Partially Observable Markov Decision Process for Managing Robot Collaboration with Human

Abir-Beatrice Karami, Laurent Jeanpierre, Abdel-Illah Mouaddib

Partially Observable Markov Decision Process for Managing Robot Collaboration with Human

Abir-Beatrice Karami and **Laurent Jeanpierre** and **Abdel-Illah Mouaddib**
GREYC-CNRS/University of Caen
Boulevard Marechal Juin, BP5186, 14032 Caen cedex
akarami,mouaddib,laurent@info.unicaen.fr

Abstract

We present a new framework for controlling a robot collaborating with a human to accomplish a common mission. Knowing that we are interested in collaboration domains where there is no shared plan between the human and the robot, the constraints on the decision process are more challenging. We study the decision process of a robot agent for a specific shared mission with a human considering the effect of the human presence, the planning flexibility according to human comfortability and achieving mission. We choose to formalize this problem with Partially Observable Markov Decision Process, then we describe a new domain example that represent human-robot collaboration with no shared plan and we show some preliminary results of solving the POMDP model with standard optimal algorithms as a base work to compare with state-of-the-art and future-work approximate algorithms.

1. Introduction

A lot of research interest in human-robot interaction domains is focusing nowadays on assistant robots, in [**?**] assistant robots for elderly people are presented as part of the Nursebot Project, and the increasing success of those research is leading to more trust of robots not only entering our daily life but also to help those of us who have cognitive disabilities [**?**]. Those kinds of applications explain the need to develop systems with increased capability of operating autonomously, and increased flexibility to be able to co-exist with humans around them.

In this paper we are interested in robot planning while simply sharing a mission with a human but not sharing a plan with him, adding an extra challenge which is the uncertainty of the human's next action and his intentions. The robot planning for best action to perform should consider the frequent change in main environment variables, and as robots exist to help not to disturb, it should dis-consider actions that would incommode the human. For that reason, it is very important to find the best way to represent the environment and human information (acts and movements) in order to interpret his intention into the decision process.

We propose a Partially Observable Markov Decision Process POMDP framework for addressing this problem, knowing that POMDP's are one of few decision models that handles uncertainty. We suggest a domain example that fits the human-robot collaboration with no shared plan. This example is quite limited, but it could be applied to real-life examples like moving in/out of a flat, filling commands in warehouses or cleaning an area.

2. Decision Making with POMDP's

The uncertainty about the human's intention raise the need to observe information about the human that would help in constructing a better belief over his intentions, and use that to build a policy that will take into consideration the human's intention.

The POMDP standard model: A Partially Observable Markov Decision Process POMDP [**?**] relies on a probabilistic model that is represented by a tuple $\langle S, A, T, Z, O, R, b_0 \rangle$, where S is a finite set of states that represent the environment for an agent; A is a finite set of the agent's actions; T is a state transition probability distribution, $T(s, a, s') = Pr(s_t = s' | s_{t-1} = s, a_{t-1} = a)$ is the probability of transitioning from state s to state s' after doing action a, where $\sum_{s' \in S} T(s, a, s') = 1 \forall (s, a)$; Z is a finite set of observations; O is a discrete probability distribution over Z, $O(a, s', z) = Pr(z_t = z | a_{t-1} = a, s_t = s')$ is the probability of observing z from state s' after making action a, where $\sum_{z \in Z} O(a, s', z) = 1 \forall (a, s')$; R is the reward function mapping $S \times A \times S$ to a real number that represent the agent's immediate reward for making action a while being in state s and ending in state s'; given that the state is not directly observable, the agent instead main-

tains a belief distribution over S, b_0 is the initial state probability distribution, $b_t(s)$ is the probability that the system is in state s, given the history of all observations/actions the agent received/affected and the initial belief state b_0, $b_t(s) = Pr(s_t = s|z_t, a_{t-1}, z_{t-1}, ..., a_0, b_0)$.

Optimal Policies: Knowing the last action a_t and last observation z_{t-1}, the agent calculates a new belief state at each time step t by applying the belief update function:

$$\begin{aligned} b_t(s) &= \tau(b_{t-1}, a_t, z_{t-1}) \\ &= \frac{Pr(z_{t-1}|s', a_t, b_{t-1}) \, Pr(s'|a_t, b_{t-1})}{Pr(z_{t-1}|a_t, b_{t-1})} \end{aligned}$$

The objective of the agent is to calculate a policy $\pi_{POMDP} : b_t \rightarrow a$ which assigns for each possible belief an optimal action that maximizes the long-term expected reward $E[\sum_0^\infty \gamma^t r_t]$ where γ is a discount factor and r_t is the reward at time t.

Difference from MDP's: An MDP is a fully observable Markov Decision Process represented by a tuple $\langle S, A, T, R \rangle$ where the state of the environment at any time t is completely observable. An MDP policy maps each *actual state* of the system to an action $\pi_{MDP} : s_t \rightarrow a$ which is the reason of the low complexity compared to POMDP's.

Solving POMDP's: One of the most famous optimal approaches to find POMDP policies is the value iteration approach, where iterations are applied in order to compute more accurate values for each belief state b depending on a chosen action and best reward the agent could receive up to time T. Equation (**??**) describes the value function:

$$V^*(b) = max_a \left[\sum_{s \in S} b(s) R(s,a) + \gamma \sum_{z \in Z} Pr(z|a,b) V^*(\tau(b,a,z)) \right] \quad (1)$$

where, $V_0(b) = max_a \sum_{s \in S} b(s) R(s,a)$. Once iterations lead to a convergence, an optimal policy is defined by mapping the action that gives the maximum value given by $V(b)$. Other algorithms exist for solving POMDP's optimally [**?**], but the enormous computational complexity of those optimal algorithms has been an obstacle toward applying POMDP's to practical problems. Due to that, a wide variety of approximate methods has been developed like Point-Based value Iteration (PBVI) [**?**], Forward Search Value Iteration (FSVI) [**?**], Heuristic Search Value Iteration (HSVI) [**?**], Topological Orders Based Planning [**?**] and others.

POMDP model for our domain: In order for a robot to be able to plan through a shared mission with a human, it should access information that will help it succeed in its mission. Those information should include details about the robot (its position, power status...), details about the human (his position, his possible intentions...) and finally details about the environment (status of objects, navigation constraints...), $S = S_R \times S_H \times S_E$. As there is no shared plan between the robot and the human, information that represent human's intention are the most important in our model. Supposing that robot actions are fully deterministic: the transition function shows uncertainty about the end state as a cause of the uncertain changes caused by the human, i.e. the end state in the transition function $T(s, a, s')$ will include only one possible s'_R with any possible combination of s'_H and s'_E. Observations will help in recovering part of this uncertainty. Observed information about the human or the changes in the environment will help the robot to update its knowledge about the state and also update its belief over the human's intention. What is different in this model is that the observation function does not depend on the action of the robot, instead it depends only on the observation the robot gets about the human and the possible end state. Thus, $O(s', z) = Pr(z_t = z|, s_t = s')$. Depending on the robot's belief about the human's intention, the reward function will lead the robot into choosing the right action knowing that the reward function must be defined in a way that motivates the robot for doing actions that respect the human's intention and avoid conflict actions with possible human intentions.

3. Formalizing Arranging Objects Mission

In this section we will describe a domain example that represent a robot planning in a shared mission with a human, in Fig. **??** we present the arranging objects shared mission environment, it consists of a robot, a human, a box and some scattered objects that should be gathered in the box. Mission is considered accomplished when all the objects are put in the box. We remind that the decision process has no control over the human actions. The mission can be

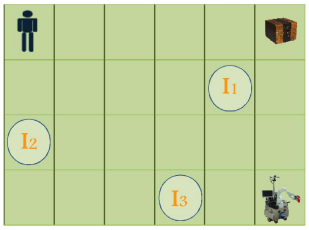

Figure 1: Arranging objects mission Environment.

divided into subtasks; we refer to a subtask the fact that robot/human intend a specific object, moves toward its position, carries it, moves toward the box position and drops it. There is no preference upon the objects: they may be handled in any order. However, generalizing to any priority or manipulation restriction (for example, only the human should handle crystal glasses) is just a matter of tweaking the reward function.

During the mission, the robot should respect the human's

intention to complete a specific subtask: Having no shared plan, the robot will not be able to know for sure what subtask the human has chosen, the only information that will help is an observation that will be received by the robot at each time step holding the human's last action. In other words, the robot will observe the changes related to human(new position and environment changes) and will use that to build a belief over all possible intentions(subtasks). From here, the robot will be able to choose the best action (towards the best subtask) following the calculated policy π. Another condition conclude that the robot existence should not be a burden for the human towards accomplishing his subtask: This can be the case if the robot kept standing as an obstacle in the human's path.

The robot's belief over the human's intended subtask is subject to the observations that the robot receives, this belief might not be exactly true as cause of bad belief update, or maybe a sudden change of the human's real intention. Our goal of this work is to compute a policy that would adapt fast enough to any possible change in the environment's variables including human's intention with respect to mission's success.

State Representation: The state space is characterized as $S = S_R \times S_H \times S_E$ where: S_R includes robot's position $r_{(x,y)}$ and his status r_{st}(if he is carrying an object or not), S_H includes human's position $h_{(x,y)}$, his status h_{st} and his intended subtask h_{in} and S_E includes *constant* objects positions, *constant* box position and the status of each object $e_{(i1,i2,i3)}$ (still on floor or already in the box). The human intention can be subtask to pick up any of the still on ground objects or a non specified intention. **Actions:** $A = \{south, west, north, east, wait\}$, possible actions for the robot are: moving south, moving west, moving north, moving east and do nothing. **Transition Function:** The effects of the robot action a on a state s are relatively clear, however, the transition from state s to state s' is not only defined by the robot action. Human unknown actions have similar affects on the state transition. As result, the transition function is the sum of all possible end states given all possible human actions at state s. $T(s, a, s') = Pr(s'|s, a)$. Where all possible end states s' has the same changes caused by the robot's action, but each one of them has a different possible change caused by the human (new human position, change in items status). Given $s = r_{(x,y)}, r_{st}, h_{(x,y)}, h_{st}, h_{in}, e_{(i1,i2,i3)}$, the $Pr(s'|s, a) =$

$$Pr(< r'_{(x,y)}, r'_{st}, h'_{(x,y)}, h'_{st}, e'_{(i1,i2,i3)}|s, a) \times \sum Pr(h'_{in}|s). \quad (2)$$

In case the robot has any information about the human's policy, a way to represent that in the model is to build a Completely Observable Markov Decision Process MDP with state representation including only the human position and items status and actions similar to robot actions and integrate the information about the humans policy in the transition function and the reward function of this MDP. Then, the state value for certain task in the human MDP can be used in calculating $Pr(h'_{in}|s)$ in eq. **??**. **Rewards:** There are two different rewards received, $R(s, a, s') = reward > 0$ if an object goes into the container by robot or human, $R(s, a, s') = highreward$ When the last object goes to the container by robot or human. On the other hand, the robot receives a penalty in one of these cases: $R(s, a) = penalty < 0$ if a is an impossible moving action due to obstacle existence, $R(s, a) = highpenalty$ if robot picked up an item that was intended by human or if a led the robot to the same position as human intend to be. **Observations:** At each time step, the robot observes the changes in human position, human status, environment information. An observation can be any possible combination of those three type of information $z = \langle h_{(x,y)}, h_{st}, e_{(i1,i2,i3)} \rangle$. **Observation Function:** Depending only on the end state s', the probability for each observation:

$$Pr(z|s') = \begin{cases} 1 & z = s'_{(h_{(x,y)}, h_{st}, e_{(i1,i2,i3)})} \\ 0 & else \end{cases}$$

4. Tested Solvers & Experimental Results

In this paper we present results for three solvers: belief-MDP [**?**], QMDP [**?**] and MMDP [**?**]. We note that this paper does not discuss results of new approximate state-of-the-art approaches, **QMDP:** Q-Function MDP is a POMDP approximation based on a fully observable MDP, where:

$$Q^*_{MDP}(s, a) = R(s, a) + \gamma \sum_{s' \in S} Pr(s'|s, a) V^*_{MDP}(s')$$

The QMDP is then used to calculate the value of each belief state depending on action:

$$\hat{V}(b) = max_{a \in A} \sum_{s \in S} b(s) Q^*_{MDP}(s, a)$$

Belief MDP: an approximation for POMDP by representing continuous belief space with a discretized number of values (eg. n values discretized equally over the belief space). **MMDP:** An MMDP $M = \langle \alpha, A_{i\{i \in \alpha\}}, S, T, R \rangle$ where α is a finite collection of n agents and $A = \times A_i$ represents possible joint action by n agents. Considering the joint action space as the basic set of actions, an MMDP can be viewed as a standard (single-agent) MDP, and can be solved to an optimal joint policy using standard MDP solvers like value iteration. We can manipulate a Multi-agent MDP approach in a way that we handle the POMDP uncertainty of the human information by representing MMDP with a joint action $A = A_R \times A_H$ consists of the robot action and the human action, which means that the solver calculates the optimal joint policy for both of

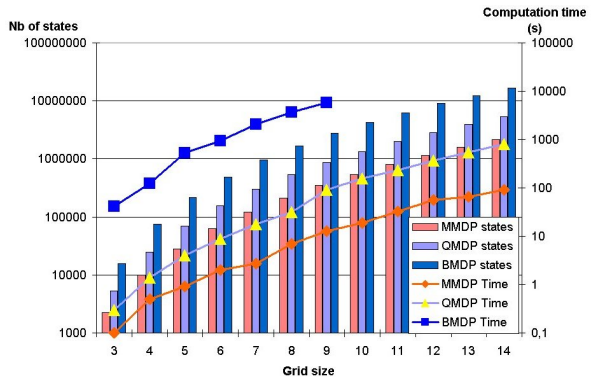

Figure 2: State space and computation time for grid sizes.

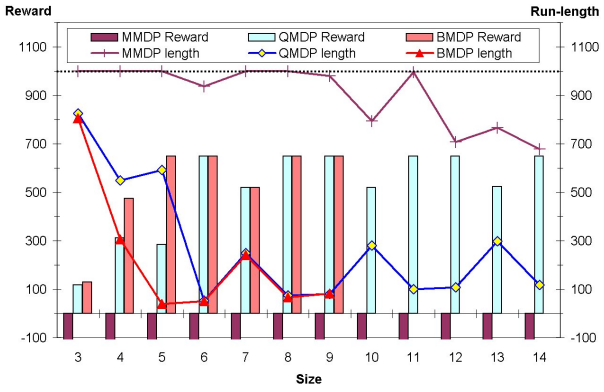

Figure 3: Policy results with random Human behavior.

them. Later at run time, we apply only the robot's optimal action from the optimal policy's joint action, but the human does not necessarily follow the human action part from the policy's optimal joint action.

We present some results we obtained with a simulator written in Java running on 64 bits Linux and Sun's JDK6 with 2GB of memory. Figure **??** compares the number of states and computation time for each of the solvers. We implemented and tested 2 different human behaviors: following random policy and following closest-first policy. In the first case, the human agent chooses a random action from the 5 possible ones, which implies worst case scenario. In the second behavior, the human just go and fetch the closest object, moving through shortest path to take the object, and then returning to the container. Of course, the robot is not informed of the chosen human strategy. Figure **??** shows the test results for the three solver's policies given a random human behaviour. We can see that the MMDP behaves badly in all the trials and generating lots of human-robot conflicts. On the other hand, the results in the case when human follows closest-first policy are much more better and almost alike for all three solvers, where MMDP behaves much better than in the first case.

5. Conclusion & Future Work

We have formalized the addressed problem using a POMDP that relies on a partially observable state of the human. The model has been solved with Belief-MDPs, QMDPs and MMDPs, and it has shown a great potential, at the expense of a large computational cost. The addressed problem is exponential, we are aware that results will be much more interesting in time and state space using special approximate solvers or divide-and-conquer approaches like Policy-Contingent-Abstraction PolCA+ [**?**], and we leave the proof and details for later work.

In case we succeed in future work to handle missions with bigger subtask space(more objects), the agent can maintain a belief over the human's different possible policies (possible chain of tasks). This belief will be updated at each time the human is done with a subtask, which will give the robot more clear belief about the human's policy specially in an advanced time step during the mission.