Helping People On The Fly: Ad Hoc Teamwork for Human-Robot Teams

João G. Ribeiro[1,2], Miguel Faria[1,2], Alberto Sardinha[1,2], and Francisco S. Melo[1,2]

[1] INESC-ID
[2] Instituto Superior Técnico, Universidade de Lisboa

Abstract. We present the Bayesian Online Prediction for Ad hoc teamwork (BOPA), a novel algorithm for ad hoc teamwork which enables a robot to collaborate, on the fly, with human teammates without any pre-coordination protocol. Unlike previous works, BOPA relies only on state observations/transitions of the environment in order to identify the task being performed by a given teammate (without observing the teammate's actions and environment's reward signals). We evaluate BOPA in two distinct settings, namely (i) an empirical evaluation in a simulated environment with three different types of teammates, and (ii) an experimental evaluation in a real-world environment, deploying BOPA into an ad hoc robot with the goal of assisting a human teammate in completing a given task. Our results show that BOPA is effective at correctly identifying the target task, efficient at solving the correct task in optimal and near-optimal times, scalable by adapting to different problem sizes, and robust to non-optimal teammates, such as humans.

Keywords: Ad Hoc Teamwork · Multi-Agent Systems · Human-robot Collaboration

1 Introduction

As the number of robots increases in our everyday environment, many scenarios (e.g., healthcare, search-and-rescue teams, warehouse management) will require them to collaborate with humans in order to accomplish a given task. Hospitals, for example, can now count on medical robotic assistants (e.g., Terapio [11] and Robear [9]) to help nurses in tasks such as recording patients' vitals, delivering resources, and lifting patients out of bed. However, humans and robots may not be able to coordinate in advance. Hence, designing robots for these environments can be a very challenging problem, especially if you need the robot to learn how to collaborate without any pre-coordination protocol.

The research problem of collaboration without pre-coordination is known as *ad hoc teamwork* [10]. Within the robotics community, this research problem has been addressed by several robotic systems in the *drop-in player competition* at the annual RoboCup world championships [7]. The competition served as testbed for ad hoc teamwork with robots, and highlighted several important

problems that must be addressed if ad hoc teamwork is to be ported to real-world interactions. First, the ad hoc agent may not know the task it has to perform in advance, because the teammate may not explicitly communicate the task to the robot. Second, the robot may not have the capability to perceive the teammate's actions due to limited perception capabilities. Lastly, the robot may not receive any (explicit or implicit) reward signals during the interaction [6].

State-of-the-art algorithms [3,4,8] for ad hoc teamwork can, in theory, be used to allow robots to collaborate with humans on-the-fly, without any pre-coordination protocol. Unfortunately, they are not tailored for the specific challenges of human-robot collaboration identified above. For instance, PLASTIC Model [3] and PLASTIC Policy [4] rely on reward signals from the environment; other works [8] assume that a robot can observe the teammates' actions. However, these assumptions may not hold in real-world human-robot interaction settings, where the robot plays the role of "ad hoc agent" and the human plays the role of teammate.

This paper addresses the aforementioned challenges by presenting a novel approach for ad hoc teamwork. In particular, we present *Bayesian Online Prediction for Ad hoc teamwork* (BOPA), which enables a robot to learn how to collaborate on the fly with human teammates by relying only on state observations. We build on the work of Melo & Sardinha [8] but with a widely different set of assumptions. In particular, we make the following assumptions: i) there are no visible actions and the reward signals are not available; ii) the current task is described by a multi-agent Markov decision process (MMDP); iii) teammates may not always follow an optimal policy; and iv) the ad hoc agent has access to a library of possible tasks (each described as an MMDP).

In order to test our BOPA algorithm, we conducted an empirical evaluation in two different environments. The first environment is a simulation of an ad hoc robot and a human teammate in a grid world, where we evaluate the effectiveness, efficiency, scalability, and robustness of our algorithm. In the second environment, a live robot collaborates with a human teammate in order to explore uncharted areas of a map. Our empirical results, both in simulation and in a real-world scenario, show that our algorithm is not only efficient at identifying the correct task but also capable of completing all cooperative tasks without reward feedback or knowledge of human actions.

Hence, this work makes two novel contributions to the robotics community by (i) presenting the first ad hoc teamwork algorithm tailored for human-robot collaboration, together with a theoretical bound on the performance of our approach, and (ii) evaluating the ad hoc robot in order to show the effectiveness, efficiency, scalability, and robustness of our algorithm.

2 Notation and background

We resort to a *multi-agent Markov decision process* (MMDP) framework to model our tasks. An MMDP can be described as a tuple

$$\mathcal{M} = (N, \mathcal{X}, \{\mathcal{A}^n, n = 1, \ldots, N\}, \{\mathbf{P}_a, a \in \mathcal{A}\}, r, \gamma)$$

where N is the number of agents in the MMDP, $\mathcal{X}$ is the (finite) state space (we write X_t to denote the state at time step t), $\mathcal{A}^n$ is the (finite) individual action space for agent n, $n = 1, \ldots, N$. $\mathcal{A}$ is the set of all *joint actions*, i.e., $\mathcal{A} = \mathcal{A}^1 \times \mathcal{A}^2 \times \ldots \times \mathcal{A}^N$. We denote an element of $\mathcal{A}^n$ as a^n and an element of $\mathcal{A}$ as a tuple $a = (a^1, \ldots, a^N)$, with $a^n \in \mathcal{A}^n$. Similarly, we write a^{-n} to denote a *reduced joint action*, i.e., a tuple $a^{-n} = (a^1, \ldots, a^{n-1}, a^{n+1}, \ldots, a^N)$, and $\mathcal{A}^{-n}$ to denote the set of all reduced joint actions. We also write A_t, A_t^{-n} and A_t^n to denote, respectively, the joint action, a reduced joint action and the individual action of agent n at time step t. $\mathbf{P}_a$ is the transition probability matrix associated with joint action a. We usually write $\mathbf{P}(y \mid x, a)$ to denote the probability $\mathbb{P}\left[X_{t+1} = y \mid X_t = x, A_t = a\right]$. Finally, $r(x)$ denotes the reward associated with a given state x. The reward is common to all agents and translates the goal of the team as a whole. γ is a scalar *discount* such that $0 \leq \gamma < 1$.

The goal of the agents in an MMDP is to select a joint policy, π, that maximizes the total discounted reward. Letting

$$v_r^{\pi}(x) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(X_{t+1}) \mid X_0 = x\right],$$

the goal of the agents can be formulated as computing a joint policy, π_r, such that $v_r^{\pi_r}(x) \geq v_r^{\pi}(x)$ for any policy π.

3 Bayesian online prediction for ad hoc teamwork

We now formalize ad hoc teamwork as a Bayesian prediction problem.

3.1 Assumptions

The ad hoc agent is denoted as α and the teammates as a single "meta agent", denoted as $-\alpha$. We assume the teammates know the task r and follow the corresponding MMDP's optimal policy, $\pi_r^{-\alpha}$, but the ad hoc agent does not. Additionally, and unlike [2], we do not consider a reinforcement learning setting, whereby the ad hoc agent, at each step t, is actually able to observe a reward R_t resulting from the current state X_t and joint action A_t. Instead, the ad hoc agent is only able to observe, at each step t, the current state, X_t.

Finally, we assume that the ad hoc agent knows the dynamics of the world (i.e., the transition probabilities $\{\mathbf{P}_a, a \in \mathcal{A}\}$) and that the (unknown) reward r belongs to some pre-specified library of possible rewards, $\mathcal{R} = \{r_1, \ldots, r_M\}$ (which are then used to compute the MMDP's optimal policies $\pi_{r_1}, \ldots, \pi_{r_M}$). By simply observing how the state evolves through time, the ad hoc agent must infer both the task and the teammate's policy.

3.2 Preliminaries

We treat the unknown MMDP reward, r, as a random variable—henceforth denoted as R to make explicit its nature as a random variable. Let $\pi_m^{-\alpha}$ denote the optimal policy for the teammates if the $R = r_m, r_m \in \mathcal{R}$, and define

$$\mathbf{P}_m(y \mid x, a^{\alpha}) \triangleq \mathbb{P}\left[X_{t+1} = y \mid X_t = x, A_t^{\alpha} = a, A_t^{-\alpha} \sim \pi_{r_m}^{-\alpha}\right].$$

We can compute $\mathbf{P}_m(y \mid x, a^\alpha)$ as

$$\mathbf{P}_m(y \mid x, a^\alpha) = \sum_{a^{-\alpha}} \mathbf{P}(y \mid x, (a^\alpha, a^{-\alpha}))\pi_m^{-\alpha}(a^{-\alpha} \mid x). \tag{1}$$

Let p_0 denote some (prior) probability distribution over $\mathcal{R}$, with $p_0(m) = \mathbb{P}[R = r_m]$. More generally, we define

$$p_t(m) = \mathbb{P}\left[R = r_m \mid \left\{x_0, a_0^\alpha, x_1, \ldots, x_{t-1}, a_{t-1}^\alpha, x_t\right\}\right]. \tag{2}$$

From Bayes theorem,

$$p_t(m) = \frac{1}{Z} \sum_{m=1}^{M} \mathbf{P}_m(x_t \mid x_{t-1}, a_{t-1}^\alpha) p_{t-1}(m),$$

where Z is a normalization constant. Finally, for $r_m \in \mathcal{R}$, we define the MDP

$$\mathcal{M}_m = (\mathcal{X}, \mathcal{A}^\alpha, \{\mathbf{P}_{m,a^\alpha}\}, r_m, \gamma), \tag{3}$$

where the transition probabilities $\mathbf{P}_{m,a}$ are defined as in (1). The optimal policy for $\mathcal{M}_m$, henceforth denoted as π_m, is the optimal "ad hoc policy" when $R = r_m$.

3.3 Bayesian Online Prediction for Ad hoc teamwork (BOPA)

At each time step t, the ad hoc agent selects an action A_t^α in the current state, X_t. To that purpose, it may choose to follow the action prescribed by any of the optimal policies in the set $\{\pi_1, \ldots, \pi_M\}$. The agent is only able to observe the transition between states and its own action. After observing a transition (x, a^α, y), and independently of which policy is followed,

$$\mathbb{P}[(x, a^\alpha, y) \mid R = r_m] = \mathbf{P}_m(y \mid x, a^\alpha).$$

The agent can thus update its current belief over which is the target task, p_t, using (2). Given the target reward r_m, we define the loss of policy selecting action a^α at time step t given that the target task is m as

$$\ell_t(a^\alpha \mid m) = v^{\pi_m}(x_t) - q^{\pi_m}(x_t, a^\alpha),$$

where π_m is the solution to the MMDP $\mathcal{M}_m$.

It is important to note that both $v^{\pi_m}(x_t)$ and $q^{\pi_m}(x_t, a^\alpha)$ can be computed offline when solving the MMDP $\mathcal{M}_m$. Note also that $\ell_t(a^\alpha \mid m) \geq 0$ for all a^α, and $\ell_t(a^\alpha \mid m) = 0$ only if $\pi_m(a^\alpha \mid x_t) > 0$. The action for the ad hoc agent at time step t can now be computed using our Bayesian setting as

$$\pi_t(a^\alpha \mid x_t) \triangleq \mathbb{P}[A_t^\alpha = a^\alpha \mid X_t = x_t] = \sum_{m=1}^{M} \pi_m(a^\alpha \mid x_t) p_t(m). \tag{4}$$

We can derive a bound for the loss of our agent, when compared against an agent considering a distribution q over tasks. We use the following lemma [1].

Lemma 1. *Given a set of hypothesis $\mathcal{H} = \{1, \ldots, H\}$, for any measurable function $\phi : \mathcal{H} \to \mathbb{R}$ and any distributions p and q on $\mathcal{H}$,*

$$\mathbb{E}_{h\sim q}\left[\phi(h)\right] - \log \mathbb{E}_{h\sim p}\left[\exp(\phi(h))\right] \leq \mathrm{KL}(q \parallel p).$$

We want to bound the loss incurred by our agent after T time steps. Before introducing our result, we require some auxiliary notation. Let m^* be the (unknown) target task at time step t. The expected loss at time step t is

$$L_t(\pi_t) = \mathbb{E}\left[\ell_t(A^\alpha \mid m^*)\right] = \sum_{m=1}^{M} p_t(m)\ell_t(\pi_m \mid m^*),$$

where, for compactness, we wrote

$$\ell_t(\pi_m \mid m^*) = \sum_{a^\alpha \in \mathcal{A}^\alpha} \pi_m(a^\alpha \mid x_t)\ell_t(a^\alpha \mid m^*).$$

Let q denote an arbitrary distribution over $\mathcal{R}$, and define

$$L_t(q) = \sum_{m=1}^{M} q(m)\ell_t(\pi_m \mid m^*).$$

Then, setting $\phi(m) = -\eta\ell_t(\pi_m \mid m^*)$, for some $\eta > 0$, and using Lemma 1,

$$\mathbb{E}_{m\sim q}\left[\phi(m)\right] - \log \mathbb{E}_{m\sim p_t}\left[\exp(\phi(m))\right] \leq \mathrm{KL}(q \parallel p_t)$$

which is equivalent to

$$-\log \mathbb{E}_{m\sim p_t}\left[\exp(\phi(m))\right] \leq \eta L_t(q) + \mathrm{KL}(q \parallel p_t). \tag{5}$$

Noting that $-2\eta\frac{R_{\max}}{1-\gamma} \leq \phi(m) \leq 0$ and using Hoeffding's Lemma,[3] we have that

$$-\log \mathbb{E}_{m\sim p_t}\left[\exp(\phi(m))\right] \geq \eta L_t(p_t) - \frac{\eta^2 R_{\max}^2}{2(1-\gamma)^2}. \tag{6}$$

Combining (5) and (6), yields

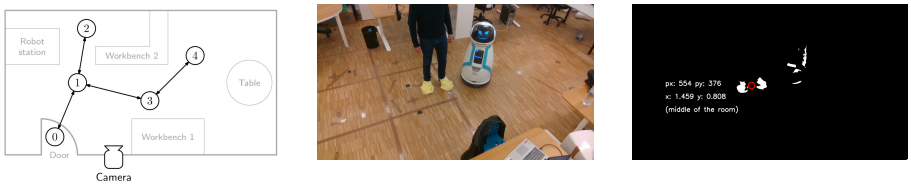
$$L_t(p_t) \leq L_t(q) + \frac{1}{\eta}\mathrm{KL}(q \parallel p_t) + \frac{\eta R_{\max}^2}{2(1-\gamma)^2}$$

which, summing for all t, yields

$$\sum_{t=0}^{T-1} L_t(p_t) \leq \sum_{t=0}^{T-1} L_t(q) + \frac{1}{\eta}\sum_{t=0}^{T-1} \mathrm{KL}(q \parallel p_t) + \frac{T\eta R_{\max}^2}{2(1-\gamma)^2}.$$

[3] Hoeffding's lemma states that, given a real-valued random variable X such that $a \leq X \leq b$ almost surely and any $\lambda \in \mathbb{R}$,

$$\mathbb{E}\left[e^{\lambda X}\right] \leq \exp\left(\lambda\mathbb{E}\left[X\right] + \frac{\lambda^2(b-a)^2}{8}\right).$$



(a) Sketch of the environment layout.

(b) RGB feed from the Intel RealSense Camera.

(c) Location using color segmentation and planar homography.

Fig. 1: Environment for ER scenario, including the environment layout, a frame where both robot and human are in position 3 (next to Workbench 1), and the segmentation and homography, used to locate the human in the environment.

Since η is arbitrarily, setting $\eta = \sqrt{\frac{T}{2}}$ leads to

$$\sum_{t=0}^{T-1} L_t(p_t) \leq \sum_{t=0}^{T-1} L_t(q) + \sqrt{\frac{2}{T}} \sum_{t=0}^{T-1} \mathrm{KL}(q \parallel p_t) + \sqrt{\frac{T}{2}} \cdot \frac{R_{\max}^2}{(1-\gamma)^2}. \tag{7}$$

Aside from the term $\sqrt{\frac{T}{2}} \cdot \frac{R_{\max}^2}{(1-\gamma)^2}$ (which grows sub-linearly with T), the bound in (7) is similar to to those reported by Banerjee for Bayesian online prediction with bounded loss [1], since

$$\sum_{t=0}^{T-1} \mathrm{KL}(q \parallel p_t) = \mathrm{KL}(\mathbf{q} \parallel \mathbf{p}_{0:T-1}),$$

where $\mathbf{q}, \mathbf{p}_{0:T-1}$ refer to distributions over sequences in $\mathcal{R}^T$.

4 Evaluation

We evaluate BOPA in two different environments, a simulated environment—*Panic Buttons*, or PB [5]—and a real world environment using a real robot as the ad hoc agent and a human as the teammate—*Environment Reckon*, or ER. PB is a benchmark grid-world environment where N agents must simultaneously press N buttons. ER is a real-world ad hoc teamwork scenario, where a human and a robot explore specific uncharted areas in the environment in Fig. 1a. The task is complete once all uncharted areas are visited.

We consider three different configurations for both scenarios, which correspond to the different tasks in the ad hoc agent's library.[4] Each environment/configuration is described as an MMDP with a distinct reward function.

[4] In the PB environment, different configurations correspond to different positions for the buttons; in the ER environment, different configurations correspond to different uncharted locations in the map.

The joint optimal policies are computed using value iteration for the underlying MDP. In both environments, the ad hoc agent observes only the state of the teammate (i.e., its position in the environment) and must infer the task (i.e., configuration) and act accordingly.

4.1 Evaluation procedure

The two scenarios are used to assess different aspects of our proposed approach. In both scenarios, the ad hoc agent can only observe the state of the MMDP, and can observe neither the teammate's actions nor any reward.

The PB scenario is used to assess the scalability, efficiency and robustness to different teammates. To the best of knowledge, our work is the first addressing ad hoc teamwork problems where the ad hoc agent has only state information available. To evaluate our approach, we compare BOPA against two baselines: an "ad hoc agent" following a random policy (named *random*), and an "ad hoc agent" following optimal policy for the task at hand (named *greedy*). The two baselines provide upper and lower bounds on the performance of BOPA.

The ER scenario, on the other hand, is used to assess the applicability of our approach in a real human-robot interaction scenario, where the state perception is not perfect, and the teammate (the human user) does not necessarily follow a pre-specified policy. We deploy our algorithm, BOPA, into a human-sized robot from our laboratory (see Fig. 1b). The position of the robot is detected using the robot localization (determined using odometry and a laser sensor). The position of the human is determined using a RealSense RGB camera (see Fig. 1b for a snapshot). The user wears high contrast shoes that are segmented from the background and used to locate the user in the room using planar homography (Fig. 1c). The human user is told beforehand the task (i.e., which locations should be visited) and asked to move between adjacent nodes at each time step and coordinate with the robot to visit the un-visited areas as quickly as possible.

4.2 Metrics

In the PB scenario, the reported values consist of averages and 95% confidence intervals over 32 independent trials, where a single trial consists of running the three agents (greedy, BOPA, and random) against an unknown teammate. To gain some additional insight regarding the robustness of our approach, we pair the ad hoc agent with different teammates—an optimal teammate, that knows the task and acts optimally; a sub-optimal teammate, that knows the task but chooses not to act with a probability 0.3, and a teammate that acts randomly.

We report four different metrics that seek to assess effectiveness, efficiency, scalability and robustness to sub-optimal teammates. Effectiveness is measured by determining whether or not BOPA is able to identify the correct task. Efficiency is measured by evaluating whether or not the ad hoc agent is able to solve the task in near-optimal time. Scalability is assessed by observing the performance of the ad hoc agent in different problem sizes (3×3, 4×4 and 5×5 grids). Robustness is evaluated by reporting whether or not an ad hoc agent is

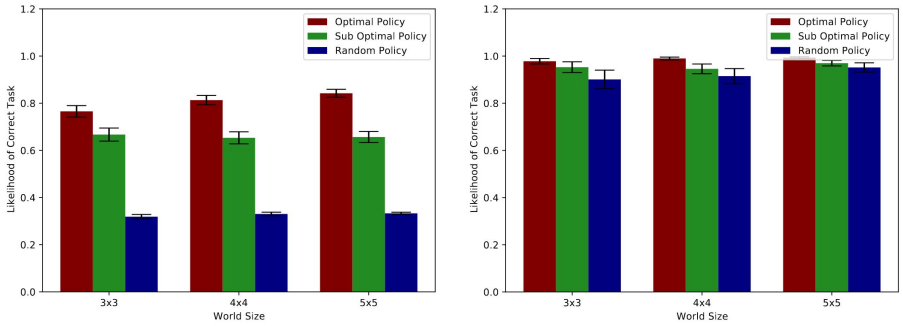


Fig. 2: Probability of correct task averaged across the whole episode, $1/T \sum_t p_t(r)$ (left) and probability of correct task, at the last step of the episode, $p_T(r)$ (right). The error bars correspond to the variability in the agent's estimate.

Table 1: Average number of steps required for task completion.

	Optimal		**Sub-optimal**		**Random**	
3×3 Greedy	2.7±	0.5	3.4±	1.2	24.2±	23.3
3×3 Bopa	3.3±	0.8	4.3±	2.3	35.9±	37.7
3×3 Random	25.2±	27.4	26.7±	28.2	144.8±	154.1
4×4 Greedy	4.0±	0.8	5.2±	1.9	54.0±	56.1
4×4 Bopa	4.7±	0.8	6.0±	2.4	61.0±	67.0
4×4 Random	47.3±	50.6	56.8±	59.2	497.4±	426.3
5×5 Greedy	5.3±	0.9	6.7±	1.6	85.7±	97.4
5×5 Bopa	5.8±	0.8	8.5±	3.7	168.8±	169.5
5×5 Random	104.7±	106.9	103.3±	109.0	1120.4±	1003.1

able to cope with non-optimal teammates. In the ER scenario, we report only the first two metrics (effectiveness and efficiency).

5 Results

We now present and discuss the results of our experiments.

5.1 PB Scenario

The results for the PB scenario are summarized in Fig. 2 and Table 1. The plots in Fig. 2 depict the ad hoc agent's ability to identify the unknown target task, r. Figure 2 (left) presents—for the different environment sizes and teammates—the likelihood of r according to the agent's belief, averaged across the whole trial,

i.e., for an episode of length T,

$$p_{\text{ave}}(r^*) = \frac{1}{T}\sum_{t=1}^{T} p_t(r^*).$$

Figure 2 (right) presents the likelihood of r^* according to the agent's belief and the final step of the trial, i.e., for an episode of length T, $p_T(r^*)$.

The plots of Fig. 2 allow us to conclusively assess BOPA's effectiveness: in all environments and for all teammates, the algorithm is able to identify the target task with great certainty. The plot also shows successfully identifying the target task largely depends on the teammate's behaviors: if the teammate behaves in a misleading way (i.e., sub-optimally), this will sometimes lead to poor belief updates, hindering the algorithm's ability to identify the target task.

In terms of BOPA's efficiency (i.e., its ability to solve the target task), we can observe in Table 1 that the performance of BOPA—when playing with an optimal teammate—closely follows that of the greedy agent (i.e., the agent knowing the target task). This is in accordance with our results on effectiveness: since BOPA is able to quickly identify the target task, it performs near-optimally in all tasks.

In terms of scalability and robustness (i.e., how BOPA's performance depends on the size of the problem and the quality of the teammates), two interesting observations are in order. On one hand, the difference in performance between BOPA and the greedy agent attenuates for larger environments. This can be understood as the larger environments provide more data (i.e., teammate's action effects through state observations) for the ad hoc agent to recognize the action and immediately head to the goal. On the other hand, the negative impact of playing with sub-optimal teammates is larger for larger environments.

To conclude, and taking all results into account, we conclude that BOPA is a robust approach to the problem of ad hoc teamwork, being able to identify the unknown task in near-optimal time even with non-optimal teammates.

5.2 ER Scenario

For the ER scenario, we provide results for each of the three task configurations. In all trials, both robot and human depart from node 0 ("Door"). In the first configuration, the uncharted areas correspond to the "Door" (node 0), "Robot station" (node 1) and the "Table" (node 4). In the second configuration, the uncharted areas correspond to nodes 1, 2, and 3 ("Robot station", "Workbench 2", and "Workbench 1", respectively). Finally, in the third configuration, the uncharted areas correspond to nodes 1, 2, and 4 ("Robot station", "Workbench 2", and "Table", respectively). The observed runs—in terms of states and agent's beliefs—are depicted in Figure 3. No mis-detections were observed (i.e., the sensors on the robot were always able to correctly locate the robot, while the camera system always correctly located the human user).

In the first run, corresponding to the first configuration, the optimal policy is for one of the agents to go towards the last unexplored node ("Table"). We can see that in the first time step, the robot had the highest uncertainty. In this turn,

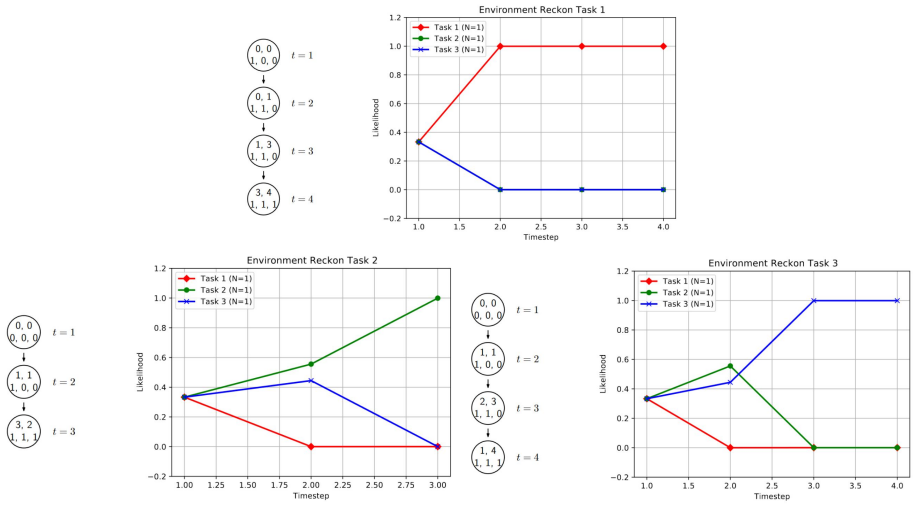


Fig. 3: Trajectories observed during the interaction with a human user (for tasks 1, 2 and 3). The diagrams on the left represent the sequence of states, (numbers on top correspond to the positions of the robot and the human, respectively, while the bits on the bottom denote whether the uncharted areas have been visited). The plots on the right depict the evolution of the robot's beliefs.

only the human user moved, to "Robot station". In turn 2, human proceeded to go towards the last unexplored node ("Table"), solving the task in optimal time. As the human moves towards this final node, the robot's belief on the target task goes up to 1.0. This first run enables two conclusions: first, there is no need for actual cooperation in this task, meaning if one of the agents is solving the task the other may do nothing. Second, unsurprisingly, BOPA successfully identified the correct task by observing the movement of the human user.

In the second run, corresponding to the second configuration, the optimal policy requires cooperation in order to be optimally solved. In the first timestep, the robot moved towards node 1. By observing this transition alone, we can see that the likelihood of the first task decreases to nearly 0.0, since visiting node 0 did not activate any of the three visitation bits. After this transition, BOPA still has some uncertainty on which task is the correct one, with the second and correct task having a likelihood of around 0.55 and the third task having a likelihood of around 0.44. This uncertainty is expected, since in both tasks, the observed transition is required in order to optimally solve them. After the third and last transition, however, where the robot moved towards node 3 and the human went to node 2, the state now indicates that all unexplored nodes have been explored, enabling BOPA to identify the correct task with 100% certainty. The task was also solved in its optimal number of steps.

The third run, corresponding to the third configuration, also requires cooperation in order to be optimally solved. In the optimal policy, both agents go

towards node 1 first and then split up, one going towards node 2 and the other towards node 4, having to pass through node 3. This task provides ambiguity with the other two, since it needs the forth node to be explored (like the first task) and the second node to be explored (like the second task).

We can see that in the first timestep, the robot had the highest uncertainty and, once again by chance, moved towards node 1 (which is considered an optimal action for all tasks). Like with the second task, by observing this first transition alone, we can see that the likelihood of the first task decreases to nearly 0.0. After this transition, BOPA has the same uncertainty it had on the previous task (which makes sense given the exact same transition), with the second and correct task having a likelihood of around 0.55 and the third task having a likelihood of around 0.44. After the second transition, however, the robot moves towards node 2 and the human moves towards node 3. Since the flags indicating whether each node has been explored are all set to one (unlike what happened in the second task), BOPA is now able to identify the correct task with 100% certainty. This final task was also concluded in its optimal number of steps.

To conclude, taking these results into account, we can see that BOPA is not only able to identify the correct task with great certainty by inferring the teammate's behavior through state observations, but also capable of adapting to non-optimal teammates by still being able to solve the tasks.

6 Conclusion and Future Work

This paper presented and evaluated the Bayesian Online Prediction Algorithm for Ad Hoc Teamwork (BOPA), a novel approach for the ad hoc teamwork problem, where an agent had to learn to cooperate with both optimal and non-optimal teammates in solving an unknown task, without being able to observe the teammates' actions and the environment's reward signals.

Having performed both an empirical evaluation in a simulated environment following the OpenAI Gym API and a live experimental evaluation with a live robot in our laboratory running BOPA which had to assist a human teammate in solving a task, our results show that our approach is effective at identifying the correct task, efficient at solving the correct task in optimal and near-optimal times, scalable, by being able to adapt to different problem sizes, and robust, by being able to adapt non-optimal teammates, such as humans, in order to solve unknown tasks without having access to the teammates' actions and environment's reward signals.

Given that in our experimental evaluation, all sensors did not show any faulty behavior, preventing a deeper analysis of BOPA whenever the state is incorrect, our next logical line of work will be to setup a second experimental scenario where there isn't full observability of the current state (or if the current state is faultily created). In this setting we will compare BOPA against a successor which does not assume the state is fully observable, modeling the tasks as partially observable Markov decision processes instead of multi-agent Markov decision

processes in order to provide yet another layer of robustness when working with real life robots and humans.

Acknowledgements

This work was partially supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020 (INESC-ID multi-annual funding) and the HOTSPOT project, with reference PTDC/CCI-COM/7203/2020. In addition, this material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-19-1-0020, and by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215. The first author acknowledges the PhD grant 2020.05151.BD from FCT.

References

1. Banerjee, A.: On Bayesian bounds. In: Proc. 23rd Int. Conf. Machine Learning. pp. 81–88 (2006)
2. Barret, S., Stone, P.: An analysis framework for ad hoc teamwork tasks. In: Proc. 11th Int. Conf. Autonomous Agents and Multiagent Systems. pp. 357–364 (2012)
3. Barrett, S.: Making Friends on the Fly: Advances in Ad Hoc Teamwork. Ph.D. thesis, The University of Texas at Austin (2014)
4. Barrett, S., Stone, P.: Cooperating with Unknown Teammates in Complex Domains : A Robot Soccer Case Study of Ad Hoc Teamwork. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (January), 2010–2016 (2015)
5. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym. arXiv preprint arXiv:1606.01540 (2016)
6. Christiano, P.F., Leike, J., Brown, T.B., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 4302–4310. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
7. Genter, K., Laue, T., Stone, P.: Three years of the robocup standard platform league drop-in player competition. Autonomous Agents and Multi-Agent Systems **31**(4), 790–820 (2017). https://doi.org/10.1007/s10458-016-9353-5
8. Melo, F.S., Sardinha, A.: Ad hoc teamwork by learning teammates' task. Autonomous Agents and Multi-Agent Systems **30**(2), 175–219 (2016)
9. Pepito, J.A., Locsin, R.: Can nurses remain relevant in a technologically advanced future? International Journal of Nursing Sciences **6**(1), 106 – 110 (2019). https://doi.org/https://doi.org/10.1016/j.ijnss.2018.09.013, `http://www.sciencedirect.com/science/article/pii/S2352013218301765`
10. Stone, P., Kaminka, G.A., Kraus, S., Rosenschein, J.S.: Ad hoc autonomous agent teams: Collaboration without pre-coordination. In: Twenty-Fourth AAAI Conference on Artificial Intelligence (2010)
11. Tasaki, R., Kitazaki, M., Miura, J., Terashima, K.: Prototype design of medical round supporting robot "terapio". In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 829–834 (May 2015). https://doi.org/10.1109/ICRA.2015.7139274