Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Integration of Planning with Recognition for Responsive Interaction Using Classical Planners

Richard G. Freedman and **Shlomo Zilberstein**
College of Information and Computer Sciences
University of Massachusetts
Amherst, MA 01003, USA
{freedman,shlomo}@cs.umass.edu

Abstract

Interaction between multiple agents requires some form of coordination and a level of mutual awareness. When computers and robots interact with people, they need to recognize human plans and react appropriately. Plan and goal recognition techniques have focused on identifying an agent's task given a sufficiently long action sequence. However, by the time the plan and/or goal are recognized, it may be too late for computing an interactive response. We propose an integration of planning with probabilistic recognition where each method uses intermediate results from the other as a guiding heuristic for recognition of the plan/goal in-progress as well as the interactive response. We show that, like the used recognition method, these interaction problems can be compiled into classical planning problems and solved using off-the-shelf methods. In addition to the methodology, this paper introduces problem categories for different forms of interaction, an evaluation metric for the benefits from the interaction, and extensions to the recognition algorithm that make its intermediate results more practical while the plan is in progress.

1 Introduction

In order for agents, whether they are robots, computers, or humans, to interact with people in the world around them, it is important that they are not just aware of the people's presence, but also able to understand what those people are doing. In particular, interaction involving multiple agents requires some form of coordination during plan execution. In some cases, coordination could be achieved by communication or executing a precise predetermined plan, but other times interaction occurs spontaneously based solely on observations (such as aiding someone who seems to be struggling with a task). Maeda et al. (2014) collected data from human teams performing tasks that enabled robots to learn probabilistic motion primitives to mimic the interaction. Their form of interaction is *reactive*, serving as a low-level reflex to the partner's motions. We instead focus on *responsive* interaction, considering the partner's activities at a higher level and deciding how to act alongside her. Such coordination must account for the uncertainty about the duration and manner in which agents execute their actions.

The fields of *plan* and *goal recognition* have produced methods for identifying an observed agent's task or goal, given her action sequence. This is often regarded as the inverse of *planning* in which, given a set of goal conditions representing a task, the agent aims to derive a sequence of actions that will achieve the conditions when performed from a given initial state. Although planning was one of the earliest research areas within artificial intelligence, recognition was not investigated as rigorously until recently, and the two problems were not initially interconnected. Classical planning was focused on formulating tasks (Fikes and Nilsson 1971; McDermott et al. 1998; Younes and Littman 2004) and developing reasoning and search methods for solving these formulated problems (Hart, Nilsson, and Raphael 1968; Hoffmann and Nebel 2001; Helmert 2006), but early work in plan and goal recognition was motivated by extracting logical information from text documents to derive explanations of the described tasks (Kautz 1991; Charniak and Goldman 1991). The logical databases of facts did not often resemble the formulations used in classical planning until Geib and Steedman (2007) provided a formal relation between natural language parsing and plan recognition using hierarchical task networks (Erol, Hendler, and Nau 1994).

Despite this past separation between planning and plan recognition, the *integration of these areas is crucial for interacting well with others*. Recognizing the plans and goals of those with whom an agent is interacting provides a context for the task and an expectation of how the other agents are approaching it; planning within a similar context and considering the other agents' potential actions as constraints allows the observing agent to develop a sequence of actions that work well with everyone else. This is not a one-directional relationship because the planned actions may not be exactly correct, or the other agents may vary their behaviors and invalidate the observed agent's responding plan. Thus plan recognition must still be performed to confirm the validity of the current action choices and update the information for potential replanning.

Some approaches to such interactive systems have been developed, for example by creating action tables with precalculated responses to recognized events (Koppula and Saxena 2013; Kelley et al. 2012), specifying branching paths of a plan where an interactive agent recognizes the path taken and then plans for actions that will resolve missing causal links for upcoming actions along the chosen path (Levine and Williams 2014), or querying the observed agent about hypothesized activities and then asking additional questions about independent subtasks that it may perform as assistance (Geib, Craensen, and Petrick 2016). The latter two methods used both planning and plan recognition in tandem, but both still have their limitations for the timing and forms of interaction. Levine and Williams's work only allows interaction at specific moments of an orchestrated plan assigned to the observed agent. This is reasonable for the factory domains that motivate the approach, but general interaction in daily life usually lacks a scripted procedure for others to follow as a guide. Moreover, people often vary their approach to a given task or interaction, requiring *plan recognition to be as general as possible* and *planning a response to be as flexible as possible*. Although Geib, Craensen, and Petrick's method allows this flexibility, the negotiation process only enables the observing agent to do what it is permitted, and these permitted tasks are subgoals that do not directly interact with the observing agent. Although many tasks have independent components, Levine and Williams show that the interacting agent can also serve as an extra set of hands during more complicated procedures, providing tools and support without waiting for permission.

We introduce an approach with both these characteristics by compiling intermediate results from Ramírez and Geffner's (2009; 2010) recognition algorithm into a new planning problem. Their recognition approach, which inspired significant follow up work (Chen et al. 2013; Sohrabi, Riabov, and Udrea 2016) and the metrics developed for goal recognition design (Keren, Gal, and Karpas 2014; 2015), established a simple transformation of plan and goal recognition problems into classical planning problems for which efficient off-the-shelf software is available. The outcome is a probability distribution over a set of tasks. However, their strong dependence on the observed agents' optimality makes the recognition most useful as a *post-processing* step when some of the final actions are observed. This is usually too late for effective interaction because the observed agents are nearly finished performing their tasks. We propose an extension to create a foresight effect, making their work more applicable to recognition of plans in progress, which is arguably more useful for human computer/robot interaction.

We start with a formal discussion of the background and recognition approach in Section 2. Then, Section 3 presents a method for extracting information from the intermediate results that allows us to formulate a planning problem that determines a relevant interactive response from the current state. Thus the observing agent may run a second pass of the off-the-shelf planner to determine its own actions to perform during interaction. This section also defines different types of responsive interaction and a metric for a response's amount of assistance. Section 4 discusses minor modifications to the recognition method that may reduce the effect of the optimality bias, permitting foresight when the plan is in-progress with fewer observations. We conclude with an illustration of responsive interaction, a discussion of the effectiveness of the approach, and proposed future directions.

2 Background

A classical planning problem is defined by the tuple $\mathcal{P} = \langle F, I, A, G \rangle$ where F is the set of propositional statements that define the world, $I \subseteq F$ is the initial state describing what is currently true in the world, $G \subseteq F$ is the set of goal conditions that must be satisfied to solve the task, and $a \in A$ is an action that the agent may perform to alter the world's state. Each action specifies these changes via add and delete lists $Add(a), Del(a) \subseteq F$ that respectively add and remove propositions from the current state: $a(s \subseteq F) = (s - Del(a)) \cup Add(a)$. Each action also has a set of preconditions $Pre(a) \subseteq F$ that must be true in the current state before the action is applicable, and it often has a cost $c(a) > 0$. A solution to P is a sequence of actions $a_1, a_2, \ldots, a_x \in A$ called a plan π_G such that $G \subseteq a_x(a_{x-1}(\ldots a_1(I)\ldots)) = \pi_G(I)$ and for all $1 \leq i \leq x, Pre(a_i) \subseteq a_{i-1}(a_{i-2}(\ldots a_1(I)\ldots))$. As there are usually multiple possible plans for a given problem, the objective is to find an optimal plan π^*_G with *minimal* cost:

$$c(\pi^*_G) = \min_{\pi_G} \left\{ \sum_{i=1}^{x} c(a_i) \middle| \pi_G = \bigcirc_{i=1}^{x} a_i \right\}$$

where $\bigcirc$ is the composition over an action sequence.

Plan recognition model Ramírez and Geffner (2010) define the plan recognition problem as triplet $T = \langle \mathcal{P} \setminus G, \mathcal{G}, O \rangle$ where $\mathcal{P} \setminus G$ is the above planning problem without the specified goal conditions. Instead, there is a set of potential goal conditions $\mathcal{G} \subseteq 2^F \setminus \emptyset$ (referred to as a *domain* to differentiate itself from a *library* that describes specific states that satisfy at least one set of goal conditions) for which one is being approached via the strictly ordered (and possibly incomplete) observed action sequence $O \in A^{|O|}$ (Sohrabi, Riabov, and Udrea (2016) alternatively observe fluents in the sequence of states $O \in F^*$). The objective is to determine which $G \in \mathcal{G}$ the agent is trying to solve based on O, and the probabilistic version determines a distribution over $\mathcal{G}$ of the likelihood of each goal condition given the observations.

Notation and problem statement For clarity, we use a slightly different notation to transform T into a planning problem $\mathcal{P}_T$: $F_T = F \cup \{p_i \,| 0 \leq i \leq |O|\}$, $I_T = I \cup \{p_0\}$, and A_T is the same as A with modified Add_T:

$$Add_T(a) = Add(a) \cup \{p_{i-1} \rightarrow p_i \,| a = o_i \in O\},$$

where the new propositions p_i represent the perception of action $o_i \in A$ from sequence O. We include p_0 in the initial state as the initiation of perception and then the new add effects enforce the ordering of the observation sequence O. For each set of goal conditions $G \in \mathcal{G}$, a classical planner now solves for two modified sets of goal conditions. $G + O = G \cup \{p_{|O|}\}$ must satisfy both the observations and the goal, and $G + \overline{O} = G \cup \{\neg p_{|O|}\}$ must satisfy the goal without all the observations. These new planning problems of the form $\mathcal{P}^{G+O}_T = \langle F_T, I_T, A_T, G + O \rangle$ and $\mathcal{P}^{G+\overline{O}}_T = \langle F_T, I_T, A_T, G + \overline{O} \rangle$ allow a comparison of how much effort is necessary to solve the task with respect to O because a rational agent is assumed to be optimal. That is, adhering to the observed actions for a relevant task implies that $c\left(\pi^*_{G+O}\right) < c\left(\pi^*_{G+\overline{O}}\right)$ because O must be performed to optimally solve G, but performing these actions for an irrelevant task instead implies that $c\left(\pi^*_{G+O}\right) > c\left(\pi^*_{G+\overline{O}}\right)$ since G can be solved optimally without performing O.

The comparison is formally presented using Bayes's Rule:

$$\mathrm{P}\left(G \in \mathcal{G} \left| O\right.\right) = \alpha^{-1} \mathrm{P}\left(O|G\right) \mathrm{P}\left(G\right),$$

where α is the normalizing constant. Ramírez and Geffner assume that the prior $\mathrm{P}\left(G\right)$ is uniform and the likelihood $\mathrm{P}\left(O\left|G\right.\right)$ is a Boltzman distribution over two states (capturing the task relevance comparison above). Specifically,

$$\mathrm{P}\left(O\left|G\right.\right) = \sum_{\pi_{G+O}} \mathrm{P}\left(\pi_{G+O}\left|G\right.\right) \approx \alpha'^{-1} exp\left(-\beta \cdot c\left(\pi^*_{G+O}\right)\right)$$

where $\alpha' = exp\left(-\beta \cdot c\left(\pi^*_{G+O}\right)\right) + exp\left(-\beta \cdot c\left(\pi^*_{G+\overline{O}}\right)\right)$ is its normalization constant and β is a predefined constant. Therefore, after using an off-the-shelf classical planner to find π^*_{G+O} and $\pi^*_{G+\overline{O}}$ for each $G \in \mathcal{G}$, we compute all the likelihoods for the probability distribution over $\mathcal{G}$.

3 Deriving an Interactive Response

Following the probabilistic recognition process described in Section 2, the observing agent R_{ing} may plan her own responses with respect to her predictions of the observed agent R_{ed}. First, we use the distribution to extract the goal conditions that R_{ed} most likely intends to solve.

Definition 1. *The* **necessity** *of a proposition p with respect to the set of tasks $\mathcal{G}$ is the expected probability that p is a goal condition given a sequence of observed actions O. That is, $N\left(p \in F\left|O\right.\right) = \sum_{G \in \mathcal{G}} \mathrm{P}\left(G\left|O\right.\right) \cdot \mathbf{1}\left(p \in G\right)$ where $\mathbf{1}$ is the indicator function.*

A necessity of 1 implies that all tasks with probability > 0 require p as a goal condition and 0 implies that no predicted goals require p as a condition. Then for some threshold τ, we define $\widehat{G_{R_{ed}}} = \left\{p\left|N\left(p\left|O\right.\right) \geq \tau\right.\right\}$ as the estimated goal conditions that R_{ed} is trying to satisfy. Using this estimation in addition to her own set of actions $A_{R_{ing}}$ and current state I_{now}, we may define the responsive interaction problem as a centralized multiagent planning problem $\mathcal{P}_{R_{ing}} = \left\langle F' \cup F_{R_{ed}+R_{ing}}, I', A_{R_{ed}+R_{ing}}, G_{R_{ing}}\right\rangle$. Specifically, we propose three forms of responsive interaction:

Definition 2. **Assistive Interaction** *means that R_{ing}'s goal is to only help R_{ed} accomplish her goal. This planning problem $\mathcal{P}^{Assistive}_{R_{ing}}$ is of the form $F' = F$, $I' = I_{now}$, $A_{R_{ed}+R_{ing}} = A_{R_{ed}} \cup \{\text{no-op}\} \times A_{R_{ing}} \cup \{\text{no-op}\}$, and $G_{R_{ing}} = \widehat{G_{R_{ed}}}$.*

The remaining forms of interaction indirectly use $\widehat{G_{R_{ed}}}$ through a new fluent that denotes whether R_{ed} accomplished these conditions. We call this fluent $success$ and add it through the additional add effect $success \vee \bigwedge_{g \in \widehat{G_{R_{ed}}}} g \rightarrow success$ for each action in $A_{R_{ed}}$, implying that solving the goal conditions once is sufficient to complete the task. We respectively call these modified sets F^S and $A^S_{R_{ed}}$.

Definition 3. **Independent Interaction** *means R_{ing} has a personal goal G' to accomplish, but should avoid preventing R_{ed} from accomplishing her own task at the same time. This planning problem $\mathcal{P}^{Independent}_{R_{ing}}$ is of the form $F' = F^S$, $I' = I_{now}$, $A_{R_{ed}+R_{ing}} = \left(A_{R_{ed}} \cup \{\text{no-op}\}\right)^S \times A_{R_{ing}} \cup \{\text{no-op}\}$, and $G_{R_{ing}} = G' \cup \{success\}$.*

Definition 4. **Adversarial Interaction** *means that R_{ing}'s goal is to prevent R_{ed} from achieving her goal for some duration d. This planning problem $\mathcal{P}^{Adversarial}_{R_{ing}}$ is of the form $F' = F^S \cup \{0, 1, \ldots, d\}$, $I' = I_{now} \cup \{0\}$, $A_{R_{ed}+R_{ing}} = \left(A_{R_{ed}} \cup \{\text{no-op}\}\right)^S \times \left(A_{R_{ing}} \cup \{\text{no-op}\}\right)^{step}$, and $G_{R_{ing}} = \{\neg success, d\}$ where $\{\cdot\}^{step}$ applies an incremental add effect $i \rightarrow (i+1)$, which can be done using fluents in PDDL.*

For each form of interaction, R_{ing} may use the same off-the-shelf classical planner from the recognition step to derive the joint optimal plan

$$\pi^*_{G_{R_{ing}}} = \bigcirc_{i=1}^{y} \left(a_{R_{ed},i}, a_{R_{ing},i}\right)$$

that R_{ing} and R_{ed} should perform alongside each other. However, this plan is optimistic since R_{ed} is acting independently and she *is not guaranteed to follow the joint plan* unless there is direct communication and R_{ing} tells R_{ed} what actions to perform. If there was communication, then it would have been possible for R_{ed} to reveal G to R_{ing} in the first place; so we consider the case where direct communication between the agents is absent. Then R_{ing} can only perform her assigned actions from $\pi^*_{G_{R_{ing}}}$, which we will call

$$\pi_{R_{ing}} = \bigcirc_{i=1}^{y} a_{R_{ing},i}.$$

This introduces the need for replanning with new observations throughout the interaction; we defer identifying when to replan to future work and currently assume that replanning is only performed after R_{ing} completes her actions and the goal is not yet complete. There should be more observations available after performing $\pi_{R_{ing}}$ for more accurate recognition, meaning that $\widehat{G_{R_{ed}}}$ should also become more specific.

Adapting to Responsive Interaction

As R_{ing} performs the actions in her plan $\pi_{R_{ing}}$, new changes to the state will occur that may affect R_{ed}'s performance of π^*_G. Changes from assistive interaction are intended to facilitate R_{ed}'s ability to complete task G, changes from independent interaction should not greatly affect R_{ed}'s plan unless there is a resource conflict to resolve, and changes from adversarial interaction should inhibit R_{ed} from completing G. For all these categories, we can run another execution of the off-the-shelf classical planner to measure the *helpfulness* of $\pi_{R_{ing}}$ using a transformation similar to the one used for creating $\mathcal{P}_T$, but instead simulating $\pi_{R_{ing}}$ in R_{ed}'s centralized multi-agent planning problem. We define $\mathcal{P}_{R_{ed} \leftarrow R_{ing}} = \left\langle F_H, I_H, A_{R_{ed}} \cup \{\text{no-op}\} \times A_{R_{ing},H} \cup \{\text{no-op}\}_{H+}, G\right\rangle$

where $F_H = F \cup \left\{p_i \left| 0 \le i \le y+1 = \left|\pi_{R_{ing}}\right| + 1\right.\right\}$, $I_H = I_{now} \cup \{p_0\}$, and $\{\cdot\}_H$ and $\{\cdot\}_{H+}$ are the modified sets of actions defined by:

- $Add_H(a) = Add_{H+}(a) =$
$$Add(a) \cup \left\{p_{i-1} \rightarrow p_i \left| a = \pi_{R_{ing},i}\right.\right\}$$
- $Del_H(a) = Del_{H+}(a) = Del(a)$
- $Pre_H(a) = Pre(a) \cup \left\{ (p_y \wedge p_{y+1}) \vee \bigvee_{i \in \left\{i \left| a = \pi_{R_{ing},i}\right.\right\}} (p_{i-1} \wedge \neg p_i) \right\}$
- $Pre_{H+}(a) = Pre(a) \cup \left\{ p_y \vee \bigvee_{i \in \left\{i \left| a = \pi_{R_{ing},i}\right.\right\}} (p_{i-1} \wedge \neg p_i) \right\}$

Thus the propositions p_i now represent the performance of each action in $\pi_{R_{ing}}$, and the impossible precondition $(p_y \wedge p_{y+1})$ forces R_{ing} to execute no-ops once her plan is executed — these no-ops may later become actions if replanning is performed. Similar to the derivation of $\pi_{R_{ing}}$, the new actions that R_{ed} will perform as adaptation to R_{ing}'s responsive actions are

$$\pi_{R_{ed} \leftarrow R_{ing}} = \bigcirc_{i=1}^{z} a_{R_{ed},i}$$

where the optimal solution to $\mathcal{P}_{R_{ed} \leftarrow R_{ing}}$ is

$$\pi^*_{G+\pi_{R_{ing}}} = \bigcirc_{i=1}^{z} \left(a_{R_{ed},i}, a_{R_{ing},i}\right).$$

Definition 5. *The* **helpfulness** *of a responsive plan $\pi_{R_{ing}}$ is the change in cost from agent R_{ed} acting on her own to both agents working simultaneously $H\left(\pi_{R_{ing}}\right) = c\left(\pi^*_{G,\ge now}\right) - c\left(\pi_{R_{ed} \leftarrow R_{ing}}\right)$. We assume $c(\pi) = \infty$ if π does not exist.*

Lemma 1. *If R_{ing} knows G, R_{ing} is being assistive or independent, and there exists a non-invasive sequence of actions such that R_{ing} never affects a precondition of any action in π^*_G, then $H\left(\pi_{R_{ing}}\right) \ge 0$.*

Proof. (Sketch) Because R_{ing} knows the correct goal, $G_{R_{ing}} = G$ implying that the problems $\mathcal{P}_{R_{ing}}$ and $\mathcal{P}_{R_{ed} \leftarrow R_{ing}}$ are identical (the additional fluents and action modifications only ensure following the solution). Thus $\pi^*_{G_{R_{ing}}} = \pi^*_{G+\pi_{R_{ing}}}$, and R_{ing} could at least perform the non-invasive sequence of actions as $\pi_{R_{ing}}$ while R_{ed} performed her initial plan such that $c\left(\pi^*_{G+\pi_{R_{ing}}}\right) \le c\left(\pi^*_{G,\ge now}\right)$. Hence $H\left(\pi_{R_{ing}}\right) \ge 0$. □

Lemma 1 shows that with a good prediction from recognition, the observing agent will rarely hinder the observed agent's progress unless the domain and current state force R_{ing} to get in the way. Clearly this should not hold for adversarial interaction because such an agent wants to provide as little help as possible. However, it is more difficult to guarantee not being helpful due to the fact that there is usually more than one (optimal) plan. Thus R_{ed} could perform another sequence of actions that is not 'blocked' by $\pi_{R_{ing}}$ and still solve the task.

Lemma 2. *If R_{ing} knows G, R_{ing} is being adversarial, and there exists an invasive sequence of actions such that R_{ing} always affects a precondition of some action in* every possible *π^*_G, then $H\left(\pi_{R_{ing}}\right) \le 0$.*

Theorem 1. $-\infty < H\left(\pi_{R_{ing}}\right) \le c\left(\pi^*_G\right) - c\left(\pi^*_{R_{ed}+R_{ing}}\right)$ *where $\pi^*_{R_{ed}+R_{ing}}$ is the optimal plan that solves the centralized multi-agent planning problem $\langle F, A_{R_{ed}+R_{ing}}, I, G\rangle$ and $A_{R_{ed}+R_{ing}} = A_{R_{ed}} \cup \{\text{no-op}\} \times A_{R_{ing}} \cup \{\text{no-op}\}$.*

Proof. The lower bound follows from Lemma 2 because R_{ing} could perform some action that permanently prevents R_{ed} from satisfying the preconditions of all actions whose effects satisfy one of G's conditions. The upper bound extends from the proof of Lemma 1; the best case is that both agents work together from the beginning (even if that means R_{ing} does nothing). As each time step progresses, the cost for both the single-agent and multi-agent will decrease uniformly so that $c\left(\pi^*_G\right) - c\left(\pi^*_{R_{ed}+R_{ing}}\right) = c\left(\pi^*_{G,\ge now}\right) - c\left(\pi^*_{R_{ed}+R_{ing},\ge now}\right)$. □

Computational Complexity

For all the forms of responsive interaction, we are running the recognition algorithm that simulates a classical planner twice per possible goal, summing over the possible goals' propositions to compute necessities, and then solving a new planning problem that is generated from the necessities. Hence the complexity is similar for all three forms. As classical planning is PSPACE-complete (Bylander 1994) and the computation of necessities/creation of a new planning goal can be done with negligible memory, their space complexity is PSPACE-complete. The runtime complexity is $O\left((2|\mathcal{G}|+1) \cdot C\right) = O\left(|\mathcal{G}| \cdot C\right)$ where C is the runtime complexity of the chosen off-the-shelf classical planner, which should be far greater than the time used for summing and creating a new goal.

4 Foresight for In-Progress Recognition

All the definitions and methods for responsive interaction in Section 3 rely on the predictions from Ramírez and Geffner's (2010) recognition algorithm. While the recognized distributions are shown to be very effective in their experiments, the performance was best when a reasonable percentage of the actions were observed from the agent's (possibly optimal) plan execution. This is because a greater number of observations increases the probability of including one of the later actions taken in the sequence. Although the earlier and intermediate actions of a plan play a role in the recognition algorithm, the later ones impose constraints near the goal-satisfying states that will require an agent to go out of her way to follow/avoid the observed action(s). This distinguishing action is what motivates goal recognition design (Keren, Gal, and Karpas 2014), revealing the observed agent's intentions as early as possible.

The influence of the most-recent observation is visually evident in Figs. 1 and 2 of Ramírez and Geffner's (2010) work — the values of $\mathrm{P}(G)$ for each step of a noisy random walk produced a plot whose goal location(s) with the greatest probability was(were) closest to the current location of the agent at that time step. Thus the approach is greedy in the sense that the observed agent's optimality assumption motivates finishing the task as soon as possible. This creates a bias towards recognizing locally short-term plans and avoiding tasks that require a greater cost. The bias is given further emphasis in the extension by Sohrabi, Riabov, and Udrea (2016) since their adapted function for $\mathrm{P}(O|G)$ uses a weighted value that punishes goals with more unsatisfied conditions during the simulation of the observation sequence, including goals that could not yet be completed. These features are ideal for post-processing when it is assumed that *the observed agent has already completed its plan and satisfied the goal*.

Thus, for recognition while the *plan's execution is in-progress*, effective ways to apply foresight and recognize long-term plans earlier is necessary in order to properly predict more costly tasks. Otherwise, it may be too late to determine a proper response and/or interact. We propose minor extensions to the computations of the prior used for the Bayes' Rule computation to accomplish this.

Dynamic Prior Using Survival Analysis

The purpose of the prior in Bayesian statistics is to introduce an initial belief, biasing one's expectations of the posterior distribution. The Bayesian update process, keeping track of the observed outcomes and adjusting the prior based on the counts, eventually converges to the true distribution. A more accurate initial prior improves the convergence rate. Using a uniform prior is reasonable for test domains that lack actual users to whom the method can adapt, but this also has the price of always assuming that every task is equally likely to be performed. Thus the likelihood has all the influence in the distribution over $\mathcal{G}$, but it was designed to favor the most optimal plans in this algorithm's case.

We suggest *counter-balancing this greedy likelihood with a prior that is biased towards more costly tasks* that would typically be ignored. This is most important at the beginning of the observation process when enough actions cannot be taken to approach any of the more costly tasks, but it is not beneficial to continue to favor long-term plans after there are enough observations to identify with confidence the task the agent is performing, reversing the direction of the bias towards future tasks when the current ones are the most likely. Hence we propose a *dynamic prior* that favors long-term tasks with greater-cost optimal plans when the plan execution time/resource consumed t is lesser and converges to the true prior $\mathrm{P}(G)$ as t increases, allowing the likelihood's optimality assumption to take precedence. To do this, we revise the probabilty formulation to include t for a joint distribution over the observed agent's task $G \in \mathcal{G}$ and resource consumption $t \in \mathbb{N} \cup \{0\}$:

$$\mathrm{P}(G|O,t) = \alpha^{-1}\mathrm{P}(O|G,t)\,\mathrm{P}(G|t)$$

where $\mathrm{P}(O|G,t) = \mathrm{P}(O_{\leq t}\,|G)$ is the likelihood for the observations seen so far and $\mathrm{P}(G|t) = \alpha''^{\,-1}\mathrm{P}(t|G)\,\mathrm{P}(G)$ is the prior over the joint distribution. Although we let t be the length of the optimal simulated plan that emulates $O_{\leq t}$ in order (so $t = |O|$ if no observations are missing), t may be any resource that can define cost such as clock time (for durative actions and real-time problems), distance, energy consumed, etc.; plan length simply assumes uniform action cost. Then we only need to define the newly introduced probability $\mathrm{P}(t|G)$, which requires survival analysis.

Definition 6. *A plan π that successfully solves a planning problem with goal G is* **in-progress** *at step i if G's conditions are not yet satisfied after performing $\pi_{\leq i}$. If $\pi_{\geq j}$ for $j < |\pi|$ is no longer in-progress, then $\pi_{\geq j}$ is* **maintaining**.

We only consider plans without any maintaining actions. Then a *plan is surviving as long as it is in-progress*, and it ceases execution once the goal conditions are satisfied. Clearly any plan for some goal $G \in \mathcal{G}$ must survive at least $c(\pi^*_G)$ (such as $|\pi^*_G|$ actions) because the optimal solution is the least cost needed to complete the plan. Then, depending on the observed agent's optimality, we will observe some additional discrete amount k spent during the plan's execution. As this is counting a finite quantity of events (additional amount of discrete resources/executed actions) within a specific time window (one plan execution), we assume that the probability of R_{ed}'s plan's cost to solve G is

$$\mathrm{P}(c(\pi) = t\,|G) = \begin{cases} 0 & \text{if } t < c(\pi^*_G) \\ \frac{(\lambda+1)^{t-c(\pi^*_G)+1}}{\left(t-c\left(\pi^*_G\right)+1\right)!\cdot\left(e^{\lambda+1}-1\right)} & \text{otherwise} \end{cases}$$

The Poisson distribution $\mathrm{Poisson}(k \in \mathbb{N} \cup \{0\}\,; \lambda \in \mathbb{N}) = \left(\lambda^k\right)/\left(k!e^{\lambda}\right)$ is used for situations involving counts of events over a fixed time window where the parameter λ is the expected count and standard deviation. So a usually-optimal agent should have parameter $\lambda = 0$ while a less optimal agent should have a greater λ value, spending an expected cost of $(c(\pi^*_G) + \lambda)$. However, the Poisson distribution cannot allow $\lambda = 0$ since the numerator would always be 0; thus we used the positive Poisson distribution

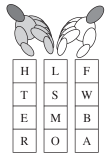
$$\mathrm{Poisson}^+(k \in \mathbb{N}; \lambda \in \mathbb{N}) = \frac{\lambda^k}{k!\left(e^{\lambda}-1\right)}$$

(Singh 1978) in the 'otherwise' case above and incremented the values $k = (t - c(\pi^*_G))$ and λ by 1 to handle the removed 0. Besides the posterior updating prior $P(G)$, λ can be updated as an agent is observed through multiple trials or interactions.

Survival analysis is an area of statistics that determines the probability that something continues to live with respect to a life-expectancy distribution. Thus, given the piecewise equation above as the life-expectancy of a plan solving goal G, the probability of the plan's survival is relative to how likely it is still in-progress for the current cost t:

$$\mathrm{P}(t|G) = \mathrm{P}(c(\pi) \geq t\,|G) = 1 - \mathrm{cdf}_t\,(c(\pi)\,|G)$$

where cdf_t is the cumulative distribution function from 0 to t. The hazard function $h(t) = \mathrm{P}(c(\pi) = t\,|G)\,/\mathrm{P}(c(\pi) \geq t|G)$ also tells us the likelihood that the plan will terminate execution at time t, which could be useful for reasoning about whether the observed agent is almost finished with her task.



HER $(6, 2)$	FATHER $(18, 8)$	FOSTER $(12, 6)$
OTHER $(16, 6)$	LATHER $(18, 8)$	HATER $(12, 6)$
BOTHER $(22, 10)$	**MASTER** $(\mathbf{16}, \mathbf{8})$	LATER $(12, 6)$
MOTHER $(18, 6)$	FASTER $(16, 8)$	WATER $(12, 6)$

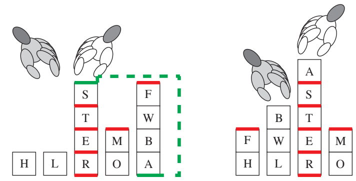


Figure 1: *Left* Initial state and goals with $c\left(\pi^*_G\right)$ and $c\left(\pi^*_{R_{ed}+R_{ing}}\right)$. *Right* States where assistive interaction begins and ends.

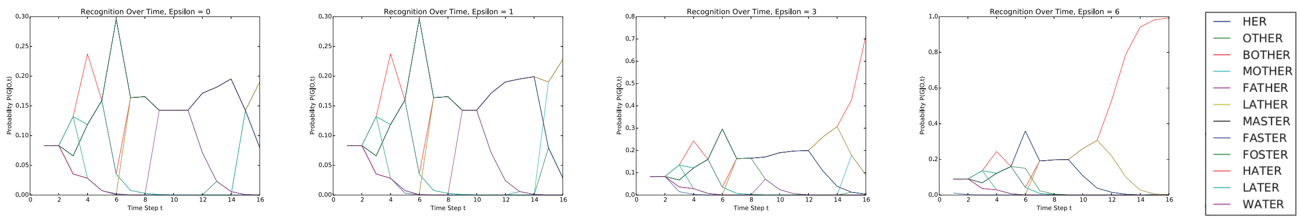


Figure 2: Recognized distribution over $\mathcal{G}$ as each action of π^*_G is taken.

Instead of using the hazard function, we will use just the survival function with *foresight to advance the resource consumption*. When computing the likelihoods, we can find the subplan with the fewest consumed resources to identify a current value for t. Then, because the likelihood will prefer goals that will not consume too many more resources than t, we compute the prior at future time $t + \epsilon$ for some $\epsilon \geq 0$. If there is enough variation between $c\left(\pi^*_G\right)$ for each $G \in \mathcal{G}$, then this will bypass the goals preferred by the likelihood and contribute probability mass to the long-term goals that cannot yet be achieved. Specific assignments for ϵ will vary by many factors such as the variation of the goals' optimal plan costs, time already elapsed (if t becomes too large, then all the goals will be surpassed for a more uniform prior), and preferred amount of look-ahead for interactive purposes.

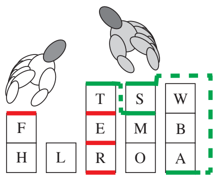
5 Illustrated Example

To demonstrate the process of responsive interaction, we create a problem based on the Block-words domain from Ramírez and Geffner's (2010) dataset[1]. Similar to the traditional Blocksworld, each block contains a letter so that stacks of blocks may spell words from top-to-bottom. Fig. 1 illustrates the initial state and lists the 12 possible goals $\mathcal{G}$ for our example. To extend Block-words for two interacting agents, all actions in $A_{R_{ed}+R_{ing}}$ are performed in parallel when different blocks are involved, but actions that share blocks are modified to account for joint actions: putting down the same block that is picked up is considered a handover, and two consecutive blocks may be picked up or put down in a stack simultaneously. We omit pairs of actions that have race conditions, such as placing a block on top of one that is being picked up.

We walk through a single example of *assistive interaction* and assign values to parameters as needed; adversarial and independent perspectives are presented afterwards as alternative interactions. We assume that R_{ing} only computes one responsive plan without replanning; so the decision of when to join can be evaluated as well. To begin, we implemented our proposed dynamic prior extension to Ramírez and Geffner's (2010) probabilistic recognition algorithm and used the state-of-the-art Fast-Downward planner (Helmert 2006) to solve the compiled problems. Then, R_{ed} uses the planner to find an initial optimal plan π^*_G to solve her goal G: spelling MASTER. As R_{ed} performs each action a_t in π^*_G, R_{ing} observes her and then computes $\mathrm{P}\left(G \,|O, t\,\right)$ where $O_{\leq t} = \pi^*_{G,\leq t}$. These recognition results for several choices of foresight parameter ϵ are shown in Fig. 2. We observe marginal differences for smaller ϵ, but larger ϵ exaggerate the probability of more costly goals as each action is taken. We will use $\epsilon = 3$ because it maximizes the necessities, but the threshold $\tau = 0.3$ (slightly less than the greatest probability given to a goal at the beginning) yields the same propositions for all $\epsilon \leq 5$.

Because all the goals share the same last two letters ER, we note that the necessity for 'stack E on top of R' and 'place R on the table' are always 1. So for any τ, these two conditions will be part of $\widehat{G_{R_{ed}}}$. When the distribution is more uniform at the earlier timesteps, any goal condition that frequently appears in $\mathcal{G}$ has greater necessity. For example, observing the first two actions (R_{ed} takes H off of T and places it on the table) does not disambiguate any of the goals so that the conditions 'stack H on top of E' and 'stack T on top of E' both have necessity 0.5; half the goals end in HER and the other half end in TER. However, both conditions cannot be true simultaneously so that $\widehat{G_{R_{ed}}} = G_{R_{ing}}$ has no solution and R_{ing} is unable to join in yet. After performing the sixth action (R_{ed} places S on top of T), the recognition algorithm identifies MASTER and FASTER as the most likely candi-

[1]Note that the legacy code borrowed for our implementation does not scale to cases larger than the problems used in the International Planning Competition for which it was designed. The competition's problems turned out to be too limited to illustrate our work due to the types of problems available (grid world-like problems do not introduce many interaction opportunities) and their typical solution lengths (R_{ing} would not have sufficient information to interact until R_{ed} completes the short tasks).



State two actions after the assistive interaction began, but with adversary R_{ing}. Stacking S on top of M undid two satisfied necessities while R_{ed} was beginning to uncover the A block.

Figure 3: Illustration of the adversarial perspective.

State four actions after assistive interaction began, but with independent R_{ing}. R_{ed} does not use the framework and thus covers H with F, but R_{ing} also moves M instead of helping R_{ed} uncover A as she did during assistive interaction.

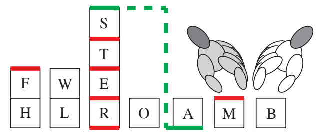


Figure 4: Illustration of independent perspective for G_1.

dates and their shared goal conditions become the only ones with sufficient necessity — greater ϵ also deem their conflicting conditions necessary. The red lines in Fig. 1 show the conditions that are already satisfied, and the green dotted lines show the unsolved condition: 'stack A on top of S.' $\pi_{R_{ing}}$ and $\pi_{R_{ed} \leftarrow R_{ing}}$ complete this together in 4 steps rather than R_{ed} completing it alone in 8 steps. R_{ed} then completes G by stacking M on top. Hence $H\left(\pi_{R_{ing}}\right) = 4$, indicating a helpful assistance in solving the task.

Adversarial Perspective

In the case that this example interaction was adversarial, then we must consider the scenario for several durations d. Recall that R_{ing}'s goal is to ensure that G is not completed within d actions after the current time step. Thus, using the same example and again waiting until the sixth action is performed, R_{ing} may easily accomplish its goal with any arbitrary plan if $d < 2$ and *almost* any arbitrary plan if $d < 10$ because the quickest R_{ed} may accomplish the task is $c\left(\pi^*_G\right) = 16$ alone and $c\left(\pi^*_{R_{ed}+R_{ing}}\right) = 8$ with an assistive partner. We emphasize 'almost' because any $\pi_{R_{ing}}$ that helps to spell G's goal word allows the goal to be completed within the reduced duration if two agents can solve the task within that time.

However, once d is great enough that R_{ed} can solve the plan on her own, then R_{ing} has to use plans that prevent at least one of G's conditions from being satisfied at every time step. For our Block-words example, there is a 'trivial plan' $\pi^{\text{hoard}}_{R_{ing}}$ where R_{ing} picks up one of the required blocks, such as S, and then performs no-ops indefinitely. R_{ed} is unable to obtain the block held by R_{ing} and she can never complete the task; thus $H\left(\pi^{\text{hoard}}_{R_{ing}}\right) = -\infty$. As most off-the-shelf classical planners apply search methods and do not use such logic, we instead expect $\pi_{R_{ing}}$ to continue to pick up and rearrange the blocks that will undo R_{ed}'s progress. Such $\pi_{R_{ing}}$ will usually have a finite negative helpfulness so that it is possible to find a (less optimal) solution $\pi_{R_{ed} \leftarrow R_{ing}}$. Fig. 3 displays the state after unstacking S from the goal word and stacking it on top of M, but R_{ed} can still complete the task if this is the entire $\pi_{R_{ing}}$; thus R_{ing} will eventually need to replan and resume taking the tower of blocks apart. This back-and-forth interaction of R_{ing}'s deconstruction and R_{ed}'s reassembly emphasizes the need for planning and recognition to continuously update each other.

Independent Perspective

Lastly, let us consider two independent interactions where G'_1 is to spell the word HOWL and G'_2 is to spell the word WOLF, both using the same blocks. G'_1 is the simpler case because none of its blocks are involved in the necessities after R_{ed} performs the first six actions. Thus $\pi_{R_{ing}}$ will eventually pick up W and put it on top of L, which helps R_{ed} by reducing the number of actions needed to reach the A block. Then $\pi_{R_{ing}}$ will later pick up O in order to place it on top of W, but R_{ing} will first need to move the M block. R_{ing} will not place M on top of the stack of blocks spelling G's goal word because that will prevent $success$ from becoming true (see Fig. 4) unless A was already put on top of S. Likewise, if R_{ing} had to move other blocks afterwards (though, in this case, she just picks up H and places it on top of O), then she would not place them on top of M because $\widehat{G_{R_{ed}}}$ contains a condition to leave nothing on top of the M block (in case G spells MASTER).

Because $\widehat{G_{R_{ed}}}$ also contains a condition to leave nothing on top of the F block in case G spells FASTER, G'_2 is more difficult to accomplish. The most R_{ing} can do is uncover the H, O, and W blocks and start to stack them. However, stacking them is not practical because she will have to undo the stack in order to place them on top of F later. Thus more observations may be necessary before R_{ing} interacts in this case so that she can later update $\widehat{G_{R_{ed}}}$ and confirm that 'stack nothing on top of F' is not a necessity. Unless R_{ed} stacks a block on top of F, then this would unfortunately require R_{ing} to wait until G is already complete due to the ambiguity of the two goals. This presents an interesting path of future work where *both agents are simultaneously observing each other* for independent interaction.

6 Discussion

Many interactive systems currently employ recognition and planning without leveraging the potential benefits of integrating the two processes. However, machines need to be versatile when cooperating with different people and cannot rely on strong assumptions about their behavior. Thus, it is necessary to consider the integration of recognizing general tasks and responding flexibly with a generated plan. We propose a novel framework for such a system that extends an existing planning-compilation approach to probabilistic recognition. The introduction of a *dynamic prior* allows the distribution over tasks to be less greedy and consider long-term goals for foresight of how to respond. This updated distribution is used to identify the *necessity* of goal conditions so that, depending on the agent's form of interaction, she may create a planning problem whose solution is a response to the currently recognized task(s). We also introduce *helpfulness* as a means of quantifying the effectiveness of the response. Lastly, we explore several examples of interaction to investigate the trade-offs for parameter choices and their effects on the integrated process. This provides initial insights into responsive interaction, which can facilitate the interaction between agents in a wide range of applications.

This work opens up many new research avenues for further exploration, ranging from simple tasks such as identifying conditions for parameter value choices to more complicated challenges such as identifying when to replan. Performing recognition during plan execution will update the necessary goal conditions with more observations and identify whether the observed agent's actions are straying from expectation. While our interaction framework works with plans in-progress, calling off-the-shelf planners multiple times per iteration is a significant drawback, particularly in settings that require real-time interaction. In future work, we plan to examine ways to improve scalability by exploiting the similarity in search spaces between multiple planner calls (Davidov and Markovitch 2006), by using landmarks for generating local intermediate plans (Hoffmann, Porteous, and Sebastia 2004), and by considering the possible role of explicit communication (Goldman, Allen, and Zilberstein 2007). Another direction is to extend the work to handle sensor information for observing actions rather than relying on high-level descriptions. With these improvements, responsive interaction will be possible to compute in realistic settings in near real-time, enhancing the applicability of the framework to complex interactions with humans.

Acknowledgments

We thank the anonymous reviewers for their feedback and Alex Fukunaga for early discussions that inspired this work while the first author collaborated with him on NSF grant 1515258. This work was supported by NSF grant 1405550.

References

Bylander, T. 1994. The computational complexity of propositional STRIPS planning. *Artificial Intelligence* 69(1-2):165–204.

Charniak, E., and Goldman, R. 1991. Probabilistic abduction for plan recognition. Technical Report CS-91-12, Brown University, Providence, RI, USA.

Chen, J.; Chen, Y.; Xu, Y.; Huang, R.; and Chen, Z. 2013. A planning approach to the recognition of multiple goals. *International Journal of Intelligent Systems* 28(3):203–216.

Davidov, D., and Markovitch, S. 2006. Multiple-goal heuristic search. *Journal of Artificial Intelligence Research* 26:417–451.

Erol, K.; Hendler, J.; and Nau, D. S. 1994. HTN planning: Complexity and expressivity. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 1123–1128.

Fikes, R. E., and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. In *Proceedings of the Second International Joint Conference on Artificial Intelligence*, 608–620.

Geib, C. W., and Steedman, M. 2007. On natural language processing and plan recognition. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 1612–1617.

Geib, C.; Craensen, B.; and Petrick, R. P. A. 2016. Combining plan recognition, goal reasoning, and planning for cooperative task behavior. In *Fourth IJCAI Workshop on Goal Reasoning*, 1–8.

Goldman, C. V.; Allen, M.; and Zilberstein, S. 2007. Learning to communicate in a decentralized environment. *Autonomous Agents and Multi-Agent Systems* 15(1):47–90.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2):100–107.

Helmert, M. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research* 26(1):191–246.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 14:253–302.

Hoffmann, J.; Porteous, J.; and Sebastia, L. 2004. Ordered landmarks in planning. *Journal of Artificial Intelligence Research* 22(1):215–278.

Kautz, H. A. 1991. A formal theory of plan recognition and its implementation. In Allen, J. F.; Kautz, H. A.; Pelavin, R. N.; and Tenenberg, J. D., eds., *Reasoning About Plans*. San Francisco, CA, USA: Morgan Kaufmann. 69–125.

Kelley, R.; Tavakkoli, A.; King, C.; Ambardekar, A.; Nicolescu, M.; and Nicolescu, M. 2012. Context-based bayesian intent recognition. *IEEE Transactions on Autonomous Mental Development* 4(3):215–225.

Keren, S.; Gal, A.; and Karpas, E. 2014. Goal recognition design. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling*, 154–162.

Keren, S.; Gal, A.; and Karpas, E. 2015. Goal recognition design for non-optimal agents. In *Proceedings of the Twenty-Ninth International Conference on Artificial Intelligence*, 3298–3304.

Koppula, H. S., and Saxena, A. 2013. Anticipating human activities using object affordances for reactive robotic response. In *Proceedings of Robotics: Science and Systems*, 2071–2078.

Levine, S., and Williams, B. 2014. Concurrent plan recognition and execution for human-robot teams. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling*, 490–498.

Maeda, G.; Ewerton, M.; Lioutikov, R.; Ben Amor, H.; Peters, J.; and Neumann, G. 2014. Learning interaction for collaborative tasks with probabilistic movement primitives. In *Proceedings of the Fourteenth IEEE-RAS International Conference on Humanoid Robots*, 527–534.

McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL – The planning domain definition language – Version 1.2. Technical Report CVC TR-98-003, Yale Center for Computational Vision and Control, New Haven, CT, USA.

Ramírez, M., and Geffner, H. 2009. Plan recognition as planning. In *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence*, 1778–1783.

Ramírez, M., and Geffner, H. 2010. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the Twenty-Fourth AAAI Conf. on Artificial Intelligence*, 1121–1126.

Singh, J. 1978. A characterization of positive poisson distribution and its statistical application. *SIAM Journal on Applied Mathematics* 34(3):545–548.

Sohrabi, S.; Riabov, A. V.; and Udrea, O. 2016. Plan recognition as planning revisited. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 3258–3264.

Younes, H. L. S., and Littman, M. L. 2004. PPDDL1.0: An extension to PDDL for expressing planning domains with probabilistic effects. Technical Report CMU-CS-04-162, Carnegie Mellon University, Pittsburgh, PA, USA.