Proceedings, The Eleventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-15)

Tuning Belief Revision for Coordination with Inconsistent Teammates

Trevor Sarratt and Arnav Jhala
Baskin School of Engineering
University of California, Santa Cruz, Santa Cruz, CA 95064
{tsarratt, jhala}@soe.ucsc.edu

Abstract

Coordination with an unknown human teammate is a notable challenge for cooperative agents. Behavior of human players in games with cooperating AI agents is often sub-optimal and inconsistent leading to choreographed and limited cooperative scenarios in games. This paper considers the difficulty of cooperating with a teammate whose goal and corresponding behavior change periodically. Previous work uses Bayesian models for updating beliefs about cooperating agents based on observations. We describe belief models for on-line planning, discuss tuning in the presence of noisy observations, and demonstrate empirically its effectiveness in coordinating with inconsistent agents in a simple domain. Further work in this area promises to lead to techniques for more interesting cooperative AI in games.

Introduction

Effective teamwork relies on coordination of individual team members which, in turn, requires understanding of the goals, intentions, and skills of each team member. There are different levels of complexity in which these can be modeled for an AI agent interacting with a human. In the simplest form these could just be assumed and hard coded so that the agent assumes the human is following exactly the most obvious, immediate goal. A slightly more sophisticated model would have the agent observe actions of the human collaborator and refine a belief model about the intentions of the player. While both cases will support specific instances of well-designed collaborative situations, they may fail for some players that are either unclear about their goal or don't have the skill (or plan) to achieve the goal.

Much of cooperation in games is based on the human player in the lead role. From a player-centric design point of view this is reasonable, but assuming AI to be a true peer with equal participating in the collaboration could possibly open up a design space of interactions that has yet been unexplored leading to emergent cooperative play. To this end, we first take a small but well-studied domain and provide a detailed empirical analysis for the feasibility of the idea that an AI agent create and maintain a belief model of an inconsistently behaving collaborator in order to maximize success in a cooperative task.

In this paper, we discuss an approach, Responsive Action Planning with Intention Detection (RAPID), for updating beliefs over an unfamiliar teammate's goals with fast adaptation to changes. It is often inaccurate to assume a teammate will stick to a single goal throughout a task, especially when state transitions provide incentive to switch, whether it be an easier route to a goal or simply a more appealing one. An ideal supporting agent should not only be able to assist its teammate in achieving its goals but also be flexible in its planning capacity to account for such changes in teammate behavior, much like a human team member would. In order to achieve this capacity in collaborative agents, it is necessary to adjust how beliefs are revised by an agent planning in a partially observable space. RAPID models the planning space as a partially observable Markov decision process (POMDP). POMDPs provide a decision theoretic basis for evaluating a plan of action under uncertainty. We restrict this uncertainty to the teammate's behavior, such that an agent must infer through observations the current goal. Furthermore, unlike much of the existing work, we do not assume a static hidden model. Instead, the teammate may switch between many potential behaviors, and the task of the agent is further complicated by identifying when these transitions occur. This relaxation provides a fairly intuitive representation of how a human player may adopt a new plan on the fly according to his or her own preferences. When enough observed evidence favors switching to a new goal, the agent can adapt quickly and coordinate more effectively.

Related Work

Uncertainty and Complexity in Multi-Agent Planning

One of the foremost hurdles for multi-agent team decision problems is computational complexity. MDP-based scenarios with uncertainty on both agents' sides with regard to world state and observation history as well as fully recursive modeling between agents fall under the category of decentralized partially-observable Markov decision problems (DEC-POMDPs). Even with a finite horizon assumption for planners, the complexity of finding an optimal joint policy is NEXP-complete (Bernstein, Zilberstein, and Immerman 2000).

In the vein of simplifying the problem directly, providing an agent with observation histories, either via the game itself or free communication between agents, can allow for scenarios to be posed as single-agent POMDPs (Pynadath and Tambe 2002), which have PSPACE complexity. POMDPs have had considerably more advances than their decentralized counterparts and are frequently solved via dynamic programming (Barto 1998) or sample-based techniques (Silver and Veness 2010).

Ad Hoc Teams in Pursuit Domains

In order to cope with this complexity problem, much of the existing work in ad hoc multi-agent teams has assumed that unknown teammates are non-recursive, typically either by being best-response agents (Stone, Kaminka, and Rosenschein 2010) or working under some designed static model (Barrett, Stone, and Kraus 2011). Barrett et al. 2011 utilize a Bayesian update over known, static models to identify likely models and plan accordingly. More recently, it has been shown that ad hoc agents can benefit from learning from experiences with initial teams, then using that knowledge when collaborating with a new team (Barrett et al. 2012).

Problem Description

This paper considers the problem of coordinating with an unknown teammate, whose goal and corresponding behavior is uncertain. Typically, such hidden information is static in nature, and inference strategies follow traditional probabilistic reasoning. This paper addresses the possibility that the hidden information changes periodically. For example, if a team of agents is pursuing group of fleeing agents and one of the teammates changed targets, the remaining member should identify the change and alter its behavior accordingly in order to cooperate effectively. However, we also assume that the precise mechanism of such an evolving system is unknown, that is, the agent has no cognitive model for predicting how and when its teammate changes its behavior.

Representation

From a support agent's perspective, the problem of uncertainty in a teammate's current intent can be conceptualized as a single-agent partially observable Markov decision process (POMDP) (Kaelbling, Littman, and Cassandra 1998). For convenience and consistency, we will adopt the representation of a POMDP while discussing belief distributions as applied to MCTS. Furthermore, in this paper, we adopt a few assumptions regarding an unknown, observed agent for clarity. First, the agent acts intentionally toward one goal at a time. Secondly, the agent acts in a primarily deterministic, non-recursive manner, meaning the agent's model of the teammate does not possess nested recursive models of the agent's beliefs, the agent's beliefs regarding the teammate, and so on from the teammate's point of view.

In order to plan successfully in an POMDP, agents must utilize a sequence of observations to update their beliefs over time. As we define observations to be derived from actions taken in the game, a history can be defined as $h_t = \langle a_1, a_2, ...a_t \rangle$. Beliefs are then described as a distribution over possible states given those histories, or $b_t = Pr(s|h_t) \quad \forall s \in S$. The goal of planning in a POMDP is to find a policy π maximizing the agent's expected reward, as given by $J^{\pi}(b_0) = \sum_{t=0}^{\infty} E[R(s_t, a_t)|b_0, \pi]$. Solving POMDPs with a large number of states is intractable in many settings; therefore, we use a sampling based method, Monte-Carlo Tree Search, for an approximate solution. Our implementation is described in the next section.

Planning

Monte-Carlo Tree Search is a search algorithm based on Monte-Carlo simulations within a sequential game. Through evaluating potential states by averaging simulation outcomes, MCTS holds several advantages over other search methods. By sampling, it examines a subset of the complete state space, yet it asymptotically converges to an optimal policy in fully observable and partially observable problems given an appropriate exploration function (Silver and Veness 2010).

Potential actions at each node are selected in a fashion that balances exploration and exploitation. The idea behind such a heuristic is to progress deeper into the game tree through actions with higher expected value (exploitation) while periodically trying less favorable moves that may have unseen value (exploration). We use Upper Confidence Bounds applied to Trees (UCT), a popular, efficient algorithm for guiding MCTS (Kocsis and Szepesvári 2006). Our implementation differs from traditional POMDP planners by employing a novel adjustment to traditional Bayesian-style belief revision.

Updating Beliefs

An important aspect of planning in a partially observable scenario is the ability to refine a set of beliefs regarding the current world state. This is completed through inference after observing an aspect of the world. As by definition, a POMDP is in part defined by a set of probabilities for observations made in each potential state. Traditionally, beliefs are revised using the observation history and Bayes Theorem:

$$P_t(s_i|o) = P_{t-1}(s_i) \times \frac{P_{t-1}(o|s_i)}{\sum_j P_{t-1}(s_j) \times P_{t-1}(o|s_j)} \quad (1)$$

where $P_t(s_i|o)$ is the probability at time t of being in state $s \in S$ given the observation $o \in \Omega$ of action $a \in A$.

Working with teammate whose action selection mechanism is unknown to an agent, we must consider an approximation to the observation probabilities of certain actions, as the likelihoods are not explicit. If our set of potential models included the agent's utility for states associated with its preferred goal, we could use $P(o|s) \propto e^{V_{a,i}}$, as suggested by (Ito, Pynadath, and Marsella 2007), where V is the utility of action $a \in A$ for agent i. However, as neither utility values nor the action probabilities for a teammate are explicit in ad hoc settings, we estimate the probabilities with a simple exponentiated loss function, which is considered equivalent to

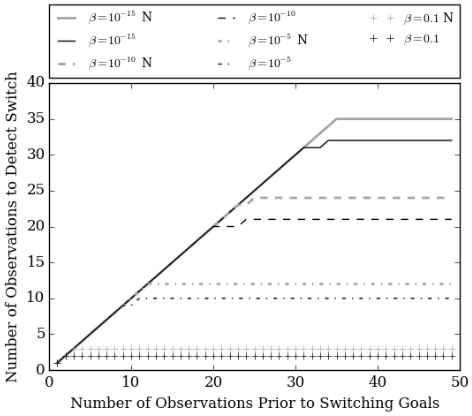

Figure 1: Number of ideal observations (those providing 0 loss only to the appropriate goal) required for the belief distribution to align with the corresponding goal as a function of the number of observations supporting the prior goal. Lines denoted with N correspond to the normalized belief revision approach. All remaining cases are not normalized.

a Bayes rule update in certain contexts (Bousquet and Warmuth 2003). Here, we define our loss function, L_i, to be 0 if the model working toward goal i predicts the observed action, o, and 1 otherwise.

$$P_t(s_i|o) = P_{t-1}(s_i) \times \frac{e^{-L_i}}{\sum_j P_{t-1}(s_j) \times e^{-L_j}} \tag{2}$$

Furthermore, as the concept of an agent with shifting priorities has natural similarities to shifting experts/online-learning problems, we borrow the concept of modifying our update step by additionally adding a mix of past posteriors, as described in (Bousquet and Warmuth 2003). By mixing the updated belief vector with the initial belief vector, we are able to enforce upper and lower bounds on the possible values of the agent's belief probabilities. This ensures the capacity for an unlikely target to surpass the most likely target quickly given a small number of appropriate observations. Equation 3 shows the mixing alteration, which is then normalized in Equation 4.

$$P_t^*(s_i|o) = \beta P_{t=0}(s_i) + (1-\beta)\frac{P_{t-1}(s_i) \times e^{-L_i}}{\sum_j P_{t-1}(s_j) \times e^{-L_j}} \tag{3}$$

$$P_t(s_i|o) = \frac{P_t^*(s_i|o)}{\sum_j P_t^*(s_j|o)} \tag{4}$$

The mixing parameter is constrained by $0 \leq \beta \leq 1$. When $\beta = 0$, the updated probability takes the form of Bayes rule, while a value of 1 results in a static probability equal to the initial probability assigned. In this context, selection of β is dependent on the noise encountered within a scenario and is outlined in the next section.

Parameter Tuning

The selection of an appropriate value of β is crucial for identifying hidden state changes. With a belief update that does not account for this need, as is the case for the traditional belief revision approach, identification of hidden state transitions can require long sequences of observations. In fact, in the ideal case, with observations only supporting the true underlying state, identification of a goal switch is linearly dependent on the number of observations supporting a previous goal. This is particularly undesirable for domains with hundreds or thousands of observations before a switch occurs.

The alteration proposed in Equation 4 does not have a closed form solution for the number of steps required for a state switch to be identified. However, empirically we observe that tuning of β creates an upper bound for the number of steps required under ideal observations. Figure 1 depicts the effect of various tested β values under thirty potential goals and perfect observations. Despite an increase in the number of steps a teammate pursues the first goal, the required number of observations supporting a second goal to converge to the appropriate belief is bounded by a finite value.

Balancing Noise and Responsiveness

Given that the modified belief revision approach can bound the number of required observations to any arbitrary number, selecting a value for β has the tradeoff between responsiveness and susceptibility to noise. A series of inaccurate observations or observations of actions by an agent imperfect in its pursuit of a goal can lead belief convergence to the wrong target goal. Tuning β is dependent on the likelihood of such observations in the domain tested. We discuss here a tuning strategy for a static noise rate.

Behavior Noise We define noisy observations as those supporting any subset of goals not including the true underlying goal currently being pursued by an agent. A worst case, then, occurs when observations support exactly one incorrect goal, that is $L_i = 0$ for an incorrect goal i and $L_j = 1$ for $\forall j \in S, j \neq i$. If multiple successive noisy observations occur supporting a single incorrect goal, the belief distribution can converge to the incorrect state.

Consider the case where a domain has a static noise rate r. The probability of a number of successive noisy observations, K, forms a geometric distribution with $P(K = k) = r^k(1-r)$. The expected length of a sequence of noisy observations, then, is given by $E[K] = \frac{1}{1-r}$. It is reasonable to constrain β such that the number of required observations to identify a switch in underlying state to $n \geq \left\lceil \frac{1}{1-r} \right\rceil$.

Selecting β Due to the normalization of probabilities in both Equations 3 and 4, choice of β depends on the noise rate tolerance as well as the number of alternative goals, as normalization considers the probabilities of all possible goals involved. Such normalization is necessary for decision theoretic reasoning, as the sum of probabilities of underlying states must sum to one. The unnormalized version, given by Equation 5, is useful for approximation of the normalized

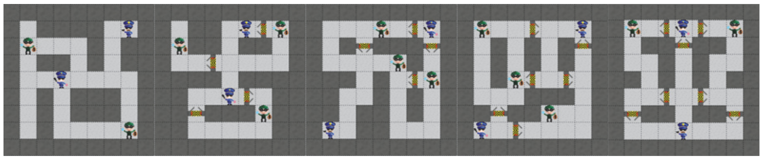

Figure 2: Mazes in Cops and Robbers, labeled *a-e*. Images from (Macindoe, Kaelbling, and Lozano-Pérez 2012).

case. Figure 1 depicts a comparison of the two version, with N denoting the normalized cases.

$$P_t(s_i|o) = \beta P_0(s_i) + (1-\beta)P_{t-1}(s_i) \times e^{-L_i} \tag{5}$$

$$P_t(s_i|o) = \beta P_0(s_i) + \beta P_0(s_i) \sum_{m=1}^{t-1} (1-\beta)^m e^{-\sum_{p=t-m}^{t-1} L_{i,p}} + (1-\beta)^t P_0(s_i) e^{-\sum_{p=1}^{t-1} L_{i,p}} \tag{6}$$

The expanded case, represented by Equation 6, allows for the direct calculation of the upper and lower bounds of probability associated with state s_i. By expanding as $t \to \infty$ and $L_t = 1 \, \forall t$, the result settles at $P_{lower}(s_i) = \frac{\beta P_0(s_i)}{1-e^{-1}(1-\beta)}$. Similarly, expanding with loss $L_j = 0 \, \forall j, j \neq i$ yields the upperbound $P_{upper}(s_i) = P_0(s_i)$. With the bounds established, we can relate the number of steps required for a state with minimal probability to succeed one with maximal probability. Assuming $P(s_i)$ is the lower bound at time t and $P(s_j)$ is the upper bound, which is the maximal separation, we wish to find a lower bound for β that guarantees $P(s_i) > P(s_j)$ after n observations in support of state s_i.

Evaluation

To evaluate our approach, we test various belief convergence strategies within a two member team version of the pursuit domain, Cops and Robbers.

Cops and Robbers

The version of Cops and Robbers used in this paper was first introduced in (Macindoe, Kaelbling, and Lozano-Pérez 2012). It is a form of the popular multi-agent pursuit scenario (Benda 1985) designed for teams consisting of two members. Figure 2 shows the five tested mazes, *a-e*, differing in layout, the number of robbers, and the inclusion of one-way doors, which can punish poor action selection by lengthening paths to targets as well as by trapping agents, as in maze *b*. Coordination is key, as an incorrect prediction of teammate behavior may allow a robber to slip by one of the agents and flee.

The domain proves challenging for multi-agent planning due to the size of its state space. For the mazes tested, the possible reachable states range from 1.6 million to 104 million with $0.282\% \pm 0.123\%$ being successful capture states. This size is ideal for an initial exploration, as it is too large for optimal solution online yet small enough to avoid any domain engineering that may muddle the comparison of approaches.

Two notable works exist in this and a similar domain. Macindoe et al. 2012 introduced Cops and Robbers as a domain for testing sequential planning for assistive agents; however, the teammate agent in the evaluation chose a single target at the start and never switched for the duration of the game. Nguyen et al. 2011 previously used a similar game, Collaborative Ghostbuster, and modeled the choice of target as a Markov decision process, with transition probabilities dependent on the resulting score of pursuing that target.

Agents

For our tests, we implemented three teammates whose goal remains uncertain to the agents.

- A* Greedy - This agent pursues the closest robber at the start of the game and never switches targets.
- Switch Once - This agent switches targets at a fixed point in the game, on the eighth turn.
- Probabilistic - This agent can potentially switch multiple times. On each turn, the probability of switching is given by

$$p = 0.2 \times \frac{distance(target)}{\sum_{r \in robbers} distance(r)} \times |robbers| \tag{7}$$

All teammates move toward their selected target using A* path planning, with 10% noise in their actions. A noisy action is randomly selected from the set of possible actions that would not pursue the active target.

For the ad hoc agent, we implement several MCTS-UCT agents, varied by how they model the unknown teammate:

- UCT - Explores both agents' actions with UCT. This agent assumes its teammate will plan and behave in an identical manner to itself. It expects the sidekick to pursue the best target as identified by the tree search and in the manner the tree search dictates.
- Bayes - Plans using UCT for its own actions but uses a belief distribution over single-target A* agents for each possible teammate goal. Updates beliefs according to Equation 2.
- RAPID - Similar to Bayes, but updates with modified belief update technique in Equation 4.
- Limited oracle - Knows the true target at each turn, even after switches occur. However, it does not have prior knowledge of when switches will occur. This agent plans with perfect knowledge of the *current* target.

Tests

Each pair of teammate and reasoning agent participate in one hundred trials of each maze. Steps taken to complete the game, beliefs of applicable agents, and targets of the teammates are logged for analysis. We allow each UCT-based agent one hundred game simulations per turn, with root parallelization (Chaslot, Winands, and van Den Herik 2008) across four cores.

To emphasize the effect of tuning, we use two versions of our RAPID agent, each with a different value for β. Given the 10% noise rate in our experiment and the geometric distribution of expected successive noisy actions, we observe that 99.9% of groups of successive noisy actions are of length 2 or fewer. Choosing $n = 3$, then, gives us a lower bound for choice of $\beta = 0.016$ from Theorem 2. For a less conservative tuning, the remaining RAPID agent uses $\beta = 0.85$ for enhanced responsiveness at the cost of noise susceptibility.

Results

We compare the performance of the RAPID agent against the agent which revises its beliefs with a traditional Bayes update. The plain UCT agent provides the base level of performance we would expect with any of the UCT-based agents, while the limited oracle agent demonstrates that there may still be room for further improvement. It should be noted that the results of the limited oracle agent could be unattainable, as the agent has access to the teammate's true target at each turn.

Belief Recovery

Table 1 reports the number of times in 100 trials the teammate switched targets as well as the average steps required for the agents to identify the change. The base UCT and limited oracle agents are omitted as they do not possess a belief system. The A* Greedy teammate is similarly absent as it never switched targets.

With respect to belief recovery time, the RAPID agent with the conservative tuning of β only outperforms the Bayes agent in one of ten test cases ($\alpha = 0.01$). Between noisy actions and those supporting potentially two or more targets, the RAPID ($\beta = 0.016$) agent could not utilize the bounded convergence time to significant effect.

The second RAPID agent, however, outperforms the Bayes agent in six of the ten relevant test cases. In these instances, the agent was able to detect a switch faster on average than the Bayes agent, resulting in an improvement of nearly seven turns in one test case.

Accuracy

With a shorter time to converge to a pursued goal, it is natural to expect an increase in accuracy of the predicted goal. For this metric, steps where the correct target probability is equal to that of another target are considered ambiguous and are counted as an incorrect identification. This explains a portion of the low observed accuracies, particularly as the first few steps in each game are not enough to distinguish targets.

With regard to overall accuracy, the RAPID agents were found to be correct more frequently in the majority of scenarios. Both β levels had significant accuracy improvements over Bayes in eight test cases each. The Bayes agent outperforms the $\beta = 0.016$ agent in two instances and the $\beta = 0.85$ agent in four instances, as seen in Table 2. This loss of accuracy in the higher β value, particularly in cases shown to have significantly shorter belief recovery periods, demonstrates the susceptibility to noise, as was expected in the tuning of β.

Steps Taken

Table 3 shows the average number of turns required to complete each test case. The less responsive of the RAPID agents had significant improvements over the Bayes agent in five of the fifteen test cases. Furthermore, it no test cases did it perform significantly worse. A higher β value, having reduced belief correction time and improved accuracy, resulted in coordination time improvements in nine test cases. The Bayes agent only achieved a higher average score than the $\beta = 0.85$ agent in one case. Additionally, results for the remaining tested agents are included for comparison. The vanilla UCT agent, which assumes identical recursive planning on the part of its teammate, demonstrates the benefit of accurate modeling, as it performed worse in every scenario than any other tested agent.

Discussion

We have outlined a computationally lightweight alteration to the traditional belief revision approach. This modification is not limited to any specific planning paradigm but can be applied to any existing approach with similar belief representations. Planning under our proposed changes to belief revisions allows an agent to quickly recognize and adapt to altered behavior indicative of a goal switch. Faster belief convergence to the correct goal boosts overall accuracy of the agent's predictions, which are directly leveraged in planning for improved coordination. Secondly, this initial empirical evidence suggests that reasoning quickly over a set of independent models may provide an acceptable approximation to modeling higher level reasoning of an unknown teammate. A predictive mechanism for goal changes is absent from our agent's model of its ad hoc teammate, as the agent possesses no knowledge regarding how an unknown teammate chooses its target or revises its plan. Rather, the agent's model's set of potential behaviors serve as an approximation to a complete cognitive model when the agent can quickly identify changes in such behavior. Human-agent teamwork is one potential application of this responsive adaptation. If an agent is to assist a human in an environment that has clear potential goals and corresponding behaviors, our approach may prove advantageous. It is likely easier to design predictive models for simple goals, compared to more complex cognitive models.

References

Agmon, N.; Barrett, S.; and Stone, P. 2014. Modeling uncertainty in leading ad hoc teams. In *Proceedings of the*

		Bayes		$\beta = 0.016$			$\beta = 0.85$		
	Teammate	n	Average	n	Average	p	n	Average	p
a	SwitchOnce	100	5.04	96	**3.76**	<0.001	100	**1.00**	<0.001
	Probabilistic	269	4.42	400	5.14	0.096	363	**2.78**	<0.001
b	SwitchOnce	99	18.04	92	18.40	0.457	92	23.30	0.079
	Probabilistic	369	12.96	326	17.83	<0.001	472	11.72	0.145
c	SwitchOnce	94	7.57	66	6.53	0.248	67	9.06	0.221
	Probabilistic	454	12.92	502	14.25	0.128	356	**8.10**	<0.001
d	SwitchOnce	100	15.87	100	14.55	0.347	100	11.75	0.085
	Probabilistic	557	15.48	644	14.19	0.133	532	**9.45**	<0.001
e	SwitchOnce	100	18.7	61	18.18	0.443	100	**11.79**	0.007
	Probabilistic	506	8.85	356	7.86	0.132	396	**6.09**	<0.001

Table 1: Average actions observed before sidekick's true target is most likely in agent's belief distribution. Bold values indicate significant results over the Bayes agent ($\alpha = 0.01$). The rows divide results according to the maze tested, as indicated by a-e. n is the number of goal switches observed in each test set.

		Bayes		$\beta = 0.016$			$\beta = 0.85$		
	Teammate	n	% Correct	n	% Correct	p	n	% Correct	p
	A*	2188	17.69	2226	17.83	0.448	1617	**23.69**	<0.001
a	SwitchOnce	3201	71.88	3333	**78.97**	<0.001	2794	**80.96**	<0.001
	Probabilistic	2159	57.94	2634	**62.34**	<0.001	2476	**64.01**	<0.001
	A*	3792	26.85	3796	**48.97**	<0.001	4181	25.52	0.089
b	SwitchOnce	3771	39.30	4574	**46.55**	<0.001	5100	34.27	<0.001
	Probabilistic	3712	33.14	4280	30.72	0.010	4029	**38.07**	<0.001
	A*	2671	40.43	2406	22.94	<0.001	2522	33.51	<0.001
c	SwitchOnce	3472	60.77	2521	52.52	<0.001	2689	51.95	<0.001
	Probabilistic	3533	42.37	3960	**46.31**	<0.001	2763	**47.09**	<0.001
	A*	2516	14.63	2708	**31.50**	<0.001	2962	**26.00**	<0.001
d	SwitchOnce	6358	49.53	5223	**57.15**	<0.001	4927	48.81	0.227
	Probabilistic	4412	45.42	5157	**50.69**	<0.001	5048	**57.81**	<0.001
	A*	2527	14.88	1356	15.71	0.245	1939	13.67	0.125
e	SwitchOnce	4480	35.71	2420	32.73	0.006	3562	**38.63**	0.004
	Probabilistic	3454	40.56	2536	42.59	0.058	2885	35.94	<0.001

Table 2: Percentage of steps with correct target identified by belief distribution. Bold values indicate significant results over the Bayes agent ($\alpha = 0.01$). n is the total number of steps across each test set.

		Bayes	$\beta = 0.016$		$\beta = 0.85$		UCT	Ltd Oracle
	Teammate	$\overline{steps}$	$\overline{steps}$	p	$\overline{steps}$	p	$\overline{steps}$	$\overline{steps}$
	A*	21.88	22.26	0.323	**16.17**	<0.001	51.95	31.97
a	SwitchOnce	32.01	33.33	0.480	**27.94**	0.005	71.07	18.74
	Probabilistic	21.59	26.34	0.063	24.76	0.006	64.89	25.91
	A*	37.92	37.96	0.125	41.81	0.460	57.76	54.89
b	SwitchOnce	37.71	45.74	0.164	51.00	0.010	61.81	49.83
	Probabilistic	37.12	42.8	0.171	40.29	0.378	56.6	41.48
	A*	26.71	**24.06**	<0.001	**25.22**	0.004	58.11	28.74
c	SwitchOnce	34.72	**25.21**	<0.001	**26.89**	<0.001	67.58	35.88
	Probabilistic	35.33	39.60	0.309	**27.63**	0.002	71.78	31.74
	A*	25.16	27.08	0.174	29.62	0.068	69.65	39.94
d	SwitchOnce	63.58	52.23	0.010	**49.27**	0.001	84.08	75.97
	Probabilistic	44.12	51.57	0.058	50.48	0.043	81.05	42.39
	A*	25.27	**13.56**	<0.001	**19.39**	0.004	62.08	11.38
e	SwitchOnce	44.80	**24.20**	<0.001	**35.62**	<0.001	81.24	20.17
	Probabilistic	34.54	**25.36**	<0.001	**28.85**	0.009	73.21	19.7

Table 3: Average steps taken by the agent/teammate pair to complete the maze. Bold values indicate significant improvement (as indicated by the p-value) over the Bayes agent ($\alpha = 0.01$).

2014 international conference on Autonomous agents and multi-agent systems, 397–404. International Foundation for Autonomous Agents and Multiagent Systems.

Barrett, S.; Stone, P.; Kraus, S.; and Rosenfeld, A. 2012. Learning teammate models for ad hoc teamwork. In *AAMAS Adaptive Learning Agents (ALA) Workshop*.

Barrett, S.; Stone, P.; Kraus, S.; and Rosenfeld, A. 2013. Teamwork with limited knowledge of teammates. In *AAAI*.

Barrett, S.; Stone, P.; and Kraus, S. 2011. Empirical evaluation of ad hoc teamwork in the pursuit domain. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 567–574. International Foundation for Autonomous Agents and Multiagent Systems.

Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT press.

Benda, M. 1985. On optimal cooperation of knowledge sources. *Technical Report BCS-G2010-28*.

Bernstein, D. S.; Zilberstein, S.; and Immerman, N. 2000. The complexity of decentralized control of markov decision processes. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, 32–37. Morgan Kaufmann Publishers Inc.

Blum, A., and Monsour, Y. 2007. Learning, regret minimization, and equilibria. In N. Nisan, T. Roughgarden, E. T., and Vazirani, V., eds., *Algorithmic Game Theory*. Cambridge University Press.

Bousquet, O., and Warmuth, M. K. 2003. Tracking a small set of experts by mixing past posteriors. *The Journal of Machine Learning Research* 3:363–396.

Chaslot, G. M.-B.; Winands, M. H.; and van Den Herik, H. J. 2008. Parallel monte-carlo tree search. In *Computers and Games*. Springer. 60–71.

Ito, J. Y.; Pynadath, D. V.; and Marsella, S. C. 2007. A decision-theoretic approach to evaluating posterior probabilities of mental models. In *AAAI-07 workshop on plan, activity, and intent recognition*.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101(1):99–134.

Kocsis, L., and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006*. Springer. 282–293.

Macindoe, O.; Kaelbling, L. P.; and Lozano-Pérez, T. 2012. Pomcop: Belief space planning for sidekicks in cooperative games. In *AIIDE*.

Nair, R.; Tambe, M.; Yokoo, M.; Pynadath, D.; and Marsella, S. 2003. Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In *IJCAI*, 705–711.

Nguyen, T.-H. D.; Hsu, D.; Lee, W. S.; Leong, T.-Y.; Kaelbling, L. P.; Lozano-Perez, T.; and Grant, A. H. 2011. Capir: Collaborative action planning with intention recognition. In *AIIDE*.

Pynadath, D. V., and Tambe, M. 2002. Multiagent teamwork: Analyzing the optimality and complexity of key theories and models. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, 873–880. ACM.

Silver, D., and Veness, J. 2010. Monte-carlo planning in large pomdps. In *Advances in Neural Information Processing Systems*, 2164–2172.

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

Stone, P.; Kaminka, G. A.; and Rosenschein, J. S. 2010. Leading a best-response teammate in an ad hoc team. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*. Springer. 132–146.

Whiten, A. 1991. *Natural theories of mind: Evolution, development and simulation of everyday mindreading*. Basil Blackwell Oxford.