RAPID: A Belief Convergence Strategy for Collaborating with Inconsistent Agents

Trevor Sarratt and **Arnav Jhala**
University of California Santa Cruz
{tsarratt, jhala}@soe.ucsc.edu

Abstract

Maintaining an accurate set of beliefs in a partially observable scenario, particularly with respect to other agents operating in the same space, is a vital aspect of multiagent planning. We analyze how the beliefs of an agent can be updated for fast adaptivity to changes in the behavior of an unknown teammate. The main contribution of this paper is the empirical evaluation of an agent cooperating with a teammate whose goals change periodically. We test our approach in a collaborative multiagent domain where identification of goals is necessary for successful completion. The belief revision technique we propose outperforms the traditional approach in a majority of test cases. Additionally, our results suggest the ability to approximate a higher level model by utilizing a belief distribution over a set of lower level behaviors, particularly when the belief update strategy identifies changes in the behavior in a responsive manner.

Introduction

In this paper, we present a new approach, called Responsive Action Planning with Intention Detection (RAPID), for updating beliefs over agent goals with fast adaptation to changes. It is often inaccurate to assume a teammate will stick to a single goal throughout a game, 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 team 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, we extend existing ideas of approaching the problem, namely by planning in a partially observable space with a set of beliefs. However, we alter the belief update protocol such that potential alternative goals are kept at relevant weights. This approach contrasts with existing belief space planning approaches where each action toward a particular goal incurs a multiplicative update in order to maximally separate the current task's belief weight from the remaining tasks' weights.

As in previous work, RAPID models the planning space as a partially observable Markov decision process (POMDP). POMDPs provide a decision theoretic basis for evaluating policies within the constraints of a world described by states through which actions can cause transitions to new states. Partial observability occurs through the masking of states, leading to uncertainty in how planned actions will effect to overall game state. In this paper, we restrict the partial observability to the teammate's model, 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 may adopt a new plan on the fly according to his or her own preferences, with new goals represented by various noisy models by a collaborative agent. When enough observed evidence favors switching to a new goal, the agent can adapt quickly and plan accordingly.

Due to the complexity of planning in POMDPs, which is PSPACE-complete for finding optimal solutions given a finite horizon, we adopt an approximate approach using Monte-Carlo tree search (MCTS). MCTS in an online tree search algorithm that uses asymmetric tree growth to explore promising action policies while minimizing time spent in irrelevant parts of the search space. As a scalable, anytime algorithm, it has in recent years been popularly applied to multiagent domains with large state spaces, of which one is ad hoc collaboration.

Planning for collaborative action is challenging, particularly under the uncertainty of teammate models. Convergence to the correct goal of a teammate is a vital aspect of multiagent planning. We propose a computationally lightweight alteration to the traditional belief revision approach, which improves the adaptability of agents when coordinating with teammates of non-static goals. This modification is not limited to sampling-based planners but can be applied to existing approaches with similar belief representations. We outline in detail both the problem space and the mechanisms behind our approach, then evaluate using a known example domain, Cops and Robbers. Finally, we discuss extensions of this work as well as applications to human-agent teams.

Related Work

Multiagent systems is a field comprised of a breadth of well-studied topics. In this section, we draw on some of the existing work motivating the problem of coordinating in ad hoc settings.

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).

Nair et al. (2003) propose fixing teammate policies and searching for locally optimal agent policies until an equilibrium is reached, resulting in a significant reduction in computational time. In the vein of simplifying the problem directly, providing an agent with action and 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 multiagent 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). Though some recent work has begun to address recursive modeling in this area (Agmon, Barrett, and Stone 2014), we will leverage the simpler assumption of non-recursive teammates for this paper while we explore adaptability through belief revision.

This paper has direct motivation from the Barrett et al. (2011) work in discerning team models in the pursuit domain. The authors 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). This is further enhanced by employing transfer learning to generalize that knowledge to a small number of observations with a new team (Barrett et al. 2013).

The three previous papers were evaluated in the pursuit domain, where a team of agents must coordinate to capture a prey on a toroidal grid (Stone and Veloso 2000). As significant work exists in this specific domain, we adapt a modified version suited for experimentation in non-static teammate models.

POMDPs

In the case where an agent's potential goals are finite and intentionally acted on in a sequential manner, the problem of uncertainty in an agent'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 it does not take into account the possible actions of other agents in the scenario.

A POMDP is a generalization of an Markov decision process (MDP) where some aspect of the world state is not directly observable to an agent. The partial observability, in this context, is constrained to the current goal the observed agent is working toward. In this framework, a POMDP can be represented as a tuple $\langle S, A, \Omega, B, R \rangle$, where

- S is the set of world states, comprising of both the game state and the current goal of the observed agent. Due to the latter uncertainty, an agent cannot directly observe the world state.
- A is the set of actions available to the agents.
- Ω defines a set of observations on actions occurring in the game. In this case, we allow for all actions to be observable at all times.
- B describes the set of belief states possible given an initial state and a set of observed actions.
- R relates the common definition of a reward function in MDPs, where $R : S \times A \rightarrow \mathbb{R}$.

In order to plan successfully in an POMDP, agents must utilize an observation history 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 $J^\pi(b_0) = \sum_{t=0}^{\infty} E[R(s_t, a_t)|b_0, \pi]$. This can be done by searching through the belief space to identify the optimal value function, $V^*(b_t) = \max_{a \in A} \left[R(b_t, a) + \sum_{o \in O} \Omega(o|b_t, a) V^*(b_{t+1}) \right]$, where Ω is a probability distribution over observations. 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.

Monte-Carlo Tree Search

Monte-Carlo Tree Search is a search algorithm based on Monte-Carlo simulations within a sequential game. Through

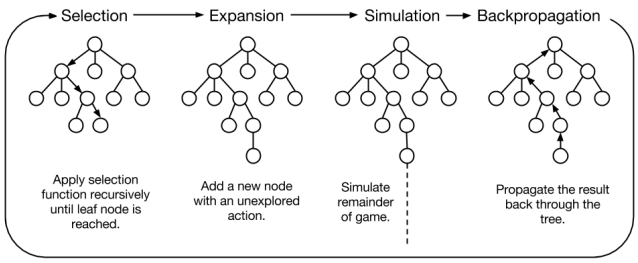

Figure 1: Outline of Monte-Carlo Tree Search

evaluating potential states by averaging simulation outcomes, MCTS holds several advantages over other search methods. By sampling, it bypasses the curse of dimensionality of large numbers of state transitions. Black box simulations can be used for problems too complex to represent fully, and it can be used effectively without prior domain knowledge (Kocsis and Szepesvári 2006). In addition, it converges to an optimal policy in fully observable and partially observable problems given an appropriate exploration function (Silver and Veness 2010) while also being an anytime approach.

MCTS performs a large number of simulations of a game, from the current state to the end of the game. As a new simulation begins searching through the game tree, MCTS considers information gathered from previous playthroughs. Specifically each step of the game, MCTS selects the next action with a bias toward those with a higher success rate for the agent. When an action is taken for the first time, the rest of the game is played out randomly. The result is then back-propagated through the deliberately explored nodes, in this case, just the root. Over many simulations, the program focuses on better moves, leading to farther look-ahead without giving as much consideration to inferior moves.

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).

For this paper, we employ a modified version of UCT for planning in POMDPs, similar to (Silver and Veness 2010). Our implementation differs in that instead of utilizing a particle filter for updating the agent's beliefs, we use a novel adjustment to traditional Bayesian-style update to adjust belief probabilities. The specific implementation details are discussed in next section.

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 some aspect of the world or action of an agent. 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:

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

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 additional value for preferences of various targets, we could use $P(o|s) \propto e^{V_{a,m}}$, as suggested by (Ito, Pynadath, and Marsella 2007). However, in order to avoid adding unnecessary complexity to our choice of models, we utilize a simple exponentiated loss function, which can be interpreted as 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 for goal i predicts the observed action, o, and 1 otherwise.

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

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). This modified approach bears much resemblance to the Polynomial Weights algorithm (Blum and Monsour 2007) as used in previous work in ad hoc teams (Barrett et al. 2012). In the latter approach, using a polynomial weight slows the belief convergence to a particular teammate model such that no model is discarded prematurely; however, given a sufficiently long series of observations supporting one model, the probabilities can still diverge sufficiently to prohibit a change in beliefs in a reasonable amount of time. In contrast, by mixing the updated belief vector with the initial uniform 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.

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

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

Cops and Robbers, first introduced in (Macindoe, Kaelbling, and Lozano-Pérez 2012), is a form of the popular multiagent pursuit scenario (Benda 1985) designed for teams consisting

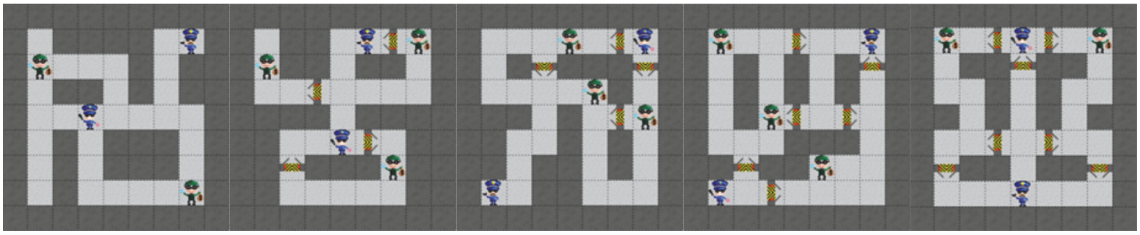

Figure 2: Mazes in Cops and Robbers, labeled *a-e*. Images from (Macindoe, Kaelbling, and Lozano-Pérez 2012).

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*. In order for the agents, "cops" in this domain, to successfully complete the game, they must coordinate to simultaneously be present within the cell of a targeted robber, at which point they receive a score of $100 - stepsTaken$, where $stepsTaken$ is the number of rounds that have passed in the game. Cops and Robbers is an inversion of the traditional pursuit problem, where an agent must coordinate with three other agents to trap a single target. Here, an agent must collaborate with one teammate to capture one of the robbers, but there is uncertainty as to which is currently being pursued by the teammate.

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. The approach selects actions maximizing the expected reward of pursuing each possible target, as weighted by the belief probability of the teammate similarly pursuing the target. This strategy, however, can favor remaining near two less likely targets rather than pursue the most likely target, as pointed out in (Macindoe, Kaelbling, and Lozano-Pérez 2012).

As discussed earlier, reasoning over a set of models in a pursuit setting has been explored in previous work. The unknown model in such work is static, and a Bayes-style update of the belief probabilities is often sufficient to identify the correct one effectively. This paper explores the possibility that the unknown model or goal driving a teammate's behavior is transient in nature, which poses two challenges: identifying changes in behavior quickly and coordinating to achieve that goal before another transition occurs.

Agents

For our tests, we implemented three teammates whose goal remains uncertain to the agents. The teammates behave as follows:

- 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 switches with a probability of

$$p = 0.2 \times \frac{distance(target)}{\sum_{r \in robbers} distance(r)} \times |robbers| \quad (4)$$

All teammates move toward their selected target using A* path planning, with 10% noise in their actions.

We also implement several agents for reference as well as for comparison of belief update strategy:

- UCT - plans using MCTS, exploring both agents' actions with UCT.
- Bayes - plans using UCT for its own actions but uses single-target A* for each possible teammate goal. Updates beliefs according to Equation 1.
- RAPID - Similar to Bayes, but updates with modified belief update technique in Equation 3.
- 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.

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. Furthermore, our belief update uses $\beta = 0.85$, an empirically chosen value.

Results

This section compares 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 our UCT-based agents, while the limited oracle agent demonstrates that there may still be room for further improvement in a few test cases. 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 every turn. Furthermore, as the agent has no prescient knowledge of upcoming target changes, it may be slightly more at risk of committing to a poor decision early, as experienced in mazes b and d.

		Bayes		RAPID		
	Teammate	n	Average	n	Average	p
a	SwitchOnce	100	5.04	100	**1.00**	<0.001
	Probabilistic	269	4.42	363	**2.78**	<0.001
b	SwitchOnce	99	18.04	92	23.30	0.079
	Probabilistic	369	12.96	472	11.72	0.145
c	SwitchOnce	94	7.57	67	9.06	0.221
	Probabilistic	454	12.92	356	**8.10**	<0.001
d	SwitchOnce	100	15.87	100	11.75	0.085
	Probabilistic	557	15.48	532	**9.45**	<0.001
e	SwitchOnce	100	18.7	100	**11.79**	0.007
	Probabilistic	506	8.85	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 ($\alpha = 0.01$).

Belief Recovery

Three metrics serve to evaluate our proposed approach: the average number of observations required to revise an agent's beliefs to the appropriate target, the percentage of steps in our tests in which an agent has correctly identified the target, and the average number of steps required to capture it. 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* teammate is similarly absent as it never switched targets. With respect to belief recovery time, the RAPID agent outperforms the Bayesian agent six of the ten relevant test cases ($\alpha = 0.01$). This provides direct evidence that our approach identifies changes in behavior more quickly than the traditional method.

Accuracy

In regard to overall accuracy, the RAPID agent is found to be correct more frequently in the majority of scenarios. It is only outperformed by the Bayes agent in four instances, as seen in Table 2. In 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. We note that pairings where the RAPID agent has faster belief recovery but poor overall accuracy indicate that the approach can be susceptible to noisy behavior, though this may be mitigated by tuning of the β parameter. We leave optimal tuning of the parameter to future work.

Steps Taken

Table 3 shows the average number of turns required to complete each test case. In nine of the fifteen comparisons between the traditional belief update approach and RAPID, our approach performs significantly better ($\alpha = 0.01$). Only in one of the remaining cases does the Bayes update version achieve a significantly better score. The comparisons with

		Bayes		RAPID		
	Teammate	n	% Correct	n	% Correct	p
a	A*	2188	17.69	1617	**23.69**	<0.001
	SwitchOnce	3201	71.88	2794	**80.96**	<0.001
	Probabilistic	2159	57.94	2476	**64.01**	<0.001
b	A*	3792	26.85	4181	25.52	0.089
	SwitchOnce	3771	**39.30**	5100	34.27	<0.001
	Probabilistic	3712	33.14	4029	**38.07**	<0.001
c	A*	2671	**40.43**	2522	33.51	<0.001
	SwitchOnce	3472	**60.77**	2689	51.95	<0.001
	Probabilistic	3533	42.37	2763	**47.09**	<0.001
d	A*	2516	14.63	2962	**26.00**	<0.001
	SwitchOnce	6358	49.53	4927	48.81	0.227
	Probabilistic	4412	45.42	5048	**57.81**	<0.001
e	A*	2527	14.88	1939	13.67	0.125
	SwitchOnce	4480	35.71	3562	**38.63**	0.004
	Probabilistic	3454	**40.56**	2885	35.94	<0.001

Table 2: Percentage of steps with correct target identified by belief distribution. Bold values indicate significant results ($\alpha = 0.01$).

the vanilla UCT agent and the oracle version demonstrate the benefit of modeling an uncertain teammate accurately as well as show that in some cases, room for improvement persists.

Conclusion & Future Work

Most existing work in ad hoc teamwork assumes a static behavior or goal for the unknown teammate(s). This paper introduces a variation of the pursuit domain in order to evaluate approaches to working with an unknown teammate whose goals and corresponding behavior can change periodically. 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 better multiagent 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, as long as its base goals can be inferred.

Though this paper proposes an effective technique for this problem within ad hoc teamwork, it raises many interesting questions to be examined in future work. Furthermore, some of the recent work has focused on learning a teammate's model while cooperating. Learning individual goals that an agent may work toward and cycle between would be a substantial improvement. As the current teammates tested have purely self-interested behaviors, we must consider how collaborating with a teammate with recursive modeling capacity may be affected by our technique. Another increase in complexity would result from considering teammates that pursue two or more goals simultaneously. The combinatorial number of possible subsets of goals provides a novel,

	Teammate	Bayes	RAPID	p	UCT	P.K.
a	A*	21.88	**16.17**	<0.001	51.95	31.97
	SwitchOnce	32.01	**27.94**	0.005	71.07	18.74
	Probabilistic	**21.59**	24.76	0.006	64.89	25.91
b	A*	37.92	41.81	0.460	57.76	54.89
	SwitchOnce	37.71	51.00	0.010	61.81	49.83
	Probabilistic	37.12	40.29	0.378	56.6	41.48
c	A*	26.71	**25.22**	0.004	58.11	28.74
	SwitchOnce	34.72	**26.89**	<0.001	67.58	35.88
	Probabilistic	35.33	**27.63**	0.002	71.78	31.74
d	A*	25.16	29.62	0.068	69.65	39.94
	SwitchOnce	63.58	**49.27**	0.001	84.08	75.97
	Probabilistic	44.12	50.48	0.043	81.05	42.39
e	A*	25.27	**19.39**	0.004	62.08	11.38
	SwitchOnce	44.80	**35.62**	<0.001	81.24	20.17
	Probabilistic	34.54	**28.85**	0.009	73.21	19.70

Table 3: Average steps taken by the agent/teammate pair to complete the maze. Bold values indicate significant differences ($\alpha = 0.01$) between Bayes and RAPID.

unexplored area for further research.

Human-agent teamwork is one potential application of using a set of basic goals to approximate a high level behavior. Existing work for modeling human cognition often utilizes *theory of mind* concepts (Whiten 1991) or relies on learned or hand-authored models. If an agent is to assist a human in an environment that has clear potential goals, our approach may prove advantageous. It is likely easier to design predictive models for simple goals, compared to more complex cognitive models. Furthermore, more responsive switching of tasks may be an acceptable response to the high-level decision making of the human teammate. It forgoes much computation on the larger body of tasks to be completed in favor of coordinating on the task at hand. Naturally, this puts the agent in a supporting role while a human takes the lead in prioritizing goals.

References

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

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.