In *Adaptive and Learning Agents Workshop at AAMAS 2021* (ALA 2021),
An Online Workshop, May 2021

Reasoning about Human Behavior in Ad Hoc Teamwork

Jennifer Suriadinata
The University of Texas at Austin
Austin, Texas
jsuriadinata@utexas.edu

William Macke
The University of Texas at Austin
Austin, Texas
wmacke@cs.utexas.edu

Reuth Mirsky
The University of Texas at Austin
Austin, Texas
reuth@cs.utexas.edu

Peter Stone
The University of Texas at Austin
Sony AI
Austin, Texas
pstone@cs.utexas.edu

ABSTRACT

Ad hoc teamwork is a decentralized multi-agent problem in which agents must collaborate online without pre-coordination. An interesting challenge in ad hoc teammate design is working efficiently with human agents, which may require a model of how these agents behave in a team. In this paper, we investigate a scenario in which one of the teammates is a human, as part of a work in progress to construct an ad hoc teammate that can collaborate in mixed human-agent environments.

This paper presents an experiment that evaluates human behavior in ad hoc teamwork under three different conditions: A control group which is given a basic set of instructions and two treatment groups which are given varying levels of additional information about the collaborative nature of the task. We measure the users' performance in terms of optimality and legibility. We show that these values are significantly different between the conditions, thus highlighting the importance of acquiring a model that encompasses different human behaviors.

KEYWORDS

ad hoc Teamwork, human-in-the-loop, collaborative agents, cooperation

1 INTRODUCTION

Autonomous agents are becoming increasingly capable of solving complex tasks, but encounter many challenges when required to solve such tasks as a team. In many multiagent tasks, the coordination strategy is either learned or decided a priori while assuming full knowledge of the teammates and the task at hand. However, as agents become more robust and diverse, they are more likely to cooperate in new situations without the ability to coordinate in advance. This motivation is the basis for *ad hoc teamwork*, which is defined as collaborating with teammates without pre-coordination [3, 18]. This terminology reflects that the *collaboration* is ad hoc – the ways in which the agents learn, act and interact may be quite principled. There are two main properties that distinguish ad hoc teamwork from other multiagent systems. First, it assumes that all teammates strive to be collaborative [18]. Second, the properties of the environment and of the teammates cannot be changed by the ad hoc agent. Its task is to reason and plan under these conditions.

While works on ad hoc teamwork don't explicitly assume that the teammates are all artificial agents, these works are generally evaluated in scenarios in which none of the teammates are human. This evaluation scheme fails to account for a highly common type of ad hoc teamwork in the real-world, where there is at least one human in a team of adaptive agents. The fundamental question of this paper is "how do human agents behave when collaborating with other agents in ad hoc settings?"

We start this investigation using a specific ad hoc teamwork setup, called the *tool fetching domain*. This domain is a grid world with workstations, in which there are two teammates - one agent, the worker, needs to reach a specific workstation, and while the other agent, the fetcher, needs to fetch the worker an appropriate tool from a toolbox, according to the workstation the worker goes to. In our experimental setup, the role of the worker is taken by a human user, and the fetcher is an artificial agent that is trying to recognize the worker's goal. We test three conditions: A control group which is only given a basic set of instructions on the task and two other groups which are given differing levels of additional information about the collaborative nature of the task. We measure the users' performance in terms of both optimality and legibility.

The results of our experiments show that in ad hoc settings (1) people usually do not act in an optimal way, not taking the shortest plan to their goal; (2) people do not choose the most legible path, even when choosing that path would not lengthen their trajectory; (3) providing incentives for improving performance can assist in making people act more legibly than without these aids, but (4) without an explicit mention that other agents need to become aware of the user's goal, people do not always execute the most legible plan that will minimize the total cost of the joint plan of all the teammates. These results show that when humans are presented with the same task in different ways they will behave differently, and also motivate having varying representations of humans in ad hoc teamwork scenarios.

The rest of this paper is organized as follows. First we present related work to our study. We then discuss the ad hoc teamwork domain investigated in this paper, the Tool Fetching Domain, and describe how we augment it to allow for human teaming. We present our hypotheses about human behavior in this domain, and describe the experiments we used to test them. Finally we present our results and discuss whether each hypothesis was validated or not.

2 RELATED WORK

Ad hoc teamwork was first introduced as the challenge to create an autonomous agent that is able to efficiently and robustly collaborate with previously unknown teammates on tasks to which they are all individually capable of contributing as team members [18]. Subsequent works proposed to model teammates by mapping them to one out of a set of types [2, 14] or by directly modeling them [6]. Wang et al. [20] recently proposed an Inverse Reinforcement Learning technique to infer teammates' goals on the fly. Some works added assumptions to an ad hoc teamwork model that are inspired by real-world scenarios, such as available communication channels [5, 8, 12], influencing the behavior of teammates [1, 19], and agents that can leave or enter the environment [13]. While these works do narrow the gap between artificial agents collaborating in simulation and real-world ad hoc teamwork, they have not evaluated human users or human decision makers as one or more of the teammates in their tasks.

On the other hand, numerous works have focused on modelling human agents in collaborative settings [17]. Several works used these models to design intelligent agents that can provide advice to human teammates, such as a human operator in a multi-robot supervision task [15], in automobile climate control [16], and in presenting complex plans to care takers [4]. The ad hoc assumption of collaboration without prior coordination holds in these works, but unlike the ad hoc scenario investigated in this paper, the advising artificial agent in those works cannot perform ontic actions, hence its collaboration is limited to advice giving.

Chang et al. [9] present a novel algorithm, Teammate Algorithm for Shared Cooperation (TASC), that prioritizes its actions according to specific aspects of the teamwork in a similar fashion to people's preferences in human-human interactions. In their work, they focused on improving the behavior of an agent, as perceived by human users, while we focus on investigating the users' behavior. Huber et al. [10] augmented strategy summaries of RL agents' behavior, that were presented and evaluated by human users. Their results show that *global information* that describes the overall policy of the agent strongly affects people's understanding of agents, more than *local information* about specific decisions. These works focus on modifying an artificial agent's behavior in accordance to human teammates' preferences, where in our work we investigate the behavior of the human teammates themselves. As we will discuss in Section 4, we were informed by these works in our choices of signals to the human user in order to encourage collaboration.

3 EVALUATION DOMAIN

The domain which we modified to enable human users is the *tool fetching domain*, in which there are two agents that work collaboratively to fetch the correct tool for the given goal station [11, 12]. The domain is a discrete-action world. It contains n stationary workstations and one toolbox with n different tools. Each workstation requires exactly one unique tool to work in. The two agents in the environment are: the *fetcher* and the *worker*. Tools can only be picked up by the fetcher. The worker's task is to use some workstation, while the fetcher's task is to bring the required tools as quickly as possible to that (initially unknown) workstation. This means that the fetcher's goal depends on the goal of the worker, and hence its choice of actions rely on its understanding of the worker's goal. In order to learn the worker's goal, the fetcher uses an inference method where the probability of a goal is decreased asymptotically towards 0 whenever the worker takes an action that is not optimal for that goal. The way we chose to decrease a goal's probability is by multiplying its current probability by a factor of ϵ, where ϵ is a number close to 0, and then normalizing the probability distribution. This computation ensures that no goal probability ever reaches 0, and allows inference to occur even when the worker takes actions that are not optimal for its goal. The fetcher considers a goal to be the "true" goal for the worker when all other goals have probabilities $< \alpha$, where α is another number close to 0. There are many scenarios where we have more than one goal with probability $\geq \alpha$. In these cases, there are two options: if there is an action that is optimal for all goals with a probability $\geq \alpha$, then the fetcher will take that action. Otherwise, the fetcher cannot be certain about which action to take and it needs more information to act optimally. In this situation, the fetcher will wait in its current location until there is at least one optimal action that is common to all likely goals. In our work we use a value of 0.05 for both ϵ and α.

In our user study, we let a human user play the role of the worker. We control the fetcher and model it using the baseline agent from Mirsky et al. [12], which does not communicate with the worker. An agent can change its position during the game by executing one of the following actions: U for moving up, D for moving down, R for moving right, L for moving left, or N for staying put. The Fetcher may also pick up a tool (T). At each timestep, both agents decide separately upon an action they are interested in performing. We allow agents to step on the same grid, so no conflicts can occur.

The scenario ends successfully when both agents have reached some workstation, and the fetcher holds the relevant tool to working in that station. The transitions are deterministic, and in our setup only one object can be held at a time – if the fetcher picks up a tool while holding another one, the old one automatically drops in the square of the pickup. Pickup has no effect unless the agent is at the same position as a tool. Figure 1 shows a running example of the tool fetching domain, with the two agents at their initial locations. The fetcher is represented by a circle with the label "F", the worker is represented by a circle with the label "W". In this example, there are four workstations that are represented by numbered rectangles, the toolbox is represented by a rectangle with the label "T", and the goal station is the green rectangle with the label "2". The fetcher must first determine based on the worker's actions which of the four workstations is the worker's goal, and subsequently pick up the tool from the toolbox "T" and bring it to the station "2".

4 ADDING HUMAN TEAMMATES

We used a similar setup to previous works on the tool fetching domain [11] to create a preliminary exploratory study that will provide us insights about human users in the domain.

This initial setup was created using Pygame where one could run the program in their terminal to "play" the scenario. We augmented the program to run in the browser and enabled a user to input actions for the worker, while the fetcher was modeled using the baseline agent from previous works [12] – it determines its next move based on the worker's actions. We parameterized a variety of

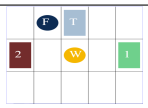
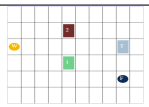
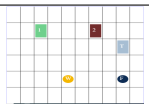
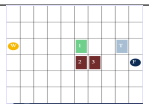
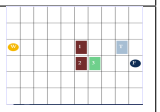
Type	Station at the left or right of worker	Split stations horizontally	Split stations vertically	Station at every corner of diamond	Station at every corner of a square	Clustered stations	Clustered stations
Min. Leg.	1	1	1	1	2	5	7
Optimal length	2	5	5	4	4	5	7
# of Instances	2	2	2	4	4	1	1
Example							

Table 1: Details about the tested scenarios. Min. Leg. is the minimal number of steps it can take for the worker's goal to become clear. Optimal length is the minimal number of steps the worker must take to reach its goal.

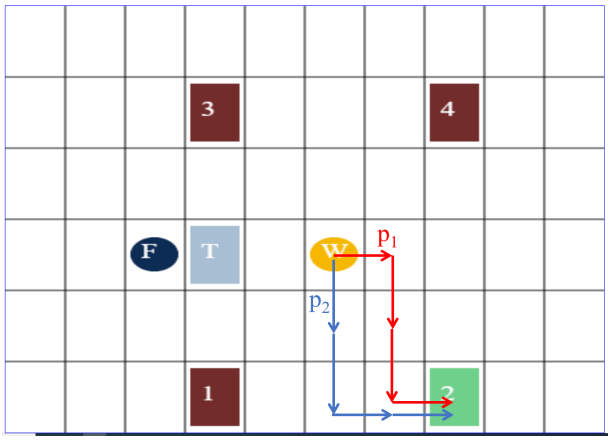


Figure 1: An experiment setup. The worker is labeled by the yellow circle, the fetcher is the dark blue circle, the toolbox is the light blue rectangle, the stations are brown rectangles, the goal station is the green rectangle.

problem instances of the tool fetching domain based on size, locations of stations, and initial positions of agents. These experiment setups were given to the user in a randomized order.

Our interaction design was inspired by several previous works: First, the concept of shared cooperative activity (SCA) is used to identify how two users mutually respond to one another and try to assist in accomplishing both shared goals, and individual intermediate goals [7]. Chang et al. [9] showed that legibility is perceived as less critical component in a SCA in human-agent collaboration than other metrics such as the value the agent brings to the promotion of the shared goal. This conclusion encouraged us to carefully explain to users the importance of *them* acting legibly in a collaboration: "The fetcher does not know the worker's goal and will need to infer this goal based on the worker's actions".

Second, Huber et al. [10] showed that users are highly affected by the global information about other agents' policies. Inspired by their results, we split our explanation about the fetcher into two parts: local information about specific decision points (e.g. "It will go to the toolbox, where it will need to choose the right tool that corresponds to the worker's goal workstation"), and global information about the fetcher's goal (e.g. "the fetcher needs to know as soon as possible which tool to fetch for you").

By combining these insights from previous works, we presented the users with the following instructions after a general description of the environment and players : " *You will play the role of the worker. Your goal is to move to the goal station and press the "Done" button once you have reached it. Each scenario will have a different layout and a different goal station, so make sure to go to the correct goal station for the given scenario.* "

Some participants were given local information about the fetcher's goal, and additional incentives to improve their performance, without an explicit reference to the metrics evaluated: " *The fetcher does not know the worker's goal and will need to infer this goal based on the worker's actions. It will go to the toolbox, where it will need to choose the right tool that corresponds to the worker's goal workstation, and then fetch that tool to the worker.* "

Lastly, some participants also received global information, in addition to the local information and incentives, to persuade them act legibly by explicitly mentioning the need of the fetcher to infer the goal of the worker: " *Your main objective should be to act in such a way that the fetcher can figure out where your goal is in the least amount of moves. This way, the fetcher will know as soon as possible which tool to fetch for you.* "

5 EXPERIMENTAL DESIGN

Prior to our main experiment, we gathered preliminary data using Amazon Mechanical Turk (MTurk), where we deployed small batches of our experiment for MTurk workers to complete. Each experiment started with the user signing a consent form and reading a page of instructions before starting the experiment. For each participant, we recorded the actions taken and the time per action in the different setups. The MTurk worker was given a unique key at the end of each experiment to confirm that they had completed the experiment successfully.

We designed three different conditions in order to gain some insight as to the human interaction with the tool fetching domain as well as how legibly or optimally the human will act.

Condition 1 - Baseline: we did not mention the fetcher's prediction of the worker's goal station in the instructions, and provided only instructions about the worker's goal.

Condition 2 - Incentive: included bolded text in the instructions about the artificial fetcher. We mentioned that the fetcher is trying to predict the goal station of the worker and obtain the correct tool for that station. This condition also included bolded text in the first screen mentioning that a bonus would be provided to the top 10 percent of workers, based on the number of steps the experiment takes but not the time it takes to complete the assignment.

Condition 3 - Instruction: included additional bolded text in the instructions about the artificial fetcher. We explicitly told the participants that they needed to act legibly, as discussed in Section 4. We also included the bolded text about the artificial fetcher and bonus from condition 2.

In our user study, we tested the following hypotheses:

- **H1.** With only a task description, human actors will not take the shortest path to their goal.
- **H2.** With only a task description, human actors will not choose a legible plan, even if the length of that plan is the same as their chosen plan.
- **H3.** Given an incentive to perform better and local information about the role of the fetcher, human actors will take the shortest path to their goal.
- **H4.** Given an incentive to perform better and local information about the role of the fetcher, human actors will prefer to execute a legible plan, as long as this plan is still on a shortest path to the goal.
- **H5.** Given an incentive, local information, and global information that the fetcher needs to infer the goal of the worker, human actors will take the shortest path to their goal.
- **H6.** Given an incentive, local information, and global information that the fetcher needs to infer the goal of the worker, human actors will prefer to execute a legible plan, as long as this plan is still on a shortest path to the goal.

Each study had the same 16 experiments, as summarized in Table 1. Some of the scenarios are identical except for the goal station location, and are then presented under the same column, with **# of instances** enumerating these individual scenarios. The number and locations of the workstations were manually picked by the researchers to cover a variety of difficulty and ambiguity levels. The grid size varied from 5×3 to 15×9, and the length of the optimal joint plan from both agents varied from 5 to 11. Consider the left clustered-stations instance in Table 1: if a user takes one action east with a goal station of 1, it is unclear where they are headed as stations 2 and 3 are also to the east. A possible legible move would then be to go north as station 1 is further north than the other stations. However, this is also a non-optimal plan and it is unclear if legibility should be more important than optimality for the human agents. Thus, the last two "clustered stations" instances were meant to be exploratory setups where a legible action might cause the worker to take a longer path to the goal than optimal. In all other instances, there is a clear legible path that is also the shortest path. In other words, the minimum number of steps it will take to make the worker's goal obvious is strictly less than the minimum number of steps to get to the goal station. This can be seen in Table 1 where **Min. Leg.** is less than **Optimal Length** in all instances except those with clustered stations. In our quantitative results, we first discuss the 14 simpler instances and then include a separate discussion for the more complicated instances.

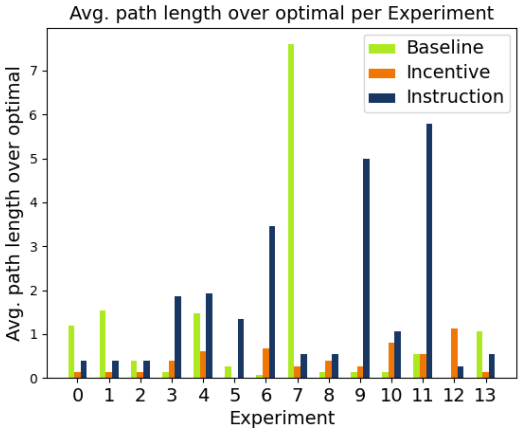


Figure 2: Average worker's path length above the shortest possible path, for each experiment setup.

Each participant was asked to act as the worker in all sixteen experiments in a randomized order. Each experiment had one tool station for the fetcher and one goal workstation for the worker. The experiment setups were manually created by the researchers to represent specific ambiguous situations. The user was informed about the worker's goal station that it should aim for both in a title above the grid, and by differentiating the goal station from the others using a brighter color.

6 RESULTS

As in accordance with the University of Texas Institutional Review Board (IRB), the experiment was conducted using Amazon Mechanical Turk (MTurk)[1]. We had a total of 45 participants, 15 in each of the conditions. All participants have given their consent to participate in this study, and were paid $0.5 for their participation. The top 10 percent from the second and third incentive conditions were given an additional $0.5.

6.1 Optimality

Figure 2 displays the average path length from the optimal path length per experiment for each condition. The average path length is the average number of steps for the worker to go to the correct workstation and work at that station. To standardize the average path length across the experiments, we use the average path length from optimal, noted as **Optimal Length** in Table 1. An average path length from optimal of 0 means that all the human agents for that experiment performed optimally and an average path length from optimal of 1 means that on average the human agents took one step more than optimal. To analyze this data we use a one-tailed paired t-test in a comparison with an optimal agent. We utilize a

[1]https://www.mturk.com/

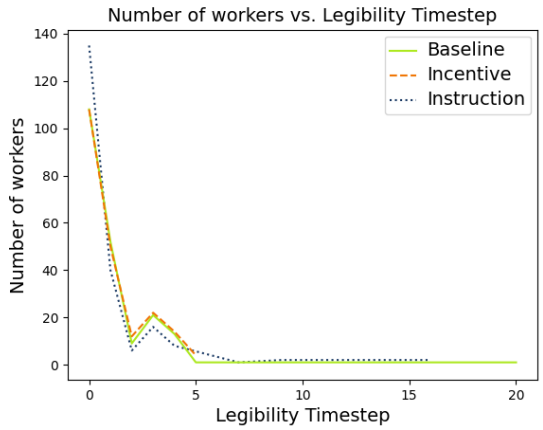


Figure 3: Worker's legibility plan for each condition. A smaller legibility timestep indicates they acted more legibly while a larger number indicates the did not act legibly.

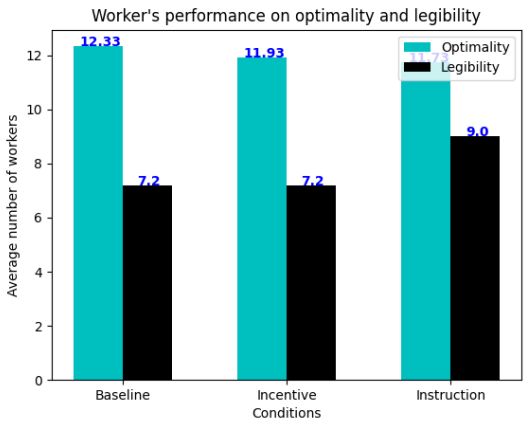


Figure 4: Number of instances each worker acted optimally vs legibly.

one-tailed test because it is not possible to have a path length less than optimal, thus a deviation from optimal will always be on one side of the distribution, greater than optimal. We use a paired test to compare each experiment value against itself, as if an optimal agent completed the experiment themselves. Based on this t-test of the data, there is a p-value of 0.034 between optimal behavior and the **Baseline** condition, < 0.001 between optimal and the **Incentive** condition, and 0.003 between optimal and **Instruction** condition.

6.2 Legibility

Firstly, we measure how many steps a worker can take before acting in a legible manner, noted as **Min. Leg.** in Table 1. Step here mean the number of actions the worker or fetcher takes, as the worker and fetcher take actions simultaneously. In this analysis of legibility, we do consider whether an optimal path was taken by the human agent. Notice that sometimes no single action is sufficient to distinguish between all goals. For example, in Figure 1, p_1 is the action sequence in red that starts with east move then south is a plan that minimizes the time to infer the goal of the agent. Since different instances had a different number of min. legibility steps, we standardize the results by counting the steps after **min.leg**. For p_1 as described above, this value will be a legibility timestep of 0. If a worker takes the action sequence south, south, then east as depicted by p_2, this plan will have a legibility timestep of 1 as it is 1 more than the minimum steps needed for the fetcher to recognize the worker's goal.

Figure 3 displays how long it took each player to make the legible move beyond the minimum required, by counting the number of workers with **min. leg.** $+0, 1, \ldots$ steps. We see that the graph is skewed towards the left, showing that most workers will act legibly as soon as possible. However, we can also see that workers with the **Instruction** condition will take the most legible action immediately much more often than in the other two conditions. We conducted a chi-square test comparing the number of workers taking the minimum number of actions and the number of workers taking more than the minimum. For **Baseline** and **Incentive** conditions, we see that there was no significant difference even though the workers are given incentive to improve their performance ($p = 0.92$).Moreover, the significant difference between the **Baseline** and **Instruction** conditions ($p = 0.01$) implies that humans can act more legibly given more insight about the other teammate's task to infer their own goal, and specifically asked to act in such a way that will entail legibility.

6.3 Additional discussion

Figure 4 visualizes the changes between the prevalence of optimality and legibility in the different conditions. It seems that even though in these experiments, there is a legible plan that is also optimal, users still perform less optimally when given instructions to guide their strategy.In the experiments analyzed for legibility and optimality, human agents acted both optimally and legibly 151 times, neither optimally nor legibly 79 times, legibly but not optimally 11 times, and optimally but not legibly 389 times. Some actors were found to take a legible move initially but not take the least amount of steps to go to the goal, thus resulting in some taking legible actions but not action optimally. Other actors were more exploratory in their behavior, taking neither a legible nor optimal path. However, the majority of participants would act optimally and legibly, and even more so would take an optimal plan but not necessarily legible.

In addition, as mentioned in Section 5, the last two instances in Figure 1 had a more complex relation between optimality and legibility, where the most legible plan might not be optimal. From the results, we noticed people would take the shortest path even if this path was not the most legible. We believe that this is due to people prioritizing optimality over legibility.

7 ANALYSIS

We address $H1$, $H3$, and $H5$ by analyzing the optimality of the paths taken by the human agents. Based on the data presented previously, there is a significant difference between optimal and the **Baseline** condition, meaning that agents performing under the

Baseline condition are not similar to optimal agents ($p = 0.034$). This result supports $H1$ which states that humans *will not* act optimally provided only basic instruction. In addition, as there is a significant difference between optimal and the **Incentive** condition ($p =< 0.001$), and between optimal and the **Instruction** condition ($p = 0.002$), both $H3$ and $H5$ are refuted, as we hypothesized that humans *will* act optimally given incentive and instruction.

We address $H2, H4$, and $H6$ by analyzing the legibility of the paths taken by the human agents. Based on the data presented previously, the left skew of the graph in Figure 3 entails that most workers will act legibly under all conditions. This result refutes $H2$ which states that humans *will not* act legibly provided only basic instructions for the task. Based on a chi-square test, the results show that there is not a significant difference between the **Baseline** and **Incentive** conditions although workers are given an incentive to improve their performance ($p = 0.92$). This refutes hypothesis $H4$ which states that humans will prefer to choose a legible path given an incentive to perform better. Moreover, there is a significant difference between the **Baseline** and **Instruction** conditions ($p = 0.01$) which implies that humans can act more legibly given information about the other teammate's task to infer their own goal and specifically asked to act in such a way that will entail legibility. This supports hypothesis $H6$ which states that human agents will prefer a legible path given incentive and instruction.

8 CONCLUSION AND FUTURE WORK

In this paper, we investigated how human users will behave in the context of ad hoc teamwork. We evaluated the human performance in terms of optimality and legibility, and tested how different incentives and instructions can affect these strategic considerations. While we hypothesized that humans will act optimally only when incentivized to do so, many users did not take a shortest path to the goal in most instances. We also hypothesized that users will not take the legible plan unless explicitly instructed to consider the legibility of their actions. This hypothesis is supported by our experiments. Another interesting phenomenon we observed is that users acted less optimally when given legibility instructions. Our results imply that there can be more than one type of intervention that can affect human behavior in ad hoc teamwork, and that an intervention can lead to more than one effect.

Next we wish to investigate if we can model human behavior using existing models for human decision making, such as quantal response [17]. Such a model is a crucial step in our long term goal to design artificial agents that can collaborate effectively with human agents in ad hoc teamwork.

ACKNOWLEDGMENTS

This work has taken place in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CPS-1739964, IIS-1724157, NRI-1925082), the Office of Naval Research (N00014-18-2243), Future of Life Institute (RFP2-000), Army Research Laboratory, DARPA, Lockheed Martin, General Motors, and Bosch. Peter Stone serves as the Executive Director of Sony AI America and receives financial compensation for this work. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

REFERENCES

[1] Noa Agmon and Peter Stone. 2012. Leading ad hoc agents in joint action settings with multiple teammates. In *AAMAS*. 341–348.
[2] Stefano V Albrecht and Peter Stone. 2017. Reasoning about hypothetical agent behaviours and their parameters. In *AAMAS*. 547–555.
[3] Stefano V Albrecht and Peter Stone. 2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence* 258 (2018), 66–95.
[4] Ofra Amir, Barbara J Grosz, Krzysztof Z Gajos, Sonja M Swenson, and Lee M Sanders. 2015. From care plans to care coordination: Opportunities for computer support of teamwork in complex healthcare. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 1419–1428.
[5] Samuel Barrett, Noa Agmon, Noam Hazon, Sarit Kraus, and Peter Stone. 2014. Communicating with unknown teammates. In *AAMAS*. 1433–1434.
[6] Samuel Barrett, Avi Rosenfeld, Sarit Kraus, and Peter Stone. 2016. Making Friends on the Fly: Cooperating with New Teammates. *Artificial Intelligence* (October 2016). https://doi.org/10.1016/j.artint.2016.10.005
[7] Michael E Bratman. 1992. Shared cooperative activity. *The philosophical review* 101, 2 (1992), 327–341.
[8] Mithun Chakraborty, Kai Yee Phoebe Chua, Sanmay Das, and Brendan Juba. 2017. Coordinated vs. Decentralized Exploration In Multi-Agent Multi-Armed Bandits.. In *IJCAI*. 164–170.
[9] Mai Lee Chang, Taylor Kessler Faulkner, Thomas Benjamin Wei, Elaine Schaertl Short, Gokul Anandaraman, and Andrea Lockerd Thomaz. [n.d.]. TASC: Teammate Algorithm for Shared Cooperation. ([n. d.]).
[10] Tobias Huber, Katharina Weitz, Elisabeth André, and Ofra Amir. 2020. Local and global explanations of agent behavior: integrating strategy summaries with saliency maps. *arXiv preprint arXiv:2005.08874* (2020).
[11] William Macke, Reuth Mirsky, and Peter Stone. 2021. Expected Value of Communication for Planning in Ad Hoc Teamwork. *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)* (2021).
[12] Reuth Mirsky, William Macke, Andy Wang, Harel Yedidsion, and Peter Stone. 2020. A Penny for Your Thoughts: The Value of Communication in Ad Hoc Teamwork. *International Joint Conference on Artificial Intelligence (IJCAI)* (2020).
[13] Arrasy Rahman, Niklas Hopner, Filippos Christianos, and Stefano V Albrecht. 2020. Open Ad Hoc Teamwork using Graph-based Policy Learning. *arXiv preprint arXiv:2006.10412* (2020).
[14] Manish Ravula, Shani Alkoby, and Peter Stone. 2019. Ad hoc teamwork with behavior switching agents. In *IJCAI*.
[15] Ariel Rosenfeld, Noa Agmon, Oleg Maksimov, and Sarit Kraus. 2017. Intelligent agent supporting human–multi-robot team collaboration. *Artificial Intelligence* 252 (2017), 211–231.
[16] Ariel Rosenfeld, Amos Azaria, Sarit Kraus, Claudia V Goldman, and Omer Tsimhoni. 2015. Adaptive advice in automobile climate control systems. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. 543–551.
[17] Ariel Rosenfeld and Sarit Kraus. 2018. Predicting human decision-making: From prediction to action. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 12, 1 (2018), 1–150.
[18] Peter Stone, Gal A Kaminka, Sarit Kraus, and Jeffrey S Rosenschein. 2010. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI*.
[19] Peter Stone, Gal A Kaminka, Sarit Kraus, Jeffrey S Rosenschein, and Noa Agmon. 2013. Teaching and leading an ad hoc teammate: Collaboration without pre-coordination. *Artificial Intelligence* 203 (2013), 35–65.
[20] Rose E Wang, Sarah A Wu, James A Evans, Joshua B Tenenbaum, David C Parkes, and Max Kleiman-Weiner. 2020. Too many cooks: Coordinating multi-agent collaboration through inverse planning. *AAMAS* (2020).