ground truth of what actions they take. Specifically, we compare the expected distribution of teammate actions to the true distribution that the teammates follow. To compute the difference between the distributions, we use the Jensen-Shannon divergence measure, which was chosen because it is a smoothed, symmetric variant of the popular Kullback-Leibler divergence measure. When the ad hoc agent has no information about a teammate's action, we assume that it uses the uniform distribution to represent its actions. Therefore, we define the knowledge measure as

$$K(T,P) = \begin{cases} 1 & \text{if } \mathrm{JS}(T,P) = 0 \\ 1 - \dfrac{\mathrm{JS}(T,P)}{\mathrm{JS}(T,U)} & \text{if } \mathrm{JS}(T,P) < \mathrm{JS}(T,U) \\ -\dfrac{\mathrm{JS}(P,U)}{\mathrm{JS}(U,\mathrm{Point})} & \text{otherwise} \end{cases} \tag{1}$$

where T is the true distribution, P is the predicted distribution, U is the uniform distribution, Point is a distribution with all weight on one point (e.g. $[1, 0, 0, \ldots]$), and JS is the Jensen-Shannon divergence measure. By this definition, $K(T,T) = 1$, so the knowledge is complete if the ad hoc agent knows the true distribution. $K(T,U) = 0$, representing when the ad hoc agent has no knowledge and relies on the uniform distribution. Finally, if the predicted distribution is less accurate than the uniform distribution, then $K(T,P)$ is negative, with a minimum value of -1. This measure captures the range [0,1] smoothly, but can still be used for the range [-1,0] [1]. However, we generally expect the prediction to

[1]One slight anomaly of this measure is that when T is the uniform distribution (e.g. [.5,.5]), K is either 1 when P is exactly correct at [.5 .5] or negative. For all other values of T, K smoothly spans the range [-1,1].

be a higher entropy distribution than the true distribution as the ad hoc agent ought to correctly model its uncertainty in its teammates' behaviors rather than being confident and wrong, which keeps the measure in the range [0,1].

We define the ad hoc agent's knowledge about its teammates' behaviors as

$$\text{TeamK} = \frac{\sum_{s=1}^{n}\sum_{t=1}^{k} K(\text{TrueAction}_t(s), \text{ExpAction}_t(s))}{nk}$$

where $1 \leq s \leq n$ is the state, $1 \leq t \leq k$ specifies a teammate, $\text{TrueAction}_t(s)$ is the ground truth action distribution for teammate t for state s, and $\text{ExpAction}_t(s)$ is the action distribution that the ad hoc agent expects teammate t to select for state s.

We assume that $\text{ExpAction}_t(s)$ is the uniform distribution if the agent has no information about teammate t's actions in state s. Thus, if the ad hoc agent has better information about its teammates' behaviors, the distance between the distributions will be smaller and TeamK will be higher.

Let us now calculate the TeamK for the MatchActions domain. The ad hoc agent has uniform beliefs over its teammate following either the FirstAction or BestResponse behaviors. However, the teammate is actually following the BestResponse behavior. With these beliefs, in s_0, the ad hoc agent expects that its teammate will always chose a_0, so $\text{ExpAction}_{s_0} = [1, 0]$. In s_1, the ad hoc agent thinks that the teammate will choose a_0 with probability 0.5 and a_1 with probability 0.5, while it actually chooses a_1 with probability 1. Thus,

$$\text{TeamK} = \frac{K([1,0],[1,0]) + K([0,1],[\frac{1}{2},\frac{1}{2}])}{2} = \frac{0+1}{2} = 0.5$$

This indicates that the ad hoc agent is fairly knowledgeable about its teammate's actions.

3.2 Environmental Knowledge

Another informative dimension is how much knowledge the ad hoc agent has about the effects of a joint action given a state, for example the transition and reward functions. If the ad hoc agent has complete knowledge about the environment, it can plan about what actions it should select more simply than if it must also consider unknown effects of actions. However, if it has incomplete knowledge, it must explore its actions and face the standard problem of balancing exploring the environment versus exploiting its current knowledge.

Similarly to teammate knowledge, we formally define the ad hoc agent's knowledge about the environment's transitions as

$$\text{TransK} = \frac{1}{nm}\sum_{s=1}^{n}\sum_{j=1}^{m} K(\text{TrueTrans}(s,j), \text{ExpTrans}(s,j))$$

where $1 \leq s \leq n$ is the state, $1 \leq j \leq m$ is a joint action, K is taken from Equation (1), $\text{TrueTrans}(s, j)$ is the ground truth transition distribution from state s given joint action j, and ExpTrans is the ad hoc agent's expected transition distribution. If the agent has no information about the transitions, we assume that $\text{ExpTrans}(s, j)$ is the uniform distribution. Intuitively, if the ad hoc agent knows more about the transition function, then the distance between TrueTrans and ExpTrans will be smaller and as a result TransK will be higher. We define the agent's knowledge about the environmental rewards similarly, and let $\text{EnvK} = (\text{TransK}, \text{RewardK})$.

Revisiting the MatchActions domain, the ad hoc agent knows the true transition function, as it only depends on the ad hoc agent's previous action, so $\text{TransK} = 1$. However, it only knows that the payoff for each action is uniformly drawn from {0.5, 0.75, 1.0} and the reward is 0 if the agents' actions do not match. There are 8 possible cases to count over, coming from 2 states and 2 actions for each of the agents, but the cases fall into 2 sets based on whether the actions match. In addition, it does not matter which value each matched action actually takes, so we can simplify the calculation. Note that there are four reward values possible: {0, 0.5, 0.75, 1.0}. This leads to

$$\text{RewardK} = \frac{4 * 1 + 4 * K([0,1,0,0],[0,\frac{1}{3},\frac{1}{3},\frac{1}{3}])}{8} = 0.582$$

Thus, $\text{EnvK} = (1, 0.582)$ As the agent observes these payoffs, it can refine its knowledge, but we are evaluating these properties prior to the ad hoc agent interacting with its environment.

3.3 Teammate Reactivity

The optimal behavior for the ad hoc agent also depends on how much its teammates react to its actions. If its teammates' actions do not depend on the ad hoc agent at all, the ad hoc agent can simply choose its actions to maximize the team reward, as if it were a single agent problem. Considering the actions of its teammates separately from that of the environment may still help computation by factoring the domain. However, if the teammates' actions depend strongly on the ad hoc agent's actions, the ad hoc agent's reasoning should consider what its teammates' reactions will be. If the ad hoc agent is modeling its teammates and its teammates are modeling the ad hoc agent, the problem can become recursive, as is directly addressed by Vidal and Durfee's Recursive Modeling Method [21].

A formal measure of the teammate reactivity needs to capture how different the teammates' actions will be when the ad hoc agent chooses different actions. We measure the distance between the resulting distributions of the teammate joint actions, using the pairwise Jensen-Shannon divergence measures. However, it is desirable for the distance to be 1 when the distributions have no overlap, so we use a normalizing constant of $\log 2$. Thus, we define the *reactivity* of a domain in state s as

$$\text{Reactivity}(s) = \frac{1}{(m-1)^2 \log 2}\sum_{a=1}^{m}\sum_{a'=1}^{m} \text{JS}(T(s,a), T(s,a'))$$

where JS is the Jensen-Shannon divergence measure, $1 \leq a, a' \leq m$ is the actions available to the ad hoc agent, and $T(s, a)$ is the distribution of the teammates' joint actions given the state s and the ad hoc agent's action, a. We use $m - 1$ in the denominator because we exclude the case where $a = a'$; in the numerator, the JS measure will be 0 in this case. For the overall reactivity of the domain, we average over the states, resulting in $\text{Reactivity} = \frac{1}{n}\sum_{s=1}^{n} \text{Reactivity}(s)$. It is possible to consider how an action affects the teammates' actions further in the future, but we restrict our focus to one step reactivity for this paper. Note that all of the sums in this formulation can be converted to integrals for continuous states or actions. This formulation is similar to the empowerment measure used by Jung et al. [13], but we consider the ad hoc agent's ability to change the actions of its teammates rather than the environment state.

Let us once again explore this dimension in the context of the MatchActions domain. Although the ad hoc agent is unsure of its teammate's behavior, the teammate is truly playing the BestResponse behavior. Thus, its actions are entirely dependent on the ad hoc agent's actions, so $\text{Reactivity} = 1$. If instead the teammate played BestResponse with probability $\frac{9}{10}$ and FirstAction with probability $\frac{1}{10}$, then we would get

$$\text{Reactivity}(s) = \frac{\text{JS}([1,0],[\frac{1}{10},\frac{9}{10}]) + \text{JS}([\frac{1}{10},\frac{9}{10}],[1,0])}{2\log 2} = 0.758$$

Therefore, we can conclude that the agent would still be very reactive, though not as reactive as the BestResponse agent.

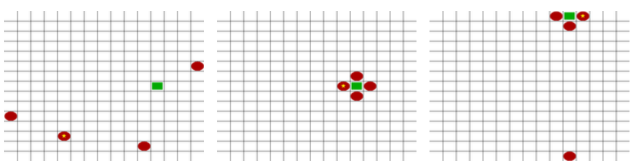
4. AD HOC TEAMWORK IN THE PURSUIT DOMAIN

Section 2 specified the framework for evaluating ad hoc team agents, but this evaluation depends on the specific domain and teammates that the ad hoc agent may encounter. Therefore, in this section, we study several concrete versions of a domain that require the cooperation of a team. Then, we explore how ad hoc agents should handle these various domains, and explain how these domains are characterized by the dimensions presented in Section 3.

4.1 The Pursuit Domain

The pursuit domain has become a popular setting for multiagent research [18]. It lends itself well to ad hoc team problems as it requires multiple agents to cooperate to capture the prey. The general idea of the pursuit problem is that a number of predators attempt to chase and finally "capture" a prey, but there are several variations of the pursuit, depending on the number of predators, the definition of "capture," and the mechanics of the world. Here we specify a number of variations of the pursuit domain that are interesting for investigating ad hoc teamwork.

4-Predator Known Behaviors Pursuit (4PKB): *In this fairly common formulation of the pursuit problem, there are four predators trying to capture a single prey, while moving around a toroidal grid, where moving off one side brings the agent back on the other side. Therefore, all four predators are required to capture the prey by surrounding it, as illustrated in Figure 1. In this formulation, all agents can fully observe the positions of the other agents and the prey moves randomly. One of the predators is an ad hoc agent and, at each time step, it must choose a direction to move to cooperate with its teammates. The other three predators follow a fixed behavior that is known to the ad hoc agent.*



(a) Random start (b) Capture position (c) Another capture

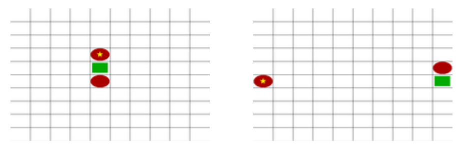
Figure 1: Start and capture positions in the 4PKB and 4PUB domains. The green rectangle is the prey, the red ovals are predators, and the red oval with the star is the ad hoc predator (the one under our control that is being evaluated).

Before continuing with describing the other domain variants, we introduce a number of high level behaviors that the predators may use to capture the prey. Specifically, we consider the following four individual predator behaviors described in Barrett et al.'s work [1]:

1. Greedy (**GR**)- Move towards the nearest unoccupied cell neighboring the prey with minimal obstacle avoidance
2. Greedy Probabilistic (**GP**) - Same as the GR behavior except that the predator has a chance of taking a longer path to its desired cell
3. Teammate-aware (**TA**) - Assign cells neighboring the prey to the teammates, minimizing the movement required by the farthest predator; move towards the assigned cell
4. Probabilistic Destinations (**PD**) - Spread out from other predators into a circle that tightens around the prey over time

4-Predator Unknown Behaviors Pursuit (4PUB): *This version is identical to 4PKB, except that the ad hoc agent is not given full information about its teammates' behaviors. Instead, the agent is given a set of known behaviors that its teammates are possibly playing. In this case, the ad hoc agent must observe its teammates and try to determine their behaviors based on their actions. For example, the ad hoc agent may be initially given a uniform distribution over the GR, GP, TA, and PD behaviors. By observing its teammates' actions, the ad hoc agent may be able to determine that all of its teammates are following the TA behavior.*

2-Predator Simultaneous Pursuit (2PS): *In this formulation of the pursuit problem, two predators move on a toroidal grid and attempt to capture the prey by simultaneously occupying any two cells neighboring the prey, as shown in Figure 2. Instead of choosing an action at every time step of an episode, the agents choose a high level behavior to play for the duration of an episode. This high level behavior defines their actions at each time step. In between episodes, the agents can choose a new behavior to play, based on their previous experience. Once again, one of the predators is controlled by the ad hoc agent, and its teammate chooses the behavior by best response, using a memory bound of k. For example, at the beginning of each episode, each predator could choose to play the GR, GP, TA, or PD behavior. If $k = 1$ and the ad hoc agent chose the GR behavior last step, its teammate would choose to play the GR behavior this step because it knows that this will result in the shortest time to capture the prey if the ad hoc agent continues playing the GR behavior.*



(a) Capture position (b) Another capture

Figure 2: Capture positions in the 2PS domain

2-Predator Teaching Pursuit (2PT): *In this version, two predators are trying to capture the prey by choosing a high level behavior to follow, similar to the 2PS domain. However, each predator must capture the prey independently, and the predators alternate episodes. Therefore, instead of cooperating during an episode, the predators cooperate between episodes. Each predator observes what high level behavior the other chose as well as how long it takes to capture the prey using that behavior. In addition, the predators are still a team and share rewards. In this domain, capture is defined as the predator occupying the same cell as the prey. For example, if the ad hoc agent chooses to play the GR behavior, its teammate observes that it chose the GR behavior as well as the length of the episode. The teammate can then use this information when it is selecting a behavior to play for the next episode.*

In addition, the ad hoc team agent has full knowledge about the performance of the behaviors, while its teammate starts with no knowledge and acts greedily with respect to the behaviors' observed sample means. If the ad hoc agent was not on a team, it could perform optimally by choosing the behavior with the best expected reward, but its teammate can observe its actions and learn from them. The ad hoc agent knows that its teammate is greedy with respect to the observed means of the different behaviors. However, the teammate has noisy actuation, so it is unable to perform deterministic behaviors, unlike the ad hoc agent. Unfortunately, the behavior with the best expected time to capture the prey is deterministic. Therefore, there is a cost to teaching as the ad hoc team agent must forego playing the best behavior to increase its teammates knowledge.

4.2 Prior Ad Hoc Teamwork Results

In Section 4.1, several variations of the pursuit domain were presented, some of which have been studied in prior research. Both the 4PKB and 4PUB domains were investigated by Barrett et al. [1]. In their work, Barrett et al. assume that three of the predators use one of the specified behaviors (the same one in most cases) and the fourth predator is the ad hoc team agent. Their ad hoc team agent plans efficiently using Monte Carlo Tree Search (MCTS), selecting actions that are expected to capture the prey quickly given its models of its teammates. In the 4PKB domain, the ad hoc agent knows the true behavior of its teammates, but in the 4PUB domain, it is only given a set of possible behaviors of its teammates. Therefore, the ad hoc agent agents tracks the probabilities that its teammates are using the known behaviors, updating their probabilities using Bayes' rule and the probability of each behavior taking the observed actions. At each time step, the agent plans using MCTS and samples from the possible teammate models with respect to their relative probabilities. This approach results in an effective ad hoc team agent.

In Barrett et al.'s work, the ad hoc agent has complete information about the environment (EnvK $= (1, 1)$), but its knowledge about its teammates is varied in different tests. In the 4PKB setting, the ad hoc agent knows the true behavior of its teammates, so TeamK $= 1$. In the 4PUB setting, it only knows that its teammates are drawn from a set of known models; resulting in TeamK $=$ 0.720 for a 5x5 world, while on a 20x20 world TeamK $= 0.807$. Finally, there are tests where the set of representative behaviors are known to the agent, but the teammates' behaviors are not drawn from this set. Instead, these agents are sampled from a set of predator behaviors written by students for a class assignment. In this case, TeamK $= 0.155$ on a 5x5 world and TeamK $= 0.237$ on a 20x20 world.

In this work, the reactivity of the teammates depends on the behavior that the teammates run as well as the size of the world. For the GP teammates on a 5x5 grid, the reactivity is only 0.0635, while if the teammates play the TA behavior, the reactivity is 0.501. Similarly, on a 20x20 grid, the reactivity of GP teammates is 0.00105 and for TA teammates it is 0.0809.

In the pursuit domain, the challenge arises from a combination of the reactivity of the teammates and the agent's imperfect knowledge about its teammates. In this research, only a single episode was considered, so there was no long term learning; the agents had to learn during the episode.

While Barrett et al. investigated the 4PKB and 4PUB domains, the pursuit domain allows for many small variations that have a large impact on where it falls along the dimensions laid out in Section 3. Sections 4.3–4.4 explore the 2PS and 2PT variants, all within the context of our framework from Section 3.

4.3 Repeated Interactions with a Best Response Agent

Whereas both the 4PKB and 4PUB domains assume that the ad hoc agent interacts with its teammates for only one episode, many teamwork settings allow for multiple interactions among the same teammates. In this case, long-term learning (across episodes) is both possible and very useful. In order to model such settings, in this section we investigate the 2PS domain.

In the 2PS domain, the ad hoc agent has perfect information about its teammate, so TeamK $= 1$. Also, the ad hoc agent completely knows the environment's transitions and rewards, so EnvK $=$ $(1, 1)$. The reactivity of the domain is high, since the teammate's actions depend highly on the ad hoc agent's actions. However, the reactivity depends on the specific behaviors that are available to the agents as well as the memory size of the teammate, k. The available information about the teammate reduces the difficulty of this problem compared to the earlier pursuit problem, but the reactivity of this problem is much higher. Therefore, the problem is still difficult, but many of the issues that must be faced are different.

Analysis of 2PS reveals that it can be modeled as a repeated normal-form game in which the agents share the payoffs. In this setting, there is a matrix of shared payoffs and two agents; one chooses a row and one that chooses a column, where the rows and columns correspond to different behaviors that can be chosen. One of the agents is a k-memory bounded, best response agent, meaning that it chooses the action that has the best expected payoff given the other agent's last k actions. The other agent is the ad hoc team agent, and its goal is to cooperate with the best response agent to achieve the highest payoff.

There is a cell in the payoff matrix with the highest reward that is best for both agents. However, if the ad hoc agent jumps immediately to the corresponding behavior, it may incur a high loss before the best response agent moves to the best action, where the loss is defined as the difference between the maximum possible reward and the received reward. Therefore, it may be desirable for the ad hoc agent to take a longer path through the payoffs, minimizing the losses.

Stone et al. [16] investigated the class of ad hoc teamwork problems that can be modeled by this normal-form game formulation. Their work provides several theoretical results as well as an efficient algorithm for finding the optimal action sequence when $k =$ 1. Furthermore, they give an algorithm for dealing with larger memory bounds, $k > 1$, but this algorithm is exponential in the memory size. Also, they consider the case where the teammate is non-deterministic and differs from the k memory bounded best response by ϵ.

In this paper, the predators select from the TA, PD, and GR behaviors. If they play on a 5x5 grid and each agent has a 0.1 chance of taking a random action, the resulting payoff matrix is given in Table 1. These payoffs were calculated by running 1,000 episodes with the agents following the specified behaviors, where the team receives an action penalty of -1 for each step until the prey is captured.

	TA	PD	GR
TA	-4.583	-5.123	-5.152
PD	-5.123	-4.946	-4.615
GR	-5.152	-4.615	-4.379

Table 1: Payoff matrix from the pursuit domain

Assume that the agents start with both agents playing the TA behavior (a Nash equilibrium) and $k = 1$. The best payoff is when both agents play the GR policy, so the ad hoc team agent wants to find the lowest cost path to that policy combination. This occurs if the ad hoc agent chooses the PD policy and then the GR policy from then on. The best response teammate will play the TA policy for the first two steps (because it is the best response to TA) then change to the GR policy, which is the best response to PD. The loss of this path is $-4.379 - -5.123 = 0.744$ while changing directly has a cost of $-4.379 - -5.152 = 0.773$. Therefore, it is advantageous for the ad hoc agent to take a longer path to its desired policy. The efficient algorithm laid out by Stone et al. finds this solution.

In this domain, both agents know the transitions and rewards, so EnvK $= (1, 1)$. Also, the ad hoc agent knows that its teammate uses the best response policy, resulting in TeamK $= 1$. The reactivity is 0.198, as the ad hoc agent's actions do influence its teammate's actions.

The version of the pursuit domain considered in this section illustrated how prior theoretical results can be applied directly in an experimental setting. In the next section, we see the same for a different theoretical approach to ad hoc teamwork.

4.4 Teaching a Novice Agent

To this point, we have assumed that the teammate has complete knowledge about the performance of the behaviors, but in some settings this is not the case. To explore such settings, we investigate the 2PT domain, in which the teammate starts with no knowledge about each behavior and must explore the behaviors to estimate their performance.

However, we assume that the ad hoc team agent has full knowledge about the behaviors and its teammate. Also, instead of having the two predators cooperate directly, we consider the case where they take turns trying to capture the prey, but both observe the results of the other's actions. Unfortunately, due to a defect, the teammate is not able to execute all of the behaviors that are available to the ad hoc agent, including the behavior with the best expected time to capture the prey. Therefore, there is a cost to the ad hoc team agent foregoing this best behavior in favor of another that will teach its teammate. The agents are still trying to maximize their shared rewards, but they take turns choosing behaviors to capture the prey. The ad hoc agent has complete information about the effectiveness of the behaviors, so it should help guide its naive teammate towards behaviors that are more effective. We consider the case where the teammate chooses behaviors greedily with respect to the sample means it has seen.

In this domain, the ad hoc team agent is the *teacher*, and it has perfect knowledge about its teammate, i.e. TeamK $= 1$. Also, its information about the environment is perfect in both the transition and reward distributions, i.e. EnvK $= (1, 1)$. Note that the other agent only has limited information about the environment. However, the reactivity of the domain depends on the number of episodes as well as the payoff distributions of the shared behaviors, but not the behavior that only the teacher can play. Similar to the scenario in Section 4.3, the challenge arises from the ad hoc agent needing to plan about the reactivity of its teammate. In addition, it must also consider how much information its teammate has, and how this affects the teammate's actions.

Therefore, the ad hoc team agent should reason about when it is useful to sacrifice choosing the behavior with the highest reward to teach its teammate about which behaviors it should be choosing. Close examination of this problem reveals that it can be modeled by a multi-armed bandit (MAB), such as that proposed by Stone and Kraus [17], where the different behaviors correspond to different arms of the bandit. In this setting, choosing a high level behavior to play for an episode corresponds to pulling the arm on the multi-armed bandit, and the length of the episode corresponds to the payoff of the arm. Stone and Kraus prove several interesting theoretical results about this formulation of ad hoc teamwork in a multi-arm bandit domain regardless of the payoff distributions of the arms, proving some theorems about when the ad hoc agent should and should not teach. Furthermore, they give efficient algorithms for handling cases where the payoff distributions are constrained.

From this research, we know that the ad hoc agent should consider playing behaviors other than the best one, which the teammate cannot play. In other words, it is advantageous for the ad hoc agent to teach its teammate despite the cost of teaching. Also, the ad hoc agent should never play the worst of the behaviors, even when the teammate's estimates of that behavior's quality is too high. Furthermore, with discrete distributions for the payoffs, Stone et al. give a polynomial time algorithm for calculating the optimal behavior for the ad hoc agent.

Consider the case in which the agents play on a 5x5 grid and must choose from three policies: the greedy, probabilistic destinations, and greedy probabilistic predators from Section 4.1. In this case, the teammate has non-deterministic actions and therefore cannot follow the purely greedy policy. These different policies give average capture times of 4.204, 4.272, and 4.911 respectively for a single predator capturing the prey, where a smaller time is better. The reactivity of this problem is 0.0342 if we consider histories of actions of length 100, and 0.128 for histories of length 10. Overall, a state is more reactive when the teacher can change the relative values of the arms' sample means, which happens when there are a small number of pulls that are far from the true arm means. When there are many pulls or the sample means closely match the true means, the reactivity is very low.

From Stone et al.'s work, we know that the ad hoc agent should consider sacrificing its reward from following the greedy policy to play other behaviors and teach its teammate. Also, we know that it should never play the greedy probabilistic behavior, even if its teammate thinks that this behavior is best.

5. CLASSIFYING EXISTING AD HOC RESEARCH DOMAINS

Section 4 introduces several ad hoc team problems and explains how they are described by the dimensions proposed in Section 3. We summarize those results here, and continue on to explore other domains used in previous ad hoc research. We then use these results to suggest new avenues for research into ad hoc teamwork.

5.1 Characteristics of the Pursuit Domain

A summary of the characteristics of variations of the pursuit domain is given in Table 2. Note that the reactivity of 2PS relies on the memory bound k of the agent (0.198 if $k = 1$), and in 2PT, it relies on the length of pull histories considered (0.0342 if the history size is 100 and 0.128 for length 10). For 4PKB and 4PUB, the values are affected by the size of the domain and the specific behaviors used by the teammates. Specifically, the reactivity is 0.00105 when for GP teammates on a 20x20 world, and 0.501 for TA teammates on a 5x5 world. The TeamK varies from 0.155 when cooperating with the teammates created by students on a 5x5 world to 0.807 for teammates drawn from {GR,GP,TA,PD} on a 20x20 world. From this table it becomes clear that research into ad hoc teamwork has focused mainly on the reactivity of the problems, with some approaches handling some uncertainty about the teammate behaviors. However, no ad hoc teamwork research thus far has handled any domains in which the environment is unknown. In addition, deviating slightly from the assumptions of either the 2PS or 2PT domains can render the theoretical results incorrect. Therefore, it is an important future direction to create general methods for handling problems with perfect knowledge about the teammates and environment, with varying reactivities. We hypothesize that the MCTS agent from Barrett et al. [1] should perform well in these domains, but this exploration remains open.

Domain	TeamK	EnvK	Reactivity
4PKB	1	(1,1)	0.00105–0.501
4PUB	0.155–0.807	(1,1)	0.00105–0.501
2PS	1	(1,1)	0.198
2PT	1	(1,1)	0.0342–0.118

Table 2: A summary of pursuit problems

5.2 Characteristics of Other Domains

Besides pursuit, there have been several other domains used in ad hoc teamwork research, though not always under the name of ad hoc teamwork. Although it is difficult to exactly calculate some of the dimensions without exact specifications of the domains, we estimate the values in Table 3.

Han et al. [9] explore using an agent to affect the collective behavior of a multi-agent system. Specifically, their work focused on adding a "shill" agent that was externally controlled, corresponding to the ad hoc team agent in our terminology. They then investigated using this agent to affect the behavior of groups of agents such as flocks of birds, directing the movement of the flock in a desired direction. They use a modified Boid model, in which each agent chooses its current heading by moving in the average direction of their neighbors located within a neighborhood of radius r. Table 3 summarizes how the domain is characterized along the dimensions. In this case, the shill agent knows its teammates' behavior and the environment, and the reactivity of its teammates depends on the number of agents and the size of the neighborhood. For these calculations, we discretize the actions into 10 degree bins. With 5 agents, the reactivity ranges from 0.880 when $r = 5$ to 0.0106 when $r = 1$, while with 100 agents, it ranges from 0.531 to 0.0732 for the same r values. This shows that while the reactivity can be large, in many settings the teammates are not strongly affected by the ad hoc agent's actions in the short term. However, Han et al.'s work shows that the long term effects of the ad hoc agent's actions are very influential, as the effects ripple out to the other agents.

Domain	TeamK	EnvK	Reactivity
Flocking control	1	(1,1)	0.0732–0.880
Cooperating with UTM-1 teammates	0	(1,1)	0
Cooperating with UTM-2 teammates	0	(1,1)	>0
Simulated pickup soccer	>0	(>0,1)	>0

Table 3: Estimates of other ad hoc team problems

More recently, Wu et al. [22] investigated ad hoc teamwork with few assumptions about the behaviors of the teammates. Their ad hoc agent plans using MCTS and uses biased adaptive play to predict the actions of teammates. Biased adaptive play can be used to estimate the policies of teammates from their previous actions. They test their agent on three domains: cooperative box pushing, meeting in a 3x3 grid, and multi-channel broadcast. They consider the case where the ad hoc agent knows the environment, but not its teammates. These teammates are referred to as unknown teammates (UTM), and two types of teammates are used in each domain: UTM-1 agents that follow a fixed set of actions and UTM-2 agents that try to play the optimal behavior but have partial observations. Along the dimensions, only the reactivity of the teammates vary between these three domains. However, the specifications of the UTM-2 agents is only that they act rationally with respect to partial observations of the system state, so it is not possible to calculate the exact values of the reactivity; it is only known that their reactivity is greater than 0. If the UTM-2 agents perform close to the rational behavior given full observations, it is expected that their reactivity is very high in these domains.

In the domain of simulated robot soccer, Bowling and McCracken [3] measure the performance of a few ad hoc agents, where each ad hoc agent is given a playbook that differs from that of its teammates. In this domain, the teammates implicitly assign the ad hoc agent a role, and then react to it as they would any teammate. This means that they react to the ad hoc agent's actions, i.e. the reactivity is greater than 0, but the extent of this reactivity is depends on their standard soccer behavior. In addition, the ad hoc agent knows a set of possible plays that may overlap with the plays that its teammates choose. It is expected that its knowledge of its teammates is fairly high, as effective soccer plays are similar, compared to random movement of the teammates, therefore TeamK is greater than 0. Although the ad hoc agent does not know ahead of time the noise caused by the simulation of the game or by the noise caused by the other team, it has reasonable expectations of the dynamics of the world, so TransK is greater than 0. In addition, it knows that scoring goals gives a positive reward, and giving up goals gives a negative reward, so RewardK is 1.

5.3 Characteristics of Future Research

Looking at how the existing research fits into the proposed dimensions gives us insight on directions for future investigation. Most research on ad hoc teams has focused on how teammates react to the ad hoc agent, as shown by the high levels of reactivity in existing domains. Little research has approached the problem of having low knowledge of the teammates, where the ad hoc agent must learn about its teammates to plan effectively. Wu et al.'s work [22] assumes that the ad hoc agent knows nothing about its teammates, but they focus on smaller domains. Barrett et al. [1] consider the idea that the ad hoc agent may start with some knowledge about its possible teammate, but must still learn about them by interacting with them. More research needs to be performed to investigate cases where the ad hoc agent knows little about its teammates. In many cases, agents can make some reasonable assumptions about the behavior of their teammates. Therefore, it is desirable to focus on ad hoc agents that cooperate with teammates starting with low, but nonzero information about their behaviors. In this case, the ad hoc agent must learn more about its teammates by interacting with them.

In addition most ad hoc teamwork research assumes that the ad hoc agent completely knows the transition dynamics of the environment as well as the short term rewards of actions. In other words, research into ad hoc teamwork has mainly focused on the difficulties of planning effectively with teammates, where the agent does not need to learn about the environment. On the other hand, many real world applications of ad hoc teamwork requires the agent to learn about its environment and adapt accordingly. In the case of search and rescue, robots must cooperate with previously unseen teammates, but they must also adjust to a noisy, changing environment. To perform effectively while exploring new environments, such as those encountered in exoplanet exploration, robots must learn about how their actions interact with the world and handle changes to their abilities caused by wear and tear. Therefore, future research in ad hoc teamwork should incorporate domains in which the dynamics begin unknown. This will create ad hoc agents that can trade off between exploring the environment, exploring interactions with their teammates, and exploiting their current knowledge. We believe that research in these areas is necessary to create robust, effective ad hoc agents.

6. RELATED WORK AND DISCUSSION

Aside from the ad hoc teamwork domains described in Section 5, some other research into ad hoc teams exists, such as Jones et al.'s [12] research into pickup teams working in the treasure hunt domain. This work assumes that the agents share a communication protocol that they use to bid on different roles. In addition, Knudson and Tumer [14] investigated ad hoc teams in a different framework. However, they assume that all agents in the domain adapt and that each agent is given a difference objective, which clearly specifies how an agent's actions affect its team's performance. Earlier research includes Brafman and Tennenholz's research [4] on agents performing a repeated joint task, where one agent attempts to teach a novice agent. A large body of research on coordinating multi-agent teams exists, specifying standardized protocols for communication and shared algorithms for coordination. These approaches include SharedPlans [8], STEAM [19], and GPGP [7].

Our work does not require any shared protocols, and does not assume that the teammates are adapting to the ad hoc team agent.

Modeling teammates is similar to the problem of opponent modeling, but it is generally safe to make stronger assumptions about teammates' behaviors. Therefore, the ad hoc agent does not need to consider the worst case scenario for every action; its teammates are trying to reach the same goal. The Workshop on Plan, Activity, and Intent Recognition (PAIR) and the Workshop on Applied Adversarial Reasoning and Risk Modeling (AARM) both have several papers relevant to applied opponent modeling, and there are also more theoretical approaches such as the AWESOME algorithm [6]. An interesting avenue for future work is to classify the much broader existing literature on opponent modeling along the same dimensions presented in this paper. These dimensions may aid in identifying approaches from opponent modeling literature that are likely to apply in corresponding ad hoc teamwork domains.

Isaacs performed seminal research on pursuit and evasion [10], and the problem was further explored by Benda et al. [2]. The pursuit domain has been well studied in multiagent research [18]. Most previous research focused on developing coordinating the predators before deploying them, rather than learning to adapt to unseen teammates. For example, MAPS [20] considers partially observable environments with the prey following sophisticated strategies, but requires a shared coordination algorithm. Other approaches focus on partial observability in continuous pursuit problems [11]. On the other hand, Chakraborty and Sen [5] investigate a pursuit scenario in which experienced agents attempt to teach novice predators, but require the agents to share a known training protocol. However, none of these methods are directly applicable to ad hoc teamwork.

7. CONCLUSION

This paper presents a set of dimensions for describing ad hoc team problems, and explains how these dimensions define the relationship among existing ad hoc team research studies. We show that reasoning about these dimensions aids in applying existing theoretical results to new problems.

The introduction of the dimensions describing the difficulties of ad hoc team problems raises several interesting avenues for future research. For example, by examining existing research in this light, it becomes clear that research thus far on ad hoc teams assumes that the adhoc agent knows the environment. Future work is needed on domains where the environment is unknown and the ad hoc agent must reason about the tradeoffs of exploration. Furthermore, domains where the ad hoc agent has less information about its teammates ought to be investigated. All of the results in this paper are in the context of the pursuit domain, broadly defined. Investigations of other domains, and whether existing algorithms apply in these domains is an important avenue for future research. From a high-level perspective, this research contributes towards understanding and solving the long-term challenge of creating robust, general ad hoc team agents.

Acknowledgments

This work has taken place in the Learning Agents Research Group (LARG) at The University of Texas at Austin. LARG research is supported in part by grants from NSF (IIS-0917122), ONR (N00014-09-1-0658), and the FHWA (DTFH61-07-H-00030). Samuel Barrett is supported by an NDSEG fellowship.

8. REFERENCES

[1] S. Barrett, P. Stone, and S. Kraus. Empirical evaluation of ad hoc teamwork in the pursuit domain. In *AAMAS '11*, May 2011.

[2] M. Benda, V. Jagannathan, and R. Dodhiawala. On optimal cooperation of knowledge sources - An empirical investigation. Technical Report BCS–G2010–28, Boeing Advanced Technology Center, Boeing Computing Services, July 1986.

[3] M. Bowling and P. McCracken. Coordination and adaptation in impromptu teams. In *AAAI*, pages 53–58, 2005.

[4] R. I. Brafman and M. Tennenholtz. On partially controlled multi-agent systems. *JAIR*, 4:477–507, 1996.

[5] D. Chakraborty and S. Sen. Teaching new teammates. In *AAMAS '06*, pages 691–693, 2006.

[6] V. Conitzer and T. Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Mach. Learn.*, 67, May 2007.

[7] K. S. Decker and V. R. Lesser. Designing a family of coordination algorithms. In *ICMAS '95*, pages 73–80, June 1995.

[8] B. Grosz and S. Kraus. Collaborative plans for complex group actions. *Artificial Intelligence*, 86:269–368, 1996.

[9] J. Han, M. Li, and L. Guo. Soft control on collective behavior of a group of autonomous agents by a shill agent. *Journal of Systems Science and Complexity*, 19:54–62, 2006.

[10] R. Isaacs. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. Dover Publications, 1965.

[11] Y. Ishiwaka, T. Sato, and Y. Kakazu. An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning. *Robotics and Autonomous Systems*, 43(4):245 – 256, 2003.

[12] E. Jones, B. Browning, M. B. Dias, B. Argall, M. M. Veloso, and A. T. Stentz. Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In *ICRA*, pages 570 – 575, May 2006.

[13] T. Jung, D. Polani, and P. Stone. Empowerment for continuous agent-environment systems. Technical Report AI-10-03, The University of Texas at Austin Computer Science Department, 2010.

[14] M. Knudson and K. Tumer. Robot coordination with ad-hoc team formation. In *AAMAS '10*, pages 1441–1442, 2010.

[15] P. Stone, G. A. Kaminka, S. Kraus, and J. S. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI '10*, July 2010.

[16] P. Stone, G. A. Kaminka, and J. S. Rosenschein. Leading a best-response teammate in an ad hoc team. In *AMEC*. November 2010.

[17] P. Stone and S. Kraus. To teach or not to teach? Decision making under uncertainty in ad hoc teams. In *AAMAS '10*, May 2010.

[18] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3):345–383, July 2000.

[19] M. Tambe. Towards flexible teamwork. *JAIR*, 7:81–124, 1997.

[20] C. Undeger and F. Polat. Multi-agent real-time pursuit. *AAMAS '10*, 21:69–107, July 2010.

[21] J. M. Vidal and E. H. Durfee. Recursive agent modeling using limited rationality. In *ICMAS*, 1995.

[22] F. Wu, S. Zilberstein, and X. Chen. Online planning for ad hoc autonomous agent teams. In *IJCAI*, 2011.