Plan Recognition as Planning Revisited

Shirin Sohrabi Anton V. Riabov Octavian Udrea
IBM T.J. Watson Research Center
1101 Kitchawan Rd, Yorktown Heights, NY 10598, USA
{ssohrab, riabov, udrea}@us.ibm.com

Abstract

Recent work on plan recognition as planning has shown great promise in the use of a domain theory and general planning algorithms for the plan recognition problem. In this paper, we propose to extend previous work to (1) address observations over fluents, (2) better address unreliable observations (i.e., noisy or missing observations), and (3) recognize plans in addition to goals. To this end, we introduce a relaxation of the plan-recognition-as-planning formulation that allows unreliable observations. That is, in addition to the original costs of the plan, we define two objectives that account for missing and noisy observations, and optimize for a linear combination of all objectives. We approximate the posterior probabilities of generated plans by taking into account the combined costs that include penalties for missing or noisy observations, and normalizing over a sample set of plans generated by finding either diverse or high-quality plans. Our experiments show that this approach improves goal recognition in most domains when observations are unreliable. In addition, we evaluate plan recognition performance and show that the high-quality plan generation approach should be preferred in most domains.

1 Introduction

Plan recognition is the problem of recognizing the plans and the goals of an agent given a set of observations. There exists a number of different approaches to the plan recognition problem including the use of SAT solvers and planning (e.g., [Ramírez and Geffner, 2009; Zhuo *et al.*, 2012]) where the domain theory is given as an input as well as the use of techniques that assume a plan library is given as an input [Sukthankar *et al.*, 2014]. Plan recognition continues to be an important problem to study as it has many practical applications such as assisted cognition [Pentney *et al.*, 2006], computer games [Synnaeve and Bessière, 2011], and network monitoring [Sohrabi *et al.*, 2013; Riabov *et al.*, 2015].

In this paper, we revisit the work on plan recognition as planning by Ramírez and Geffner 2009; 2010, where the domain theory is given as an input and planning algorithms are used to generate a solution to the plan recognition problem. The advantage of using domain theory and planning is that plan library is no longer required as input. In addition, AI planning techniques can be exploited to compute the solution. Hence, use of domain theory and planning algorithms for plan recognition is more flexible, general, and scalable.

While the use of planning for plan recognition by Ramírez and Geffner 2009-2010 showed great promise and allowed for some noisy behavior, it did not effectively deal with unreliable observations. Unreliable observations can be a result of a sensor malfunction, intentional obfuscation by malware, or simply a mistake that an agent makes [Thorsley *et al.*, 2008]. We consider two classes of unreliable observations: (1) observations that are noisy - cannot be explained by the actions of a plan for a particular a goal; hence, need to be discarded, and (2) observations that are missing - observations that should have been picked up but have not, possibly because the sensors malfunctioned. Unreliable observations present a challenge for the previous work due to the way the posterior probability of a goal G given observations O, $P(G|O)$, is calculated. Even if we have a reasonable prior estimate of $P(G)$, $P(G|O)$ as calculated by previous work, will be underestimated when observations are unreliable, because the cost difference of achieving G and O, and achieving G and not O will be large. This is because either the plan that satisfies G and complies with O has a much higher cost than the plan that satisfies G and does not comply with O (or complies to just part of the observations in O), or possibly no plan satisfying both O and G, either optimal or suboptimal, exist.

The main contribution of this paper is to provide a relaxation of the plan-recognition-as-planning formulation that allows for unreliable observations. That is, we provide a transformation of the original plan recognition problem to a new planning problem with action costs, where the costs now include penalties for missing or noisy observations. Hence, in addition to the original costs of the plan, we define two objectives that account for missing and noisy observations, and optimize for a linear combination of all objectives. We approximate the posterior probabilities of plans by taking into account the cost of the plan in the transformed planning problem, and normalizing over a sample set of plans generated by finding either diverse or high-quality plans. The posterior probabilities of goals are then computed by a summation over the posterior probabilities of the sampled set of plans that

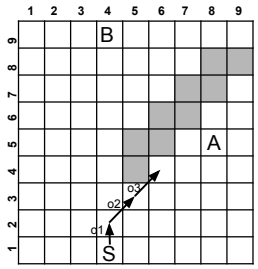

Figure 1: Goal Recognition Example

achieve this goal. In addition, unlike most plan recognition approaches, we consider observations over fluents and argue that in many applications the actual actions of an agent may not be observable, instead, their effects through the change in the states are observable.We use a diverse planner, LPG-d [Nguyen *et al.*, 2012], and a top-k planner, TK*, [Riabov *et al.*, 2014] to generate a set of plans. Our experiments show that our approach improves goal recognition in most domains when observations are unreliable. In addition, we evaluate plan recognition performance and show that the top-k plan generation approach performed better in most domains.

2 Motivating Example

Consider the grid example shown in Figure 1 where an agent initially at cell marked S heads to one of the goals A or B. The agent can move diagonally, horizontally or vertically all at a cost of 1. The arrows indicate the observations.

First, assume that the agent can move to any neighboring cells including the grey cells. Using the previous approach's algorithm [Ramírez and Geffner, 2010] where $P(G|O)$ is defined by $\frac{e^{-\beta\Delta}}{1+e^{-\beta\Delta}}$, $P(G{=}B|O){=}0.65$, because the cost difference or Δ is 0, while $P(G{=}A|O){=}0.35$. Note, the numbers are normalized. However, the algorithm that we will describe in this paper concludes that $P(G{=}A|O)$ is higher than $P(G{=}B|O)$ because pursing goal A will result in having less number of missing observations. Hence, knowing that the agent is pursuing goal A, our approach assigns a higher posterior probability to the true goal. The previous approach fails to assign a high probability to the true goal because it only penalizes for the agent not pursing the optimal plan and does not take into account the higher number of missing observations if the agent were to choose the optimal plan.

Now consider a modified version of the grid example where we know that the agent cannot go to the same cell it visited before, cannot move to the grey cells, and cannot go diagonally to left and up (i.e., north west). Then using the previous approach $P(G{=}B|O){=}0$ because there is no plan that would achieve both G and O, either optimally or sub-optimally, and $P(G{=}A|O){=}1$. However, it is possible that the third observation is noisy, and that the agent was really pursuing goal B. Hence, our algorithm does not prune goal B and instead it assigns probabilities such that $0 < P(G{=}B|O) < P(G{=}A|O)$. Therefore, in the case that the agent was pursing goal B, our approach assigns a higher posterior probability to the true goal compared to the previous approach.

3 Plan Recognition with Domain Theory

Before describing the plan recognition problem we highlight a few key differences between our approach and that of prior work: (1) unlike many plan recognition approaches we do not require plan libraries as input. Instead we require a domain theory; (2) we consider observations over fluents as we believe in many applications the actual actions of an agent may not be observable, instead, their effects through the change in the states are observable; (3) we consider unreliable observations (i.e., noisy or missing); (4) our approach can infer or recognize both goals and plans. In the rest of this section, we define the plan recognition problem and its solution. We will also define how to compute the posterior probabilities of both goals as well as plans given the unreliable observations.

Definition 1 (Planning Problem) *A planning problem is a tuple $P = (F, A, I, G)$, where F is a finite set of fluent symbols, A is a set of actions with preconditions,* $\text{PRE}(a)$*, add effects,* $\text{ADD}(a)$*, delete effects,* $\text{DEL}(a)$*, and non-negative action costs,* $\text{COST}(a)$*, $I \subseteq F$ defines the initial state, and $G \subseteq F$ defines the goal state.*

A state, s, is a set of fluents that are true. An action a is *executable* in a state s if $\text{PRE}(a) \subseteq s$. We define the successor state as $\delta(a, s) = ((s \backslash \text{DEL}(a)) \cup \text{ADD}(a))$ for the executable actions. The sequence of actions $\pi = [a_0, ..., a_n]$ is executable in s if state $s' = \delta(a_n, \delta(a_{n-1}, \ldots, \delta(a_0, s)))$ is defined. Moreover, π is the solution to the planning problem P if it is executable from the initial state and $G \subseteq \delta(a_n, \delta(a_{n-1}, \ldots, \delta(a_0, I)))$. Furthermore, π is said to be optimal if it has minimal cost, where $\text{COST}(\pi){=}\sum \text{COST}(a_i)$.

Definition 2 (Plan Recognition Problem) *A plan recognition problem is a tuple $R = (F, A, I, O, \mathcal{G}, \text{PROB})$, where (F, A, I) is the planning domain as defined above, $O = [o_1, ..., o_m]$, where $o_i \in F$, $i \in [1, m]$ is the sequence of observations, $\mathcal{G}$ is the set of possible goals G, $G \subseteq F$, and* PROB *is the goal priors or a probability distribution over $\mathcal{G}$.*

In general, O can be expressed as a Linear Temporal Logic (LTL) formula or Past LTL formula [Emerson, 1990; Sohrabi *et al.*, 2011]. In other words, each observation can be a logical expression over the set of fluents. While it is possible to address this general type of observations, our focus in this paper is on observations that are sequenced, or are totally ordered, and that each observation is an observable fluent. Observations over fluents are more general and flexible than observations over actions, because often in practice, actions are not directly observable, and instead some of their effects can be observed, for example, through sensors. However, we can also deal with observations over actions by assigning a unique fluent per action that is added only by that action. This is how we are able to directly compare with the prior work which focused on observations over actions.

Next, we define the satisfaction of an observation sequence O by an action sequence. This definition is a relaxation over the original "satisfaction" definition because it allows for observations to be left unexplained. This is necessary in order to address the noisy observations.

Definition 3 (Satisfaction) *Let $\sigma = s_0 s_1 s_2 ... s_{n+1}$ be an execution trace of an action sequence $\pi = [a_0, ..., a_n]$ from the initial state, where $\delta(a_i, s_i) = s_{i+1}$ is defined, for any $i \in [0, n]$. Given a planning domain (F, A, I), an observation sequence $O = [o_1, ..., o_m]$ is said to be satisfied by an action sequence $\pi = [a_0, ..., a_n]$, and its execution trace σ if there is a non-decreasing function f that maps the observation indices $j = 1, ..., m$ into the state indices $i = 0, ..., n+1$, such that for all $1 \leq j \leq m$, either $o_j \in s_{f(j)}$, or $o_j \notin s_{f(j)}$.*

The above definition takes into account the observation order through the mapping of the non-decreasing function. It also allows for leaving some observations unexplained. In one extreme, all observations can be explained by the sequence of states, and in the other extreme, all observations are discarded as it may be possible that all observations are noisy, but very unlikely. Hence, there may be many such non-decreasing functions and that the definition holds as long as at least one such mapping exists. Next, we formulate the definition of noisy observation and missing observation.

Definition 4 (Noisy/Missing) *let $T \subseteq F$ be the set of observable fluents, $P = (F, A, I, G)$ be a planning problem, O be an observation sequence, and π be a plan for G that satisfies O. Then O is said to be noisy with respect to the given G and π, if at least one of its observations, $o \in O$, is noisy. o is said to be noisy, if o is never added to the state by any of the actions $a_i \in \pi$ (i.e., $o \notin \text{ADD}(a_i)$). Further, O is said to have missing observations with respect to the given G and π if at least one of its observations is missing. o is said to be missing if (1) o is observable or $o \in T$; (2) $o \notin O$; and (3) o is added by at least one of the actions $a_i \in \pi$ (i.e., $o \in \text{ADD}(a_i)$).*

Hence, noisy observations are those that have not been added to the state for that particular goal and plan, while missing observations are those that have been added but were not observed. Going back to our example in Figure 1, o3 can be a noisy observation for goal B, but not for goal A. Also many observations were missing for goal B, but only two were missing for goal A. Next, we define posterior goal probabilities $P(\pi|O)$ and posterior plan probabilities $P(G|O)$.

$$P(\pi|O) = \beta P(O|\pi)P(\pi) = \beta P(O|\pi)P(\pi|G)P(G) \quad (1)$$

where β is a normalizing constant that depends on $P(O)$ only, and $P(G)$ is $\text{PROB}(G)$. In this expression, we assume that the agent is pursuing only one goal and $P(\pi|G)$ is 0 for the action sequences π that are not plans for G. Using Bayes rule, we also have:

$$P(G|O) = \beta P(O|G)P(G) \quad (2)$$

Assuming that the observations are independent of the goal G given π, and π ranges over all plans that achieve G and satisfy O, $P(O|G)$ can be written as:

$$P(O|G) = \sum_{\pi} P(O|\pi) \cdot P(\pi|G) \quad (3)$$

Note, unlike previous work we do not assume that $P(O|\pi)$ is 0 for plans that are inconsistent with the observation because observations can be noisy. However, $P(O|\pi)$ is small if O is noisy or has missing observations. Hence we define $P(G|O)$ as follows with π that now ranges over the set of sampled found plans, Π, that achieve G and satisfy O.

$$P(G|O) = \sum_{\pi \in \Pi} P(\pi|O) \quad (4)$$

There are two important questions yet to be discussed: (1) how to define $P(O|\pi)$ and $P(\pi|G)$, and (2) how to use planning to find the sampled set of plans.

To address the first question, we provide an approximation that takes into account not only the original cost of the actions but also the number of missing and noisy observations according to Definition 4. Hence, we define a weighted factor, $V_{O,G}(\pi)$, that combines all our three objectives as follows:

$$V_{O,G}(\pi) = \text{COST}(\pi) + b_1 \cdot M_{O,G}(\pi) + b_2 \cdot N_{O,G}(\pi) \quad (5)$$

where π is a plan that meets the goal G and satisfies O, $M_{O,G}(\pi)$ is the number of missing observations in O, $N_{O,G}(\pi)$ is the number of noisy observations in O, and b_1 and b_2 are the corresponding coefficients assigning weights to the different objectives. Hence, we approximate the value of $P(O|\pi) \cdot P(\pi|G)$ by taking into account the value of $V_{O,G}(\pi')$ for all other sampled plans π' in addition to the value of $V_{O,G}(\pi)$ for this plan π. We use a constant β' to offset large sums if necessary to get a better approximation.

$$P(O|\pi) \cdot P(\pi|G) \approx 1 - \frac{\beta' \, V_{O,G}(\pi)}{\sum_{\pi' \in \Pi} V_{O,G}(\pi')} \quad (6)$$

With $\beta' = 1$, if there are 4 sampled plans, each with the same V values, then equation 6 results in 0.75, and once normalized $P(\pi|O) = 0.25$. On the other hand, if a plan π_1 has a higher V value than another plan π_2, then $P(\pi_1|O) < P(\pi_2|O)$. Equation 6 allows us to compute $P(\pi|O)$ and $P(G|O)$ given a set of sampled plans. In the next section, we address the second question which is how to use planning to find the sampled set of plans and also how to calculate $V_{O,G}(\pi)$ from the cost of the plans in the transformed planning problem.

4 Transformation to Planning

In this section, we will provide a general transformation of the plan recognition problem into a planning problem with action costs. This transformation allows use of AI planning, in particular, the use of planners capable of finding a plan set.

To create the new planning problem, we augment the existing actions with a set of "discard" and "explain" actions for each observation o_i in the observation sequence O. These actions ensure that the observations are considered while in some cases they may need to be discarded. In particular, the noisy observations can be skipped using the "discard" actions. We also ensure that the order of the observations is preserved using a special predicates, l_{o_i}, that is set to true if the observation is either explained to discarded. We add o_0 as a dummy initial observation and set l_{o_0} to true initially. Further, we ensure that at least one of the goals $G \in \mathcal{G}$ is satisfied by the computed plans. We do so by creating an

action for each $G \in \mathcal{G}$ with a special add predicate we call "done". The goal state will be updated to include this "done" predicate. This ensures that we restrict our search space and consider only plans that meet at least one of the given goals when generating the set of sampled plans.

Definition 5 *For a plan recognition problem* $R = (F, A, I, O, \mathcal{G}, \text{PROB})$ *we create a new planning problem with action costs* $P' = (F', A', I', G')$ *such that:*

- $F' = F \cup \{done, l_{o_0}\} \cup \{l_{o_i} | o_i \in O\}$,
- $I' = I \cup \{l_{o_0}\}$,
- $G' = \{done, l_{o_m}\}$, *where* o_m *is the last observation,*
- $A' = A_{orig} \cup A_{goal} \cup A_{discard} \cup A_{explain}$, *where,*
  - $A_{orig} = \{h_a | \ a \in A, \text{COST}(h_a) = \text{COST}(a) + b_1 \times |\{o_i | o_i \in \text{ADD}(a) \wedge o_i \in T \wedge o_i \notin O\}|$, *where* $T \subseteq F$ *is the set of observable fluents*$\}$,
  - $A_{goal} = \{h_g | \ g \in \mathcal{G}, \text{COST}(h_g) = 0, \text{PRE}(h_g) = \{g\}, \text{ADD}(h_g) = \{done\}\}$,
  - $A_{discard} = \{h_{o_i} | o_i \in O, \text{COST}(h_{o_i}) = b_2, \text{PRE}(h_{o_i}) = \{\neg o_i, l_{o_{i-1}}\}, \text{ADD}(h_{o_i}) = \{l_{o_i}\}, \text{DEL}(h_{o_i}) = \{l_{o_{i-1}}\}\}$,
  - $A_{explain} = \{h_{o_i} | \ o_i \in O, \text{COST}(h_{o_i}) = 0, \text{PRE}(h_{o_i}) = \{o_i, l_{o_{i-1}}\}, \text{ADD}(h_{o_i}) = \{l_{o_i}\}, \text{DEL}(h_{o_i}) = \{l_{o_{i-1}}\}\}$.

Note, the cost of the plan for P' now encodes a penalty for the missing observations and unexplained observations. In fact, it is not hard to see that the cost of the plan for P' is the same as $V_{O,G}(\pi)$, according to Equation 5.

Theorem 1 (Soundness/Correctness) *Given a plan recognition problem* $R = (F, A, I, O, \mathcal{G}, \text{PROB})$*, and the corresponding planning problem* $P' = (F', A', I', G')$ *as defined above, for all* $G \in \mathcal{G}$*, if* π *is a plan for* R*, then there exists a plan* π' *for* P' *such that* π *can be constructed straightforwardly from* π' *by removing the extra actions (i.e., discard, explain, and goal actions) and more importantly,* $V_{O,G}(\pi) = \text{COST}(\pi')$*. On the other hand, if there is a plan* π' *for* P'*, then there exists a plan* π *for* P *that can be constructed from* π' *by removing the extra actions such that* $V_{O,G}(\pi) = \text{COST}(\pi')$*.*

Proof is based on the fact that the extra actions do not change any of the observable fluents while preserving the ordering amongst the observations. Moreover, cost of the plans now map to $V_{O,G}(\pi)$, hence, the posterior probabilities, $P(G|O)$ and $P(\pi|O)$, can be computed using the cost of the plans in the transformed planning problem. Note that these probabilities will be different based on which method is used to generate a sample set of plans.

5 Computation

We present two methods in finding a sampled set of plans. The first method is based on top-k planning, and the second method is based on finding a set of diverse plans.

5.1 Computation via Top-k Planning

We define top-k planning similar to its definition given in [Riabov *et al.*, 2014]. The top-k planning problem is a tuple (P, k), where P is the planning problem with action costs as defined in Definition 1, and k is the number of plans to find. Let n be the number of valid plans for the planning problem P. The set of plans $\Pi = \{\pi_1, ..., \pi_m\}$, where $m = k$ if $k \leq n$, $m = n$ otherwise, is the solution to the top-k planning problem (P, k) if and only if each $\pi_i \in \Pi$ is a plan for the planning problem P; and there does not exists a plan π' for P, $\pi' \notin \Pi$, and a plan $\pi_i \in \Pi$ such that $\text{COST}(\pi') < \text{COST}(\pi_i)$. Hence, Π may contain just one optimal plan (i.e., $k = 1$), all optimal plans, or all optimal plans and some suboptimal plans if k is large enough.

There are several techniques that can be used to compute the top-k plans as described by [Riabov *et al.*, 2014]. In this paper, we use the top-k planning planner, TK*, that is based on the use of a k shortest paths algorithm called K^* [Aljazzar and Leue, 2011] because it has been shown that TK* outperforms other techniques for top-k planning. K shortest paths problem is an extension of the shortest path problem where in addition of finding one shortest path, a set of paths is found, representing the k shortest paths [Hoffman and Pavley, 1959]. The K^* algorithm is an improved variant of the Eppstein's k shortest paths algorithm [Eppstein, 1998] because it does not require the complete graph of states and actions to be available in memory. TK*, applies K^* to search in state space, with dynamic grounding of actions. For more details on the TK* planner, please see [Riabov *et al.*, 2014].

Note, while the the sample set of plans we are generating is not required to be of high quality or low cost, use of the top-k planning approach is guaranteed to find such a set. In turn, the solution set will optimize for the value of $V_{O,G}(\pi)$.

5.2 Computation via Diverse Planning

In diverse planning the objective is find a set of plans m that are at least d distance away from each other. The distance between plans can be computed by considering the plans as a set of actions, a sequence of states, or casual links. The solution to the diverse planning problem, (m, d), is a set of plans Π, such that $|\Pi| = m$ and $min_{\pi,\pi' \in \Pi} \ \delta(\pi, \pi') \geq d$, where $\delta(\pi, \pi')$ measures the distance between plans.

There are several techniques that can be used to compute the diverse plans and there are several diverse planners that exist [Srivastava *et al.*, 2007; Coman and Muñoz-Avila, 2011; Nguyen *et al.*, 2012; Bryce, 2014]. In this paper, we use LPG-d [Nguyen *et al.*, 2012], which is an extension of a local search-based planner, LPG [Gerevini *et al.*, 2003], for two reasons: (1) we were able to access and run it successfully, and (2) it showed relatively better performance compared to the other diverse planners [Roberts *et al.*, 2014]. We use three settings of LPG-d, (10, 0.75), (50, 0.5), and (100, 0.75), and report on our experiments in the next section.

6 Experimental Evaluation

In this section, we evaluate our two proposed plan recognition approaches, LPG-d [Nguyen *et al.*, 2012] for diverse planning, and TK* [Riabov *et al.*, 2014] for top-k planning, against previous work [Ramírez and Geffner, 2010]. We configured the previous work so that it uses the LM-Cut[1] [Pom-

[1]We used the most recent download of the fast downward planning system (http://www.fast-downward.org/) with the parameter set to "–alias seq-opt-lmcut' to obtain LM-Cut.

			Ramírez & Geffner			Proposed Approaches					
			LM-Cut			TK*, top-1000			LPG-d, (50, 0.5)		
	%O	|E|	S	P	Q	S	P	Q	S	P	Q
Intrusion $\|\mathcal{G}\|$=20	25	0	1.2/1.1	**0.27**	40/2	0.7/0	0.21	21/0	1/3.2	0.17	**26/26**
	50	0	1/0.5	**0.37**	40/0	0.5/0	0.25	25/0	1/4.3	0.25	**45/30**
	75	0	1/0.1	**0.39**	40/0	0.2/0	0.15	15/0	1/4.2	0.30	**62/23**
	100	0	1/0	**0.39**	40/0	0.0/0	0.02	2/0	1/3.8	0.31	**58/23**
	25	2	1.9/1.2	0.18	23/15	0.6/0	0.17	17/0	1/4.0	0.15	**26/32**
	50	2	2.8/0.4	**0.27**	30/0	0.3/0	0.13	13/0	1/4.5	0.20	**32/45**
	75	2	5.3/0.2	**0.29**	30/0	0.1/0	0.02	2/0	1/3.9	0.26	**45/28**
	100	2	8.2/0	0.24	25/0	0.1/0	0.02	2/0	1/3.5	**0.29**	**51/23**
Campus $\|\mathcal{G}\|$=2	25	0	1.2/0.8	**0.72**	**95/5**	1/0.1	0.65	65/7	1/0.8	0.46	42/44
	50	0	1.1/0.5	0.81	95/5	1/0.3	**0.87**	91/7	1/0.8	0.48	53/35
	75	0	1/0.5	0.88	100/0	1/0.1	**0.98**	100/0	1/0.8	0.53	53/37
	100	0	1/0.5	0.91	100/0	1/0.1	**0.99**	100/0	1/0.8	0.56	56/40
	25	2	1.3/0.6	**0.62**	**81/19**	1/0.1	0.59	58/7	1/0.8	0.50	56/30
	50	2	1.2/0.4	0.75	**91/9**	1/0.3	0.75	74/16	1/0.8	0.50	56/33
	75	2	1.2/0.4	0.81	**100/0**	1/0.2	**0.92**	95/2	1/0.8	0.54	58/40
	100	2	1.2/0.3	0.79	98/2	1/0.3	**0.89**	93/7	1/0.9	0.52	53/40
Kitchen $\|\mathcal{G}\|$=3	25	0	1.1/0.4	0.67	57/43	1/0	**0.99**	**100/0**	1/0.7	0.79	86/14
	50	0	1/0.4	0.78	**71/29**	1/0.1	**0.84**	86/0	1/0	0.71	71/0
	75	0	1/0.3	0.86	86/14	1/0.1	**0.91**	86/14	1/0.4	0.69	71/14
	100	0	1/0.1	**0.88**	**86/14**	0.9/0.1	0.77	71/14	1/0	0.85	86/0
	25	2	2.4/0	0.14	14/0	0.4/0	0.14	14/0	1/0.3	**0.59**	**57/14**
	50	2	2.4/0	0.00	0/0	0.6/0	**0.57**	**57/0**	1/0.1	0.48	43/14
	75	2	3.0/0	0.00	0/0	0.3/0	0.29	29/0	1/0	**0.71**	**71/0**
	100	2	2.7/0	0.14	14/0	0.6/0	0.43	43/0	1/0.1	**0.83**	**86/0**
IPC-Grid $\|\mathcal{G}\|$=5	25	0	1.2/0.4	0.82	100/0	1/0.4	**0.85**	90/10	1/1.6	0.51	67/29
	50	0	1.4/0.1	0.82	86/0	1/0.1	**0.93**	95/0	1/2.1	0.38	**48/48**
	75	0	1.6/0.1	0.85	86/0	1/0	**1.00**	**100/0**	1/2.2	0.33	38/52
	100	0	1.8/0	0.80	81/0	1/0	**1.00**	**100/0**	1/2.2	0.36	38/57
	25	2	4.5/0.1	0.05	5/5	1/1.2	**0.32**	33/24	1/2.2	0.25	**19/67**
	50	2	4.8/0	0.00	0/0	1/0.8	**0.38**	43/14	1/2.6	0.31	**43/43**
	75	2	5.0/0	0.00	0/0	1/0.6	0.18	14/19	1/2.2	**0.38**	**48/33**
	100	2	5.0/0	0.00	0/0	1/1.0	**0.34**	38/19	1/2.2	0.30	**33/52**

Table 1: Comparison of our two proposed plan recognition approaches for recognizing a goal, top-k planning using TK*, and diverse planning using LPG-d against previous work by Ramírez and Geffner 2010, using LM-Cut. The columns stand for % of observations selected, number of extra observations, |E|, average number of most/less likely goals, S, posterior probability of the goal given the observations, P, and average percentage of problems where the ground truth goal is among the most/less likely goals, Q. The overall value of Q indicates the coverage for that method (i.e., sum of the most and less likely goals). The bold numbers indicate better results.

merening and Helmert, 2012] planner. We selected the LM-Cut planner because it was shown to perform well in the planning competition and it performs better than the original planners used for the configuration of the previous approach. Note, using a more efficient optimal planner or even choosing a sub-optimal planner will not improve the presented results for the previous approach because LM-Cut was able to solve all problems optimally within the time limit. We used a timeout of 30 minutes and ran all our experiments on a dual 16-core 2.70 GHz Intel(R) Xeon(R) E5-2680 processor with 256 GB RAM. Note, while we obtained the results by running the top-k planner and the diverse planner once for each problem, LM-Cut was run $2 \times |\mathcal{G}|$ times.

6.1 Planning Problems

In our experimental evaluation we used four domains and had over 1000 problems. We used the Kitchen, Intrusion, Campus, and IPC-Grid domain from the previous approach [Ramírez and Geffner, 2010] but modified the kitchen domain to disallow the extra "take" actions that are not towards achieving the goal in order to evaluate the approaches in the case of noisy observations that need to be discarded. We did not run the other domains reported by the previous approach such as the blocks words because of the use of equality which is not supported by the top-k planner. To construct the planning problems, for each domain we use a fixed given possible goal set $\mathcal{G}$ and for each goal set $\mathcal{G}$ we generate a set of optimal and suboptimal plans. From these plans, we select the observable actions in order to construct O, keeping track of the goal used in the planning problem (i.e., ground truth goal). To evaluate how well the approaches address missing observations, we create several problems that do not have the full observation sequence (i.e., some observation are missing). We do so by randomly selecting 25%, 50%, and 75% of the observations in O. Therefore, the 100% case indicates that the full observation sequence, O, is given. Furthermore, to evaluate how these approaches address noise, we randomly add extra observations to the original observation sequence. While some domains such as the IPC-Grid domain can still explain these extra observations using the extra "move" actions, a plan for G that fully explains O may not exist in other domains. In the special case that no plan is found for G and O, we assign a high cost to that plan which can lead to $P(G|O) \simeq 0$ for that goal for the previous approach.

For the LPG-d planner we used three settings of (m, d), (10, 0.75), (50, 0.5), and (100, 0.75), but report only on the (50, 0.5) case; 50 plans that are at least 0.5 distance away from each other. This is because this setting performed better than the other two both with respect to recognizing a goal and recognizing a plan. For the top-k planner, we set $k = 1000$ in order to obtain a large set of high-quality plans that are also diverse. Lower values of k might help improve the posterior probabilities but will likely result in lower converge. For the coefficients, we set $b_1 = 2$, and $b_2 = 4$ in the experiments meaning that we assign a higher penalty for the unexplained observations, and a lower penalty for the missing observations. We also experimented with different values for these coefficients but the results were not significantly different.

6.2 Goal Recognition

Table 1 shows the summary of the results when evaluating our two proposed plan recognition approaches for recognizing a goal with the previous work [Ramírez and Geffner, 2010]. The number of extra observations, |E|, indicates the amount of noise introduced to the problem; we added two extra observations resulting in about 12% noise on average relative to the total number of observations that was close to 16 on average. To evaluate the coverage and accuracy of the different approaches, in addition to the actual posterior probability of the goal given observations, $P(G|O)$, shown in the P

column, we measure the average number of the most likely goals and the average number of less likely goals found by that method, shown in the S column separated by "/" and average percentage of problems where the ground truth goal is among the most and less likely goals also separated by "/" shown in the Q column. The overall value of Q, sum of the most and less likely, indicates the goal recognition coverage for that method. The most likely goals are chosen relative to that particular approach (i.e., goals with the highest posterior probability) and the less likely goals are those goals with greater than 0.05 posterior probability. Hence, value of "1/3.2" in the S column indicates that, on average, one of the goals in $\mathcal{G}$ was found to be most likely, while 3.2 of the goals were found to be less likely. Also value of "56/40" in the Q column indicates that, on average, the ground truth goal was among the goals found to be most likely 56% of the time, while the ground truth goal was among goals found to be be less likely 40% of the time, and that that approach had on average 96% goal recognition coverage for this problem.

The results in Table 1 show that our approach, diverse planning and top-k planning, helped improve the percentage of finding the ground truth goal in many domains or the coverage, when observations are unreliable. In particular, in Kitchen and IPC-grid domain, it helped improve the percentage of finding the ground truth goal even with high probability, when the observations are unreliable. Note that there are a few cases such as in the Intrusion domain that TK* performed worse. This is mainly because the TK* ran out of time and was not able to find the top-k plans. However, we expect to obtain better performance in domains that have smaller search space, and/or use a more efficient top-k planner.

Comparing the results of the problems where %O=25 versus %O=100 (i.e., no missing observations), LGP-d and TK* performed better in recognizing the goal; LGP-d's coverage increased from the average of 79% to 87% and TK*'s coverage increased from the average of 56% to 61%. However, goal recognition coverage went down from the average of 63% to 57% for the previous approach in our problem sets. This can be caused by the general poor performance of the previous approach on long observation sequences since the observations may not all be from an optimal plan.

On average, over all our problems, LPG-d had 83% coverage, TK* had 60% coverage, and LM-Cut had 59% coverage. Moreover, when observations are unreliable, problems where $|\mathrm{E}| = 2$ or %O is not 100, the coverage stayed the same for LPG-d and TK* but it went down to 55% for LM-Cut. However, when observations are noisy, problems where $|\mathrm{E}| = 2$, the coverage went down to 79% for LPG-d, to 46% for TK*, and to 35% for LM-Cut. This indicates that the previous approach performed worse when observations are unreliable, even more so when observations are noisy. Also note that the posterior probabilities of goals, when observations are noisy, is on average 0.26 for LM-Cut, versus on average 0.42 for LPG-d. Also, the size of the most likely goals is largest for LM-Cut and lowest for the top-k approach, indicating that the top-k approach has a higher precision in detecting the ground truth goal than the previous approach.

6.3 Plan Recognition

We evaluate our two proposed plan recognition methods for recognizing a plan, top-k planning and diverse planning, and we did not compare with the previous work [Ramírez and Geffner, 2010] since their focus was on goal recognition. Also, in our benchmark problems, recognizing the full plan is often very challenging. This is because of the combinatorial property of these domains (i.e., there is often no particular order among the actions). Hence, instead of recognizing the full plan, we find a random subset of actions in the full plan and we evaluate our techniques in recognizing this partial plan.

The results we obtained show that while LPG-d performed better in some domains, TK* performed better on average over all domains with respect to finding the ground truth partial plan with high probability; TK* found the partial plan with high probability in 43% of the time, while LPG-d found the partial plan with high probability in 25% of the time. This can be because the top-k planning approach focuses on finding a set of high-quality plans, while LPG-d finds diverse plans instead and this is helpful in some domains but not all. However, both approaches were able to find the ground truth partial plan in about only 50% of the time on average as this problem was much more difficult for the two approaches. Also the posterior probability of partial plans for the top-k planning approach is on average 0.48 compared to 0.41 for the LPG-d approach. In summary, while the previous work as it stands is not able to recognize plans, our experiments show that our approach is able to recognize ground truth (partial) plans. Moreover, top-k planning approach performed better in plan recognition coverage and found a higher posterior probability for the partial plan.

7 Discussion and Summary

There exist a body of work on the plan recognition problem with several different approaches including the use of AI planning and other techniques such as the use of SAT [Zhuo *et al.*, 2012]. While most existing work assumes that the observations are reliable, a recent work by Vattam et al. 2015 addresses missing and noisy observations. However, they require plan libraries and use case-based plan recognition algorithms, whereas, we use AI planning instead. Generating a plan set rather than just one plan has gained its popularity in recent years in the context of diverse planning. There exist several diverse planners including LPG-d which we use in this paper. Using a more efficient diverse planner, specially if it accounts for cost, such as a recent approach by Vadlamudi et al. 2016, can improve the performance of our approach.

In this paper, we proposed to extend previous work to address observations over fluents, better address unreliable observations, and recognize plans in addition to goals. To this end, we provide a relaxation of the plan-recognition-as-planning formulation that allows unreliable observations. We also defined the posterior probabilities of goals and plans. We evaluate our solution using two approaches, diverse planning, and top-k planning. Our experiments showed that using our approach improves goal recognition in most domains when observations are unreliable. We also reported on the extension of our approach for recognizing plans. In the future, we

would like to tackle the plan recognition problem where actions can have durations. This is a difficult problem especially in the presence of concurrency.

References

[Aljazzar and Leue, 2011] Husain Aljazzar and Stefan Leue. K*: A heuristic search algorithm for finding the k shortest paths. *Artificial Intelligence*, 175(18):2129–2154, 2011.

[Bryce, 2014] Daniel Bryce. Landmark-based plan distance measures for diverse planning. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 56–64, 2014.

[Coman and Muñoz-Avila, 2011] Alexandra Coman and Hector Muñoz-Avila. Generating diverse plans using quantitative and qualitative plan distance metrics. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, pages 946–951, 2011.

[Emerson, 1990] E. Allen Emerson. Temporal and modal logic. *Handbook of theoretical computer science: formal models and semantics*, B:995–1072, 1990.

[Eppstein, 1998] David Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 28(2):652–673, 1998.

[Gerevini *et al.*, 2003] Alfonso Gerevini, Alessandro Saetti, and Ivan Serina. Planning through stochastic local search and temporal action graphs in LPG. *Journal of Artificial Intelligence Research*, 20:239–290, 2003.

[Hoffman and Pavley, 1959] Walter Hoffman and Richard Pavley. A method for the solution of the *n*th best path problem. *Journal of the ACM*, 6(4):506–514, 1959.

[Nguyen *et al.*, 2012] Tuan Anh Nguyen, Minh Binh Do, Alfonso Gerevini, Ivan Serina, Biplav Srivastava, and Subbarao Kambhampati. Generating diverse plans to handle unknown and partially known user preferences. *Artificial Intelligence*, 190:1–31, 2012.

[Pentney *et al.*, 2006] William Pentney, Ana-Maria Popescu, Shiaokai Wang, Henry A. Kautz, and Matthai Philipose. Sensor-based understanding of daily life via large-scale use of common sense. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pages 906–912, 2006.

[Pommerening and Helmert, 2012] Florian Pommerening and Malte Helmert. Optimal planning for delete-free tasks with incremental lm-cut. In *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS)*, 2012.

[Ramírez and Geffner, 2009] Miquel Ramírez and Hector Geffner. Plan recognition as planning. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1778–1783, 2009.

[Ramírez and Geffner, 2010] Miquel Ramírez and Hector Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI)*, 2010.

[Riabov *et al.*, 2014] Anton Riabov, Shirin Sohrabi, and Octavian Udrea. New algorithms for the top-k planning problem. In *Proceedings of the Scheduling and Planning Applications woRKshop (SPARK) at the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 10–16, 2014.

[Riabov *et al.*, 2015] Anton V. Riabov, Shirin Sohrabi, Daby M. Sow, Deepak S. Turaga, Octavian Udrea, and Long H. Vu. Planning-based reasoning for automated large-scale data analysis. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 282–290, 2015.

[Roberts *et al.*, 2014] Mark Roberts, Adele E. Howe, and Indrajit Ray. Evaluating diversity in classical planning. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 253–261, 2014.

[Sohrabi *et al.*, 2011] Shirin Sohrabi, Jorge A. Baier, and Sheila A. McIlraith. Preferred explanations: Theory and generation via planning. In *Proceedings of the 25th National Conference on Artificial Intelligence (AAAI)*, pages 261–267, 2011.

[Sohrabi *et al.*, 2013] Shirin Sohrabi, Octavian Udrea, and Anton Riabov. Hypothesis exploration for malware detection using planning. In *Proceedings of the 27th National Conference on Artificial Intelligence (AAAI)*, pages 883–889, 2013.

[Srivastava *et al.*, 2007] Biplav Srivastava, Tuan Anh Nguyen, Alfonso Gerevini, Subbarao Kambhampati, Minh Binh Do, and Ivan Serina. Domain independent approaches for finding diverse plans. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2016–2022, 2007.

[Sukthankar *et al.*, 2014] Gita Sukthankar, Christopher Geib, Hung Hai Bui, David V. Pynadath, and Robert P. Goldman. *Plan, Activity, and Intent Recognition*. Morgan Kaufmann, 2014.

[Synnaeve and Bessière, 2011] Gabriel Synnaeve and Pierre Bessière. A bayesian model for plan recognition in RTS games applied to starcraft. In *Proceedings of the 7th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2011*, 2011.

[Thorsley *et al.*, 2008] David Thorsley, Tae-Sic Yoo, and Humberto E. Garcia. Diagnosability of stochastic discrete-event systems under unreliable observations. In *Proceedings of American Control Conference*, pages 1158– 1165, 2008.

[Vadlamudi and Kambhampati, 2016] Satya Gautam Vadlamudi and Subbarao Kambhampati. A combinatorial search perspective on diverse solution generation. In *Proceedings of the 30th National Conference on Artificial Intelligence (AAAI)*, 2016. To appear.

[Vattam and Aha, 2015] Swaroop S Vattam and David W Aha. Case-based plan recognition under imperfect observability. In *Proceedings of the 23rd International Conference on Case Based Reasoning (ICCBR)*, pages 381–395, 2015.

[Zhuo *et al.*, 2012] Hankz Hankui Zhuo, Qiang Yang, and Subbarao Kambhampati. Action-model based multi-agent plan recognition. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 377–385, 2012.