2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Robust Robot Planning for Human-Robot Collaboration

Yang You[1], Vincent Thomas[1], Francis Colas[1], Rachid Alami[2], and Olivier Buffet[1]

Abstract— **In human-robot collaboration, the objectives of the human are often unknown to the robot. Moreover, even assuming a known objective, the human behavior is also uncertain. In order to plan a robust robot behavior, a key preliminary question is then: How to derive realistic human behaviors given a known objective? A major issue is that such a human behavior should itself account for the robot behavior, otherwise collaboration cannot happen. In this paper, we rely on Markov decision models, representing the uncertainty over the human objective as a probability distribution over a finite set of objective functions (inducing a distribution over human behaviors). Based on this, we propose two contributions: 1) an approach to automatically generate an uncertain human behavior (a policy) for each given objective function while accounting for possible robot behaviors; and 2) a robot planning algorithm that is robust to the above-mentioned uncertainties and relies on solving a partially observable Markov decision process (POMDP) obtained by reasoning on a distribution over human behaviors. A co-working scenario allows conducting experiments and presenting qualitative and quantitative results to evaluate our approach.**

I. INTRODUCTION

Building smart robots to assist a human partner is a topical subject with applications in manufacturing [1]–[4], healthcare [5], etc. In those applications, the robot often adapts to a single fixed human objective. But to make it robust, we need to consider how the robot could adapt if the human's objective and his induced behavior are uncertain.

To circumvent the uncertainty over human objectives, Hadfield-Menell *et al.* [6] proposed the CIRL framework (cooperative inverse reinforcement learning), where both the human and the robot have to maximize the human's reward function, which is hidden to the robot. This reward function can be seen as encoding the task of the human but also, via intermediary rewards, his preferences. In the CIRL framework, solving the planning problem is done through seeking a pair of behaviors, or policies, one for the robot and one for the human, which optimize the reward in the long term. However, CIRL relies on two strong assumptions we want to relax: 1) The human should follow the policy computed for him, while it may often be too complex to communicate or learn. 2) All state variables are visible, except for the human's objective (reward function parameter), which is hidden to the robot.

In this work, we want to compute a robot policy robust to incomplete information, and in particular on the human objectives and preferences, but also to unplanned behavior. To consider an independent human in this robot planning problem, we need a model of his behavior that accounts for the robot's possibility to collaborate. This induces a chicken-and-egg issue as the robot behavior is unknown at this stage.

Our first contribution consists in modeling a *collaborative* human behavior from a given reward function. This is done by 1) *temporarily* assuming that the human can control the robot with direct access to its observations, to compute the optimal value function of the resulting (multiagent) partially-observable Markov decision process (POMDP); 2) then, relaxing the shared observability assumption, extracting a model for the human alone from that value function.

Our second contribution is to derive and solve a planning problem to build a robust policy for the robot. This is achieved by constructing a new POMDP including, as part of the robot environment, a model of the human as a global stochastic finite-state controller (FSC), which samples the FSCs extracted for each given reward function as above.

Section II discusses related works in human-robot collaboration. Sec. III formally defines POMDPs, Decentralized POMDPs, and FSCs. Sec. IV explains how to automatically generate a stochastic human FSC for a given reward function. Sec. V describes how to build a robust robot policy to adapt to several possible human policies. Finally, Sec. VI presents empirical results obtained with both synthetic and real humans on a high-level task in a simulated environment.

Note: Additional details and results are provided in the appendix of an extended version of this paper [7].

II. RELATED WORK

One can distinguish between different approaches to human-robot collaboration depending on the (robot's) "human mental model", which, according to Tabrez *et al.* [8], can be in one of the three following categories: *first-order mental model (1oMM)*: the robot considers that the human behaves independently and does not account for the robot's possible actions; *second-order mental model (2oMM)*: the robot considers that the human accounts for the robot, which induces some form of recursive modeling up to a certain depth; *shared-mental model (SMM)*: an SMM assumes that all agents in the team have common expectations, thus reason in the same manner, which ensures an optimal coordination. Of course this categorization symmetrically applies to how the human models the robot. Let us mention that mental models can also been used in other AI planning settings, *e.g.*, related to explainability [9].

We focus here on problems with stochastic dynamics and partial observability, which leads us to considering

This work was supported by the French National Research Agency (ANR) through the "Flying Coworker" Project under Grant 18-CE33-0001.

[1] Université de Lorraine, INRIA, CNRS, LORIA, F-54000 Nancy, France firstname.lastname@loria.fr

[2] LAAS-CNRS, Université de Toulouse, CNRS, F-31000 Toulouse, France firstname.lastname@laas.fr

Markov decision models. In this setting, a 1oMM typically corresponds to a POMDP where the objective is to find the policy of one agent of interest, while the (a priori known) policy of the other agent is part of the system dynamics. For instance, a "robot POMDP" assuming a known human behavior is solved in [10]–[13]. SMMs can be formalized as *Decentralized POMDPs* (Dec-POMDPs), typically used to optimize the joint policy of a team of agents (with a common reward function). CIRL can be seen as a special case where the human has full observability and the robot's only hidden variable corresponds to the actual human objective (his reward function), which allows for dedicated solution techniques close to solving a POMDP. For their part, 2oMMs can be formalized as *Interactive POMDPs* (I-POMDPs) [14], where agents model each others in a nested manner.

1oMMs will fail in many tasks requiring an explicit collaborative behavior from the human, and the SMMs' main assumption is typically too strong when collaborating with a human. We will thus equip the robot with a 2oMM of the human. Our approach to robust robot planning could be formalized as an I-POMDP, what we avoid mainly to simplify notations. However, 2oMMs always raise a chicken-and-egg problem as deriving the required human behavior implies reasoning about the robot behavior we are trying to derive in the first place. This is true for instance in case of 1) a hand-made human policy [10], [11], where the human designer has to reason on the robot behavior; 2) a learned human policy [15], [16] or human reward (through IRL) [17], [18], which requires a collaborative robot in the first place to demonstrate a collaborative behavior; or 3) a planned human policy: the model needs to include the robot behavior.

In this work, we generate plausible human behaviors—which will serve as the robot's mental model of the human—through planning, and address the chicken-and-egg problem through answering the question: "*What if the human could also control the robot?*", thus temporarily assuming that human and robot share their observations, which amounts to the human adopting a particular SMM.

III. Background

A. Dec-POMDPs

We use Dec-POMDPs only to formalize the collaboration problem for convenience, but do not solve a Dec-POMDP to get a joint policy for the human and the robot.

Definition 1: A *Dec-POMDP* with $|\mathcal{I}|$ agents is defined by a tuple $M \equiv \langle \mathcal{I}, \mathcal{S}, \mathcal{A}, \Omega, T, O, R, b_0, \gamma \rangle$, where: $\mathcal{I} = \{1, \dots, |\mathcal{I}|\}$ is a set of *agents*; $\mathcal{S}$ is a set of *states*; $\mathcal{A} = \bigtimes_i \mathcal{A}^i$ is a set of joint actions, with $\mathcal{A}^i$ the set of agent i's *actions*; $\Omega = \bigtimes_i \Omega^i$ is a set of joint observations, with Ω^i the set of i's *observations*; $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the *transition function*, with $T(s, a, s')$ the probability of transiting from s to s' if a is performed; $O : \mathcal{A} \times \mathcal{S} \times \Omega \to \mathbb{R}$ is the *observation function*, with $O(a, s', o)$ the probability of observing o if a is performed and the next state is s'; $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the *reward function*, with $R(s, a)$ the immediate reward for performing a in s; b_0 is the *initial probability distribution* over states; and $\gamma \in [0, 1)$ is the *discount factor* applied to future rewards.

An agent's i action *policy* π^i maps its possible action-observation histories to distributions over actions. The objective is then to find a joint policy $\pi \equiv \langle \pi^1, \dots, \pi^{|\mathcal{I}|} \rangle$ that maximizes the expected discounted return from b_0:

$$V^{\pi}(b_0) \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) \mid S_0 \sim b_0, \pi\right].$$

In a POMDP, *i.e.*, when $\mathcal{I} = \{1\}$, many solvers rely on estimating the optimal value function $V^*(b)$, or the action-value $Q^*(b, a) \stackrel{\text{def}}{=} R(b, a) + \gamma \sum_o Pr(o|b, a) \cdot V^*(b_a^o)$, where b_a^o is the next belief when performing a and observing o.

B. Finite State Controllers

In this work, human policies are presented in the form of *finite state controllers* (FSC) (also called *policy graphs* [19]), *i.e.*, automata which contain in each internal state a probability distribution from which to sample an action, and whose transitions from one internal state to the next depend on the action performed and observation received.

Definition 2: For some POMDP sets $\mathcal{A}$ and Ω, a stochastic *FSC* is defined by a tuple $fsc \equiv \langle N, \beta, \eta, \psi \rangle$, where:

- N is a finite set of (internal) nodes;
- β is a probability distribution from which to sample the initial node;
- $\eta : N \times \mathcal{A} \times \Omega \times N \to \mathbb{R}$, the transition function, gives the probability $\eta(n, \langle a, o \rangle, n')$ of transiting from node n to n' if a is performed and o is observed; a deterministic transition can be noted $n' = \eta(n, \langle a, o \rangle)$;
- $\psi : N \times \mathcal{A} \to \mathbb{R}$, the action selection function, gives the probability $\psi(n, a)$ of choosing $a \in \mathcal{A}$ when in n.

IV. Generating Human Policies with Objectives

We now describe how, from a known reward function (a human objective), to generate one of the human FSCs which will feed the robot planner in Sec. V. To model uncertainty, the process can be tuned to create more or less erratic human behaviors.[1]

The reward function at hand induces a Dec-POMDP D describing a collaboration problem. As stated before, to account for possible interactions, an issue is that the human policy we are seeking depends on the robot policy we don't have in the first place. To overcome this chicken-and-egg problem, we temporarily assume that the human can control the robot's actions and has access to its observations. The Dec-POMDP is thus first relaxed as an MPOMDP M (Multi-agent POMDP) [20], *i.e.*, a single-agent problem. M can be fed to a POMDP solver to compute an optimal action-value $Q^*_M(b, a)$ for any (belief,joint-action) pair (b, a).

To derive a human policy usable in Dec-POMDP D, *i.e.*, mapping human action-observation histories (alone) to human actions, let us now assume 1) that the human does not control the robot anymore, but 2) that the robot still has access to the same belief b as inferred by the human.

[1] Uncertainty about human objectives is handled in Sec. V.

Given a belief b, we can model the uncertainty over human (and robot) behaviors using a softmax function over action-values to obtain a distribution over multiple optimal or near-optimal joint actions: $f(a|T,b) = e^{\frac{Q^*_M(b,a)}{T}} / \sum_{a'} e^{\frac{Q^*_M(b,a')}{T}}$, where $T > 0$ is a temperature parameter that makes the human more *rational* if T is low, only optimal actions being selected, and more *erratic* if T is high, with a distribution close to uniform. Then, because both are now independent, human action a_H is sampled with probability $\sigma^T_H(a_H|b) \stackrel{\text{def}}{=} \sum_{a_R} f(a_H, a_R|T,b)$, and robot action a_R is sampled w.p. $\sigma^T_R(a_R|b) \stackrel{\text{def}}{=} \sum_{a_H} f(a_H, a_R|T,b)$. Given these two action-sampling rules σ^T_H and σ^T_R, the human can update his belief in Dec-POMDP D given his last pair (a_H, o_H) by marginalizing over the robot's private actions and observations.

Using these processes to pick a human action and to update his belief, we designed an algorithm that recursively extracts a human policy (represented as an FSC) building on You *et al.*'s Algorithm 2 for standard POMDPs [21] (see also Grześ *et al.*'s work [22]). In our setting, due to the stochastic decisions, the FSC transitions depend not only on the last observation, but also on the last action. Note that our algorithm allows trading off the FSC quality with its complexity through 1) bounding the FSC's number of nodes by $N_{\max}$, and 2) merging nodes when their reference beliefs (the beliefs used when creating the nodes) are within ϵ of each other. Also, inspired by LAO* [23], we obtain a better FSC (as confirmed through experiments) by expanding, at each iteration, the (open) node which seems to contribute most to b_0's value. To that end, we select the open node n that has the highest estimated value $V^*(n.b)$, weighted by the probability to reach that node. Finally, self-loops are added in each node for human observations that are impossible under the current belief, so that the resulting human FSC can be used whatever the robot's actual behavior.

This approach (further detailed in App. A) extracts a bounded-size stochastic FSC encoding a variety of human behaviors. In our experiments, all $Q^*(b,a)$ and $V^*(b)$ values are estimated using 1) SARSOP for offline pre-computations [24], and 2) POMCP to obtain good estimates online quickly, even in beliefs not visited by an optimal policy [25].

Deterministic FSCs will also be generated to simulate various humans in our experiments. This is achieved by replacing each node's distribution over human actions by a single action sampled from that distribution. See App. B.

V. Robust Robot Task Planning

We now want to derive a robot policy that is robust to different possible (hidden) human objectives, each attached to a different behavior. This problem is formalized as a Dec-POMDP D, except that 1) the exact reward function (among ρ candidates) is only known by the human; 2) each reward function r_i is attached to a human FSC $fsc_i \equiv \langle N_i, \beta_i, \eta_i, \psi_i \rangle$ (*cf.* Sec. IV); and 3) the robot is given a probability distribution over the possible reward-FSC pairs: $P(\{(r_1, fsc_1), \ldots, (r_\rho, fsc_\rho)\})$. As detailed below, this robust robot behavior is obtained by first turning this distribution over FSCs into a single FSC, then using this FSC to derive a POMDP, and finally solving this POMDP.

To turn this probability distribution over human FSCs into a single FSC, we take their "union", the new distribution over initial nodes amounting to 1) sampling one FSC fsc_i from the distribution $P(\{fsc_1, \ldots, fsc_\rho\})$, and then 2) sampling a node from β_i. More formally, the *union FSC* is defined as:

$$N \stackrel{\text{def}}{=} \bigcup_{i=1}^{\rho} N_i,$$
$$\beta(n) \stackrel{\text{def}}{=} \beta_{i(n)}(n) \cdot P(fsc_{i(n)}),$$

where $i(n)$ is the id of n's parent FSC: $i(n) \stackrel{\text{def}}{=} i$ s.t. $n \in N_i$,

$$\eta(n, \langle a, o \rangle, n') \stackrel{\text{def}}{=} \begin{cases} \eta_{i(n)}(n, \langle a, o \rangle, n') & \text{if } n' \in N_{i(n)}, \\ 0 & \text{otherwise,} \end{cases}$$
$$\text{and } \psi(n,a) \stackrel{\text{def}}{=} \psi_{i(n)}(n,a).$$

Given this FSC, we can now formalize the robot's decision problem as a POMDP where each *extended state* $e^t \in \mathcal{E}$ contains a current world state s^t, a robot current observation o^t_R and the current node n^t_H inside the human union FSC: $e^t \equiv \langle s^t, n^t_H, o^t_R \rangle$.

Based on the Dec-POMDP D and on the associated union FSC, the dynamics and the reward function of the robot's decision problem can be written as follows:

$$T_e(e^{t+1}, e^t, a^t_R) = Pr(e^{t+1}|e^t, a^t_R)$$
$$= \sum_{a^t_H} \sum_{o^{t+1}_H} T(s^t, \langle a^t_H, a^t_R \rangle, s^{t+1}) \cdot \eta(n^t_H, \langle a^t_H, o^{t+1}_H \rangle, n^{t+1}_H) \cdot$$
$$O(s^{t+1}, \langle a^t_H, a^t_R \rangle, \langle o^{t+1}_H, o^{t+1}_R \rangle) \cdot \psi(n^t_H, a^t_H),$$
$$O_e(e^{t+1}, a^t_R, o^{t+1}_R) = Pr(o^{t+1}_R|e^{t+1}, a^t_R) = \mathbf{1}_{o^{t+1}_R = \tilde{o}^{t+1}_R}$$

(where $\tilde{o}^{t+1}_R$ is the observation in e^{t+1}), and

$$r_e(e^t, a^t_R) = \sum_{a^t_H} r_{i(n^t_H)}(s^t, \langle a^t_H, a^t_R \rangle) \cdot \psi(n^t_H, a^t_H).$$

Solving this robot POMDP gives a robust robot policy which is a best response to the provided probability distribution over human policies.

VI. Experiments

A. Experimental Setting

Our experiments have been conducted on a laptop with an 2.3 GHz i9 cpu and the source code is open sourced [26].

To test our approach, we have designed a scenario presented in Figure 1, where a robot and a human have to repair and maintain several devices located in a grid world. One device needs to be maintained by the robot, which should be on the device's cell. Repairing one of the two broken devices requires both the human and the robot to perform repair actions simultaneously at the device location, the human having previously picked a component in a toolbox. Note also that both the human and the robot have limited observation of the environment.

This task is specified as the following Dec-POMDP:

- States (S): The state $s \in S$ of the problem is made up of: 1) the human location, 2) the robot location, 3) the status of the devices, and 4) whether the human has a component or not. The human and robot locations are represented by integer coordinates (x, y). They can be on the same cell. The state of each device is either "good", "broken", or "needs maintenance".
- Human observations (Ω_H): The human observes 1) his location; 2) whether the robot is on his cell or not; 3) the status of the device in his cell (if any).
- Robot observations (Ω_R): The robot observes 1) its location; 2) the human's location; 3) the status of the device in its cell (if any).
- Actions and Dynamics: Both the human and the robot have the following actions: *Up, Down, Left, Right* are the agent's 4 move actions; *Wait*: the agent stays in his current position; *Repair*: repairs a broken device if: 1) the human holds a new component; 2) the human and the robot are in the same cell as the broken device; 3) the human and the robot both perform the repair action. Upon success, the broken device turns to *good* and the human's component is consumed.
  The human can *Pick a Component* if he is in the toolbox area and if he does not already hold one. The robot can *Maintain* a device individually if it needs to be maintained and if in the same location. Upon success, the device status turns to *good*.
- Rewards R: A reward of $+100$ is given when all devices have been repaired or maintained. A reward (penalty) of -2 is given for each action except for the human's *Wait*, and a penalty of -20 is given in case of invalid action. A penalty of -1 is given if the human waits before having repaired all the broken devices. If all devices are in "good" status, no penalty (0) is given to the human wait action. The penalty associated to the wait action encourages the robot to postpone maintenance actions if this helps the human finish repair actions early.

This large Dec-POMDP has 2 304 states, 49 joint actions (7 actions per agent), and 5 400 joint observations (30 for the human and 180 for the robot). Note: This state space grows quadratically with the number of cells.

To represent uncertainty about the human objectives, we replaced the reward function described previously by two variants: 1) the human prefers to repair the broken device on the left first, receiving a $+10$ reward in that case; and 2) symmetrically, the human prefers to repair the broken device on the right first. These objectives generate different human policies using Alg. 1. Initially, the robot only has a prior over the human's rewards and associated policy (each with probability 0.5). Due to the required coordination for repairing devices, these human policies have to account for the robot's ability to help the human when needed.

B. Experiments with Synthetic Humans

We used the method described in Sec. IV to compute the stochastic human FSCs associated to each human objective, and the method described in Sec. V to compute the robust robot policy with the POMDP solver SARSOP [24]. To save resources, an *action-threshold* (here always set to 0.1) is used to prune low-probability human actions when computing the stochastic human FSCs. We conducted experiments with 3 values for the softmax parameter T (0, 0.3 and 0.5) and several values for the maximum stochastic FSC size $N_{\max}$.

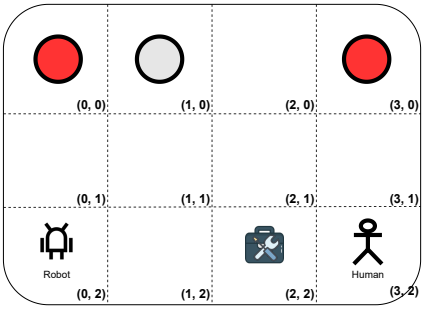

Fig. 1. A collaboration task: A robot and a human evolve in a 4×3 grid world. The top-left cell $(0, 0)$ and the top-right cell $(3, 0)$ both contain a broken device. A device to be maintained is also located in cell $(1, 0)$. A toolbox is located in cell $(2, 2)$ where the human can pick components.

1) Qualitative Results: We first observed that the extracted human stochastic FSC can effectively encode possible trajectories of the human to solve the task if $N_{\max}$ is large enough for the current T. For instance, when $T = 0.3$ and $N_{\max} = 100$, the extracted human stochastic FSCs reach depths[2] of 22 and 18 (for each objective), which are sufficient for the human to finish the task if the robot collaborates. However, when $N_{\max} = 50$, the depth covered by the FSC of the *prefer-left* objective is only 13 (15 for the 2nd objective), while a depth of 15 is required to finish the task. We also observed that the higher the temperature T, the larger the $N_{\max}$ value needed for the human stochastic FSC to regularly finish the task. When T increases, more actions are considered at each node, generating more branches and preventing sufficient depth. Note also that, even if we set T to a very low value (near 0), the softmax distribution may contain several optimal actions. This allows encoding several optimal human policies in a single FSC.

When $N_{\max}$ is large enough for the current T, we observed that the computed robust robot policy can successfuly solve the task, accounting for the uncertainties over the human objectives and behaviors. It helps the human repair all the devices and performs the required maintenance operation. For example, when $T = 0.3$, $N_{\max} = 100$, the robot following the generated robust policy first goes to cell $(0, 0)$ and waits for the human to pick a component at cell $(2, 2)$. Afterwards, if the robot observes that the human is approaching, it keeps waiting for him and then helps him repair the left device when the human reaches cell $(0, 0)$. But if it observes the human moving to the top-right corner, then the robot decides to go right and help the human repair the right device first. The robot then moves to cell $(1, 0)$ to perform a maintenance operation while the human is going to

[2]The depth of an FSC is the maximum distance between the (here unique) initial node and any node.

pick a second component. Finally, the robot helps the human repair the other broken device to finish the task.

Even if this seems like a simple pattern for the robot, it still requires the robot to reason on many possible human trajectories. For example, if there are 2 optimal actions in each state, even for 5 time steps, there are 2^5 possible optimal trajectories (and this number can increase dramatically when considering sub-optimal actions). Moreover, in this complex collaboration scenario, agents make decisions only based on partial observations, and, as illustrated above, the robot infers the human objective despite his uncertain behavior.

2) Robustness Analysis: We also wanted to quantitatively analyze the robustness of our approach by comparing the method described in Sec. IV with a method computing a robot policy only based on deterministic human policies. To that end, we extracted 50 pairs of deterministic human policies, one per objective, using $T = 0.5$ and $N_{\max} = 600$ (*cf.* end of Sec. IV), but obtaining less than 200 nodes in all cases; then, for each pair, we computed a best-reponse robot policy to the equiprobable union of these two human policies (*cf.* beginning of Sec. V). The resulting set of 3×50 policies is thus ($\Pi_H = \{\langle \pi^1_{H,l}, \pi^1_{H,r}, \pi^1_{H,l+r} \rangle, \ldots, \langle \pi^{50}_{H,l}, \pi^{50}_{H,r}, \pi^{50}_{H,l+r} \rangle\}$). For each human union policy $\pi^i_{H,l+r}$, we also compute a robot best response $\pi^i_{R,BR}$ by solving a POMDP obtained as in Sec. V.

In our experiments, we compare the robust robot policies π^*_R obtained for different values of T and $N_{\max}$ against 1) the 50 best responses $\pi^i_{R,BR}$ described above (averaging over all of them), and 2) the robot policy obtained by solving (with Inf-JESP [21]) the corresponding Dec-POMDP.[3] We measure the average value and success rate of each such robot policy against all 3×50 synthetic human policies.

The obtained results, along with standard deviations, are presented in Table I. First, as expected, the worst results are obtained with the Dec-POMDP and Best-Response solutions ($\pi_{R,Dec-POMDP}$ and $\pi_{R,BR}$), with near-zero success rates. Our approach improves a little bit in terms of success rate when using $T = 0$, *i.e.*, when assuming that the human only chooses among a small subset of actions at each time step, which leads to fairly small human FSCs (24 and 23 nodes for the two independent objectives). The average reward remains very low, though. Using $T = 0.3$ or 0.5 leads to much better results. In the *prefer-right* scenario, human FSCs with 200 nodes are even sufficient to get very good solutions (90% success rate). Moving from $T = 0.3$ to $T = 0.5$, this phenomenon is all the more important that more erratic behaviors require larger FSCs to better account for most likely trajectories. In the *prefer-left* scenario, the success rates and values drop significantly. This is due to the longer trajectories needed in this case, which in turn require large human FSCs to encode good enough policies. Then, in the "union" scenario where the human's preference is randomly sampled, the results are a simple combination of the results in the *prefer-left* and *prefer-right* scenarios.

[3]This solution assumes a strong a priori coordination since the human and robot agree on their individual policies.

Note that, in these experiments, the synthetic humans used for evaluation are rather erratic ($T = 0.5$), which explains that the best success rates are obtained with robot policies derived for that same temperature, while robots derived for $T = 0.3$ are not robust enough.

Table II presents the recorded computational time used in each step of our work. The 1st step consists in converting each Dec-POMDP task model to a MPOMDP (one by human objective); the 2nd step in generating stochastic human policies (FSCs) as presented in Sec. IV (the most time-consuming process); the 3rd step in building the robot POMDP using the task models (Dec-POMDPs) and the human FSCs; and, the final step in solving the robot POMDP using SARSOP to get a robust robot policy. Here, most of the time is spent calling POMCP in the FSC extraction.

C. Real human Experiments

To further evaluate our approach, we implemented this collaboration task as a computer game where the human is controlled through the keyboard. To make a friendly user interface, the human player can directly observe the state of the environment, as shown in Fig. 1, but the robot still faces partial observability as defined in the Dec-POMDP problem, and behaves accordingly. Then, we selected three robot policies, $\pi^*_R(0.0, 100)$, $\pi^*_R(0.3, 600)$ and $\pi^*_R(0.5, 600)$, from the previous experiment, and we invited 10 real human subjects to collaborate with those robot policies. Each subject was informed that the goal of this collaboration task was to turn each device status to "good" within at most 30 time steps, but did not know or observe the instant reward. He or she then played 8 consecutive rounds of this collaboration game per robot policy (for a total of 24 rounds), the order of the robot policies being randomly selected.

1) Qualitative results: We first observed that the robot policy $\pi^*_R(0.0, 100)$ was unable to help the human subjects since it only accounts for few high-quality human actions. For example, a common mistake of the human subjects was to forget to pick a component necessary to repair a broken device. To fix that mistake, the human subject had to go back to the toolbox, thus generating an unexpected human behaviour for robot policy $\pi^*_R(0.0, 100)$, and leading to a failure due to the robot policy getting stuck in a self-loop. On the contrary, while no specific information regarding human behaviors were provided, robot policies $\pi^*_R(0.3, 600)$ and $\pi^*_R(0.5, 600)$ managed to recover and help the human to solve the task because they consider more possible human actions in each situation. In most cases, those policies adapted to different human behaviors and were able to tolerate human mistakes such as the one mentioned above.

Moreover, after experiments, we asked our subjects how they felt about the 3 tested robot policies. 60% of our subjects felt that there was no adaptation at all when playing with $\pi^*_R(0.0, 100)$, and the other 40% felt that they needed to adapt to $\pi^*_R(0.0, 100)$ to finish the task. On the other hand, 50% of the subjects reported mutual adaptations between them and the two robot policies $\pi^*_R(0.3, 600)$ and $\pi^*_R(0.5, 600)$; 30% reported that $\pi^*_R(0.3, 600)$ and

TABLE I

AVERAGE VALUE (AND SUCCESS RATES IN PARENTHESES) OF VARIOUS ROBOT POLICIES VS. VARIOUS HUMAN BEHAVIORS. FOR EACH ROBUST ROBOT POLICY $\pi^*_R(T, N_{\max})$, WE ALSO INDICATE THE SIZES OF BOTH GENERATED HUMAN FSCS AS $N = (N_{left}, N_{right})$.

	3×50 synthetic human FSCs sampled using $T = 0.5$			10 real humans
	$\pi^{Sample}_{H,l}$	$\pi^{Sample}_{H,r}$	$\pi^{Sample}_{H,l+r}$	π^{Real}_{H}
$\pi_{R,Dec-POMDP}$	-34.9 ± 6.7 (0%)	-82.5 ± 14.4 (0%)	-58.7 ± 10.5 (0%)	
$\pi_{R,BR}$	-172.1 ± 2.5 (3%)	-174.4 ± 2.5 (3%)	-173.3 ± 2.5 (3%)	
$\pi^*_R(T = 0.0, N_{max} = 100, N = (24, 23))$	-110.0 ± 12.5 (10%)	-179.4 ± 15.4 (18%)	-144.7 ± 14.0 (14%)	-55.3 ± 6.5 (16%)
$\pi^*_R(T = 0.3, N_{max} = 200, N = (200, 200))$	-38.1 ± 9.8 (52%)	$\mathbf{10.0} \pm 2.0$ (90%)	-14.0 ± 5.9 (71%)	
$\pi^*_R(T = —, N_{max} = 400, N = (400, 400))$	-11.9 ± 7.2 (68%)	$\mathbf{10.0} \pm 2.0$ (90%)	-0.9 ± 4.6 (79%)	
$\pi^*_R(T = —, N_{max} = 600, N = (600, 443))$	-11.9 ± 7.2 (68%)	$\mathbf{10.0} \pm 2.0$ (90%)	-0.9 ± 4.6 (79%)	$\mathbf{-20.8} \pm 7.4$ (**81**%)
$\pi^*_R(T = 0.5, N_{max} = 200, N = (200, 200))$	-36.6 ± 5.6 (16%)	$\mathbf{10.0} \pm 2.0$ (90%)	-13.3 ± 3.8 (53%)	
$\pi^*_R(T = —, N_{max} = 400, N = (400, 400))$	2.7 ± 2.5 (76%)	$\mathbf{10.0} \pm 2.0$ (90%)	6.3 ± 2.3 (83%)	
$\pi^*_R(T = —, N_{max} = 600, N = (600, 600))$	$\mathbf{9.5} \pm 2.2$ (84%)	$\mathbf{10.0} \pm 2.0$ (90%)	$\mathbf{9.8} \pm 2.1$ (**87**%)	-33.7 ± 11.9 (75%)

TABLE II

CPU TIME (IN SECONDS) FOR DIFFERENT EXPERIMENTS

Step	$\pi^*_R(T, N_{\max})$ $(0.0, 100)$	$(0.3, 600)$	$(0.5, 600)$
Dec-POMDP→MPOMDP	13	13	13
Get Human Stoc. FSCs	83	2032	6144
Build Robot POMDP	3	178	203
Solve Robot POMDP	2	57	105
Total time	101	2280	6465

$\pi^*_R(0.5, 600)$ were adapting to their behaviors; and 20% felt they needed to adapt to the robot. Last but not least, we asked our subjects to choose one robot policy which makes they feel comfortable during the collaboration task, 60% of our subjects chose $\pi^*_R(0.3, 600)$ and 40% on $\pi^*_R(0.5, 600)$, but no one choose the robot policy $\pi^*_R(0.0, 100)$.

2) Quantitative results: The average cumulative rewards and success rates of human subjects are shown in the right part of Table I. For the same three robot policies, the cumulative rewards are lower than with synthetic human experiments (-55.3, -20.8 and -33.7 respectively). There are multiple reasons causing this drop of values: first, the human subjects needed some rounds to get familiar with the game, therefore, in early rounds, human usually make more mistakes and received penalties; moreover, since human subjects were not informed of the exact reward function, they were not aware of penalties (negative rewards) obtained when waiting or when going outside of the grid-world.

On the other hand, the observed success rates of human subjects with robot policies $\pi^*_R(0.3, 600)$ and $\pi^*_R(0.5, 600)$ are 81% and 75% respectively. This shows that, even if the cumulative rewards are low, or if the human performs mistakes, $\pi^*_R(0.3, 600)$ and $\pi^*_R(0.5, 600)$ could still help the human to accomplish the task most of the time. The higher success rate (and cumulative reward) with $\pi^*_R(0.3, 600)$ than with $\pi^*_R(0.5, 600)$ (contrary to the results with synthetic humans obtained with $T = 0.5$) suggests that the robot's mental model of humans is better when it builds human FSCs using $T = 0.3$ than $T = 0.5$.

VII. CONCLUSION

In this paper, we address the problem of uncertainty over human's objectives in human-robot collaboration. The contribution is twofold: 1) First, we discuss the chicken-and-egg problem in the second-order mental model, and provide a method to overcome this obstacle and automatically generate a human FSC model which aggregates various possible behaviors for each human objective. Parameters can be tuned to adjust the diversity of the generated human behaviors. 2) Second, we propose a robust robot planning algorithm that relies on a POMDP with uncertainties on the human's actual objective and behavior. We formally detail how to build this robot POMDP based on the task models (Dec-POMDPs) and a distribution over human stochastic policies (FSCs), one per possible objective. Note that the robust planning algorithm (2) does not depend on how the human policies are derived (1). Also, the human policies only serve as the robot's mental model of the human (and to conduct some experiments); actual humans may behave differently.

Through experiments, we demonstrate that our approach is robust to uncertain human behaviors with different objectives. While our scenario considered the same task but with different human preferences, different tasks could be included in the same way. We believe this work is important for collaboration settings where the robot and the human need to reason on each other's possible actions, and where considering myopic or deterministic human policies is not sufficient to generate robust robot policies. Moreover, our approach only requires a Dec-POMDP describing the human-robot collaboration task, no prior human behavior being needed. This makes our method generic to tackle different human-robot collaboration problems, the main issue being its scalability when facing large problems. Future work will focus on scaling up the approach, and allowing to replace the exact Dec-POMDP model by a simulator, *e.g.*, by relying only on simulation-based solvers. We also plan to conduct further experiments, and, in particular, implement our approach on a real scenario where a drone has to help a human to repair devices.

REFERENCES

[1] A. M. Zanchettin, N. M. Ceriani, P. Rocco, H. Ding, and B. Matthias, "Safety in human-robot collaborative manufacturing environments: Metrics and control," *IEEE Trans. on Automation Science and Engineering*, vol. 13, no. 2, 2016.

[2] K. R. Guerin, C. Lea, C. Paxton, and G. D. Hager, "A framework for end-user instruction of a robot assistant for manufacturing," in *ICRA*, 2015.

[3] X. V. Wang, Z. Kemény, J. Váncza, and L. Wang, "Human–robot collaborative assembly in cyber-physical production: Classification framework and implementation," *CIRP Annals*, vol. 66, no. 1, pp. 5–8, 2017.

[4] A. M. Zanchettin and P. Rocco, "Path-consistent safety in mixed human-robot collaborative manufacturing environments," in *IROS*, 2013.

[5] M. G. Jacob, Y.-T. Li, G. A. Akingba, and J. P. Wachs, "Collaboration with a robotic scrub nurse," *Com. ACM*, vol. 56, no. 5, pp. 68–75, May 2013.

[6] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," in *NIPS*, 2016.

[7] Y. You, V. Thomas, F. Colas, R. Alami, and O. Buffet, *Robust robot planning for human-robot collaboration*, 2023. DOI: 10.48550/ARXIV.2302.13916. [Online]. Available: https://arxiv.org/abs/2302.13916.

[8] A. Tabrez, M. Luebbers, and B. Hayes, "A survey of mental modeling techniques in human–robot teaming," *Current Robotics Reports*, vol. 1, Dec. 2020.

[9] S. Sreedharan, T. Chakraborti, C. Muise, and S. Kambhampati, "Expectation-aware planning: A unifying framework for synthesizing and executing self-explaining plans for human-aware planning," in *AAAI*, vol. 34, 2020, pp. 2518–2526.

[10] V. V. Unhelkar, S. Li, and J. A. Shah, "Decision-making for bidirectional communication in sequential human-robot collaborative tasks," in *HRI*, 2020.

[11] S. Nikolaidis, Y. X. Zhu, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in shared autonomy," in *HRI*, 2017.

[12] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, "Planning with trust for human-robot collaboration," in *HRI*, 2018.

[13] ——, "Trust-aware decision making for human-robot collaboration: Model learning and planning," *J. Hum.-Robot Interact.*, vol. 9, no. 2, Jan. 2020. DOI: 10.1145/3359616.

[14] P. Doshi and P. Gmytrasiewicz, "A framework for sequential planning in multi-agent settings," *JAIR*, vol. 24, Jul. 2005.

[15] W. Zheng, B. Wu, and H. Lin, "POMDP model learning for human robot collaboration," in *CDC*, 2018.

[16] V. V. Unhelkar and J. A. Shah, "Learning models of sequential decision-making with partial specification of agent behavior," in *AAAI*, 2019.

[17] S. Russell, "Learning agents for uncertain environments (extended abstract)," in *COLT*, 1998.

[18] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *ICML*, 2000.

[19] N. Meuleau, K.-E. Kim, L. Kaelbling, and A. Cassandra, "Solving POMDPs by searching the space of finite policies," in *UAI*, 1999.

[20] D. V. Pynadath and M. Tambe, "The communicative multiagent team decision problem: Analyzing teamwork theories and models," *JAIR*, vol. 16, Jun. 2002.

[21] Y. You, V. Thomas, F. Colas, and O. Buffet, "Solving infinite-horizon dec-pomdps using finite state controllers within JESP," in *33rd IEEE International Conference on Tools with Artificial Intelligence, (ICTAI)*, IEEE, 2021, pp. 427–434. DOI: 10.1109/ICTAI52525.2021.00069.

[22] M. Grześ, P. Poupart, X. Yang, and J. Hoey, "Energy efficient execution of POMDP policies," *IEEE Trans. on Cybernetics*, vol. 45, 2015. DOI: 10.1109/TCYB.2014.2375817.

[23] E. A. Hansen and S. Zilberstein, "LAO*: A heuristic search algorithm that finds solutions with loops," *Artificial Intelligence*, vol. 129, no. 1, pp. 35–62, 2001.

[24] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *RSS*, 2008.

[25] D. Silver and J. Veness, "Monte-Carlo planning in large POMDPs," in *NIPS*, 2010.

[26] Y. You, V. Thomas, F. Colas, R. Alami, and O. Buffet, *ANR flying co-worker project*, https://gitlab.inria.fr/anr-fcw/robustrobotplanningoffline, 2023.