The AAAI-17 Workshop on
Plan, Activity, and Intent Recognition
WS-17-13

An AI Planning-Based Approach to the Multi-Agent Plan Recognition Problem (Preliminary Report)

Maayan Shvo *
Utrecht University
Utrecht, Netherlands

Shirin Sohrabi
IBM T.J. Watson Research Center
Yorktown Heights, USA

Sheila A. McIlraith
University of Toronto
Toronto, Canada

Abstract

Plan Recognition is the problem of inferring the goals and plans of an agent given a set of observations. In Multi-Agent Plan Recognition (MAPR) the task is extended to inferring the goals and plans of multiple agents. Previous MAPR approaches have largely focused on recognizing team structures and behaviors, given perfect and complete observations of the actions of individual agents. However, in many real-world applications of MAPR, observations are unreliable or missing; they are often over properties of the world rather than actions; and the observations that are made may not be explainable by the agents' goals and plans. Moreover, the actions of the agents could be durative or concurrent. In this paper, we address the problem of MAPR with temporal actions and with observations that can be unreliable, missing or unexplainable. To this end, we propose a multi-step compilation technique that enables the use of AI planning for the computation of the posterior probabilities of the possible goals. In addition, we propose a set of novel benchmarks that enable a standard evaluation of solutions that address the MAPR problem with temporal actions and such observations. We present results of an experimental evaluation on this set of benchmarks, using several temporal and diverse planners.

1 Introduction

Plan recognition – the ability to recognize the plans and goals of agents from observations – is useful in a myriad of applications including intelligent user interfaces, conversational agents, intrusion detection, video surveillance, and now increasingly in support of human-robot and robot-robot interactions (e.g., (Carberry 2001)). Originally conceived in the context of single agent plan recognition (e.g., (Cohen, Perrault, and Allen 1981), (Schmidt, Sridharan, and Goodson 1978), (Kautz and Allen 1986), (Charniak and Goldman 1993)), recent work has turned to the more complex task of Multi-Agent Plan Recognition (MAPR). In MAPR, the goals and/or plans of multiple agents are hypothesized, based upon observations of the agents, providing a richer paradigm for addressing many of the applications noted above. Early work in this area (e.g., (Banerjee, Lyle, and Kraemer 2010)) limited observations to activity-sequences, and focused the recognition task on the identification of dynamic team structures and team behaviors, relative to a predefined plan library.

While this formulation is effective for certain classes of problems, it does not capture important nuances that are evident in many real-world MAPR tasks. To this end, in this paper, we provide an enriched characterization of MAPR that provides support for a richer representation of the capabilities of agents and the nature of observations. In particular we support (1) differing skills and capabilities of individual agents; (2) agent skills and actions that are durative or temporal in nature (e.g., washing dishes or other durative processes (cf. (Fox and Long 2003))); (3) observations with respect to the state of the system; such observations range over fluents rather than over actions as actions may not be directly observable but rather inferred via the changes they manifest; (4) observations that are missing, unreliable, or that cannot be accounted for by agents' goals and plans.

Our approach to addressing this problem is to conceive the computational core of MAPR as a planning task, following in the spirit of the single-agent characterization of *plan recognition as planning* proposed by Ramirez and Geffner 2009. This contrasts with much of the previous work on MAPR which requires explicit plan libraries; and while the work done by Zhuo et al. 2012 replaces explicit plan libraries with sets of action models, it does not make use of AI planning. In our work, the conception of *MAPR as planning* enables the leveraging of recent advances in multi-agent planning as exemplified by the planners that participated in the 2015 Competition of Distributed and Multiagent Planners (CoDMAP)[1] (e.g., (Crosby, Jonsson, and Rovatsos 2014; Muise et al. 2015)), as well as advances in temporal planning (e.g., (Benton, Coles, and Coles 2012), and in the generation of diverse plans (e.g., (Riabov, Sohrabi, and Udrea 2014)).

To realize MAPR as planning, we propose a two-step compilation process that takes an MAPR problem as input. We first compile away the multi-agent aspect of the problem and then we compile away the observations. The resulting planning problem is temporal, has durative actions and temporal constraints; hence, temporal or makespan-sensitive planners can be applied to generate a plan that is then post-

*The work was performed during an internship at IBM

[1] http://agents.fel.cvut.cz/codmap/

processed to yield a solution to the original MAPR problem. We propose three different approaches to generating high-quality MAPR results, evaluating them experimentally.

The main contributions of this paper are: (1) a formalization of the MAPR problem with potentially unreliable observations over fluents, and actions that are temporal or durative in nature; (2) characterization of MAPR as planning via a two-step compilation technique that enables the use of temporal AI planning to generate hypothesized plans and goals. (3) three approaches to computing the posterior probability of a goal given the observations, providing a measure of the quality of solutions; (4) a set of novel benchmarks that will allow for a standard evaluation of solutions to the MAPR problem; (5) experimental evaluation of our proposed techniques on this set of benchmarks using several temporal and diverse planners, with our three proposed approaches.

2 Problem Definition

In this section, we review basic definitions necessary to our work, including definitions of a planning problem, a planning problem with temporal actions, a multi-agent system, and a plan recognition problem. Then, we introduce the multi-agent plan recognition problem with temporal actions and its solution. We begin with a motivating example.

Motivating Example: *Let us consider the following motivating example, taken from the International Planning Competition (IPC) Depots domain. In this domain, there are two different types of agents, hoist operators and truck drivers. Each agent may have their own goal(s), yet a common goal might be shared by the agents and distributed amongst them; in some cases, it is not possible to solve the planning problem of each agent separately since resources are shared between agents. For example, a truck driver must wait for a hoist operator to load a crate onto the truck, before being able to drive it to its designated location. The same goes for the hoist operators, that must wait for a truck to arrive at their depot, before beginning to load crates onto them and also truck drivers have to wait for the hoist operator to become available at a certain depot before loading them.*

Now, the input given to us in a plan recognition problem, specifies the different agent types (drivers and hoist operators), their different skill sets, temporal information about the durations of their actions, and a set of ordered observations that occurred in a specific timeframe. In our example, illustrated in Figure 1, we are given a snapshot of the initial state of the world at 8 AM and know the observations took place in a 45-minute time-frame, shown in the white boxes/areas. Based on the information given to us, we would like to infer which agent performed which action, and when, and what are the most likely goals. The system hypothesizes about the different agents' actions, based on the given observations. For example, the timeline hypothesizes that hoist 2 loaded the orange crate onto the blue truck, and that the execution of that action ended at 08:07, which can successfully explain the observation that tells us that the orange crate was in the blue truck at that time. Furthermore, we can learn from this timeline that the blue truck had to wait until 08:07, which is until the orange crate was loaded onto it, in order to be able to start driving from depot 2 to depot 1. The dashed line represents an alternative possible sequence of actions that might explain the observations, which involves the red truck traveling directly to depot 2, instead of stopping by depot 3. This alternative timeline also explains the observations and we can thus present the user with a set of the most likely plans that explain the observations. As can be seen by this example, our inference offers meaningful insights and can allow the system to hypothesize about which agent performed which action, and at what time.

Definition 1 (Planning Problem) *A planning problem is a tuple $P^c = (F, A, I, G)$, where F is a finite set of fluent symbols, A is a set of actions with preconditions, pre(a), add effects, add(a), delete effects, del(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 pre(a) $\subseteq s$. The successor state is defined as $\delta(a, s)$ =(($s\setminus$ del(a)) $\cup$ add(a)) for the executable actions. The sequence of actions $\pi = [a_1, ..., a_n]$ is executable in s if the state $s' = \delta(a_n, \delta(a_{n-1}, \ldots, \delta(a_1, s)))$ is defined. Moreover, π is the solution to the planning problem P^c if it is executable from the initial state and $G \subseteq \delta(a_n, \delta(a_{n-1}, \ldots, \delta(a_1, I)))$.

Next, we modify the above definition, to include temporal actions as defined in (Fox and Long 2003).

Definition 2 (Planning Problem with Temporal Actions) *A planning problem with temporal actions is a tuple $P^t = (F, A, I, G)$, where F, I, and G are defined as above, A is a set of temporal actions with duration, d(a), precondition at start, pre$_s$(a), precondition over all, pre$_o$(a), precondition at end, pre$_e$(a), add effects at start, add$_s$(a), add effects at end, add$_e$(a), delete effects at start, del$_s$(a), and delete effects at end, del$_e$(a)*

The semantics of a temporal action is often given using two non-temporal actions "start" and "end", here we provide a similar semantics that instead uses "start" and "end" states. A temporal action a is *executable* in a state s$_{start}$, ending in state s$_{end}$ if pre$_s$(a) $\subseteq$ s$_{start}$ and pre$_e$(a) $\subseteq$ s$_{end}$. The resulting states s$_{start'}$ and s$_{end'}$ are defined as s$_{start'}$ = ((s$_{start}$ $\setminus$ del$_s$(a)) $\cup$ add$_s$(a)) and s$_{end'}$ = ((s$_{end}$ $\setminus$ del$_e$(a)) $\cup$ add$_e$(a)). Note s$_{end}$ comes after s$_{start'}$. Additionally, the overall precondition, pre$_o$(a) must hold in every state between s$_{start'}$ and s$_{end'}$. The solution to P^t, is a set of action-time pairs, allowing actions to occur concurrently, where each action is executable, and the goal G holds in the final state. Back to our motivating example, the actions drive, load, and unload, each have duration and are temporal. Also as you see in Figure 1, it is possible that two actions occur concurrently.

Next, we briefly overview the multi-agent system as defined in (Brafman and Domshlak 2008). Note, the actions are not temporal and have no durations, but the different agents are able to perform different tasks.

Definition 3 (Multi-Agent System) *The multi-agent system, also known as MA-STRIPS, is described as P^m=*

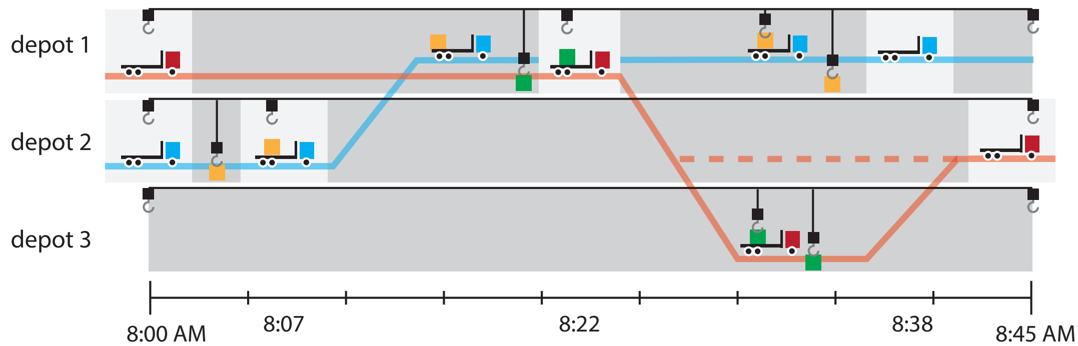

Figure 1: A timeline illustrating a 45-minute timeframe, set in the Depots domain, with 5 agents (2 truck drivers and 3 hoist operators). The y-axis is the location, spanning across the 3 different depot locations; the x-axis is the time line. The white areas indicate the observations. The lines represent alternative possible sequences of actions that might explain the observations.

$(F, N, \{A_i\}_{i=1}^n, I, G, \gamma)$*, where* F*,* I*, and* G *are defined as before,* $N = \{1, ..., n\}$ *is the number of agents,* A_i *is a set of classical actions for agent* i *as defined in Definition 1, and* $\gamma : A_1 \times ... \times A_n \to \{0, 1\}$ *specifies if a particular joint action meets the concurrency constraints.*

A simple joint action $c = (a_1, ..., a_n)$ is a member of the set $\mathcal{A} = A_1 \times ... \times A_n$, where pre($c$) = $\cup_i$pre(a_i), add(c) = $\cup_i$add(a_i), del(c) = $\cup_i$del(a_i). A joint action is executable in state s, if pre(c) $\subseteq s$, $\gamma(c) = 1$ (i.e., this joint action does not violate concurrency constraints), and del(c) $\cap$ add(c) = $\emptyset$ (i.e., the effects are not ill-defined). The sequence of actions $\pi = [a_1, ..., a_n]$ is the solution to the multi-agent system P^m if it is executable from the initial state I, and G holds in the final state. Note, no-op actions with empty precondition and effects can be used to align agent's actions.

Next, we define the plan recognition problem, following its definition in (Sohrabi, Riabov, and Udrea 2016; Ramírez and Geffner 2010).

Definition 4 (Plan Recognition Problem) *A plan recognition problem is a tuple* $P^r = (F, A, I, O, \mathcal{G}, \textsc{prob})$*, 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 probability of a goal,* $P(G)$*, or the goal priors.*

Unexplainable (aka noisy) observations are defined as those that have not been added by the effect of any actions of a plan for a particular goal, while missing observations are those that have been added but were not observed (i.e., are not part of the observation sequence). To address the unexplainable observations, Sohrabi et al. 2016 modifies the definition of satisfaction of an observation sequence by an action sequence introduced in (Ramírez and Geffner 2010) to allow observations to be left unexplained. Given an execution trace and an action sequence, an observation sequence is said to be satisfied if there is a non-decreasing function that maps the observation indices into the state indices as either explained or discarded. Hence, observations are all considered, while some can be left unexplained.

The solution to the plan recognition problem, P^r is the posterior probabilities of plans given observations, $P(\pi|O)$, and the posterior probabilities of goals given observations, $P(G|O)$. In previous work, AI planning is used to approximate these probabilities. In the work by Sohrabi et al. 2016, $P(\pi|O)$ is approximated by considering three objectives over a set of sample plans: (1) the cost of the original actions, (2) the number of missing observations, and (3) the number of unexplainable observations. Posterior probabilities of goals given observations, $P(G|O)$, is then computed by a summation over $P(\pi|O)$ for all plans that achieve G and satisfy O. Posterior probabilities of goals given observations, $P(G|O)$, can also be computed by considering the cost difference of plans, or Δ, that achieve G and O and achieve G and not O as in (Ramírez and Geffner 2010).

Next, we put everything together and define the problem we address in this paper.

Definition 5 (MAPR Problem with Temporal Actions) *The Multi-Agent Plan Recognition (MAPR) problem with temporal actions is described as* $P = (F, I, O, \mathcal{G}, \textsc{prob}, N, \{A_i\}_{i=1}^n)$*, where:*

- F *is a set of fluents,*
- $I \subseteq F$ *defines the initial state,*
- $O = [o_1, ..., o_m]$*, where* $o_i \in F$*,* $i \in [1, m]$ *is the sequence of observations,*
- $\mathcal{G}$ *is the set of possible goals* G*,* $G \subseteq F$*,*
- PROB *is the goal priors,* $P(G)$
- $N = \{1, ..., n\}$ *is the number of agents, and*
- A_i *is a set of temporal actions for agent* i *as defined in Definition 2.*

We define the solution to the MAPR problem with temporal actions to be the probability of plans given observations, $P(\pi|O)$, and the probability of goals given observations, $P(G|O)$. Note, the notion of concurrency amongst actions is now modeled via the temporal actions rather than through joint actions and the defined concurrency constraints over them. Further note that the use of joint actions indicates that

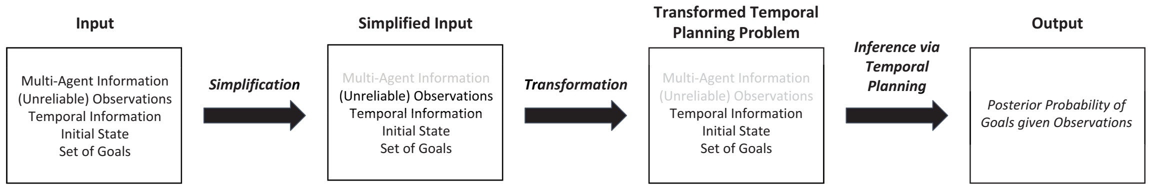

Figure 2: A pipeline showing our proposed compilation approach which consists of transforming the original MAPR problem with temporal actions and unreliable observations into a plan recognition problem, and a transformation step that compiles away the observations, allowing the use of temporal planning to compute the posterior probabilities of goals (and plans).

no two actions from the same agent occur concurrently, but actions from different agents can occur concurrently. Use of temporal actions allows concurrency of the agent's actions as well as actions of different agents.

The observations over a set of actions can be encoded in our model using special unique predicates called for example, "happened_start_a" and "happened_end_a" that are added to each action either to the "add_s" list or the "add_e" list. This can effectively model occurrence of an action.

Note that the goals being pursued by the agents need not necessarily be unrelated to one another; such a goal might be a combination of goals, e.g., a conjunction, disjunction or a subset. For example, an agent assisting two other agents who are pursuing goals G_1 and G_2 respectively, might have the goal $G_1 \cup G_2$ in its set of goals. Thus, $\mathcal{G}$, the set of possible goals pursued by the agents, might contain complex goals rather than one common goal shared by all agents.

3 Transformation to Planning

In the previous section we defined a MAPR problem P and its solution. In this section, we describe a multi-step compilation technique that allows use of planning on this problem. That is, we transform the given MAPR problem P as defined in Definition 5 into a familiar plan recognition problem as defined in Definition 4, and propose the use of temporal planning to compute the posterior probabilities in keeping with the previous plan-recognition-as-planning approaches.

The multi-step compilation pipeline is shown in Figure 2. The pipeline consists of transforming the original MAPR problem with temporal actions and unexplainable observations into a plan recognition problem, and a transformation step that compiles away the observations, allowing the use of temporal planning to compute the posterior probabilities of goals. Depending on the approach, the pipeline also allows computation of posterior probabilities of plans.

3.1 Transformation to Plan Recognition Problem

To transform the original MAPR problem with unexplainable observations to a plan recognition problem with temporal actions, we exploit the technique proposed by Muise et al. 2015 in the Competition of Distributed and Multi-Agent Planners, and modify it so that it creates a temporal planning problem instead of a classical planning problem (i.e., the temporal actions are unchanged and left as temporal). The transformation step creates a temporal planning domain, where the privacy of fluents and objects is respected. Muise et al. 2015 adhere to a privacy model that restricts the number of objects and fluents that an agent has access to; that is, agent i should be able to execute action a if and only if action a is private to agent i or action a is public to all agents. This is done using special predicates that keep track of an agent's access to fluents and objects; every object o and agent i in the domain are assigned a corresponding fluent. For an agent i to be allowed to execute an action on object o, a precondition must be met, in which the corresponding fluent holds. Finally, the privacy model is incorporated into the initial state. The above translation is both sound and complete, and any temporal planner can use the resulting encoding to produce solutions that do not violate the privacy model for the agents. This transformation step is shown as the simplification step in Figure 2 and results in a plan recognition problem where the planning domain is temporal. Although the notion of privacy did not play an explicit role in our experimentation and was not the focus of this paper, the technique we use here respects models of privacy and will therefore enable us to address the role of privacy in the Multi-Agent Plan Recognition Problem.

3.2 Transformation to Temporal Planning

Next, we compile away the observations, so that the plan recognition problem can be solved using the plan-recognition-as-planning approaches. This is referred to as the transformation step in Figure 2. Note that there are several ways to compile away the observations depending on the nature of the given observations. For example, if the observations are actions then one can take the approach described by Ramírez and Geffner 2009. Observations can also be compiled away following Haslum et al. (Haslum and Grastien 2011) using a so called "advance" action that ensures the observation order is preserved. In this paper, observations are defined over the fluents, so we will follow the technique proposed in (Sohrabi, Riabov, and Udrea 2016). We modify this technique slightly to allow the computation of the probabilities using the approach proposed in (Ramírez and Geffner 2010).

The transformation step compiles away observations, using special predicates for each fluent in the observation sequence O, and ensuring that their order is preserved. To address the unexplainable observations, the set of actions, A, is augmented with a set of "discard" and "explain" actions for each observation o_i in the observation sequence, O, with a penalty for the discard action. We set the penalty by defining a high duration to the "discard action"; whereas in Sohrabi et al. 2016 the penalty was set by defining a high cost to the "discard action". This penalty encourages the planner to explain as many observations as possible. We also update the duration of the original action, by adding a constant duration to all of them. This is the penalty for the possible missing observations. To ensure that at least one of the given goals $G \in \mathcal{G}$ is achieved, and allow the use of a diverse planner that finds a set of plans, a special predicate "done" in addition to the corresponding predicate for the final fluent in the observation sequence are added to the goal of this transformed planning problem. In addition, we add an action for each goal $G \in \mathcal{G}$ with precondition g, and effect "done" to the set of actions.

Theorem 1 *Given a MAPR problem with temporal actions $P = (F, I, O, \mathcal{G}, \text{PROB}, N, \{A_i\}_{i=1}^n)$ as defined in Definition 5, and the corresponding planning problem $P' = (F', A', I', G')$ as described above, for all $G \in \mathcal{G}$, if π is a plan for P, 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); furthermore, $d(\pi') = b_1 \cdot d(\pi) + b_2 \cdot N$, where $d(\pi)$ is the duration of the plan, b_1 and b_2 are positive coefficients that express weights to the different objectives, and N is the number of unexplainable (aka noisy) observations.*

Proof is based on the fact that the extra actions (i.e., explain, discard, and goal) only preserve the ordering amongst the observations and do not change the state of the world. Note, the duration of the new transformed planning problem takes into account the objective function that includes the original duration of the actions as well as the number of missing and unexplainable actions. We assign the weights of b_1, the added duration to all original actions, for the combined duration of the original action and the missing observations, and b_2, the penalty or the duration of the discard action, for the unexplainable observations. The probabilities, $P(G|O)$ and $P(\pi|O)$, can be then computed using the duration of the plans in the transformed planning problem.

To apply the approach proposed in (Ramírez and Geffner 2010), as well as our proposed "Hybrid" approach, we modify the transformation step discussed above to not include the "done" predicate as a new planning problem will be generated for each goal separately. In addition, the discard actions are removed for our proposed approach that is based on (Ramírez and Geffner 2010) as this approach does not address the unexplainable observations by discarding them.

4 Computation

We present three approaches to solve the transformed planning problem and compute the posterior probabilities of the different goals. Note, to be consistent we only focus on computing the posterior probabilities of the different goals and will not discuss computation of posterior probabilities of the plans as it is not supported in all three of our approaches. The first approach is based on finding, for each of the different goals, the delta between the costs of two plans, one that explains the observations and one that does not; this method is a modification of the approach suggested in (Ramírez and Geffner 2010). The second approach is based on finding a set of diverse plans and is a modification of the proposed approach in (Sohrabi, Riabov, and Udrea 2016). The third approach is a combination of the two previous approaches, in that it computes a set of diverse plans for each of the goals.

4.1 Approach 1 : Delta

Given the transformed temporal planning problem, this approach computes the posterior probability of a goal given the observations, $P(G|O)$ by running the planner twice for each goal, once with the observations, and once without. More formally, $P(G|O)$ is computed using Bayes Rules as:

$$P(G|O) = \alpha P(O|G) P(G) \tag{1}$$

where α is a normalization constant and $P(G)$ is PROB or the goal priors. The cost difference, or Δ, is defined as the difference in the cost of the optimal plan that achieves G and O, and the cost of the optimal plan that archives G but not O. $P(O|G)$ is defined as:

$$P(O|G) \approx \frac{e^{\beta\Delta}}{1 + e^{-\beta\Delta}} \tag{2}$$

where β is a positive constant. This approach assumes that the agent pursing goal G is more likely to follow cheaper plans. It also assumes that the probability that the agent is pursing a plan for goal G is dominated by the probability that the agent is pursing one of the most likely plans for goal G; hence, it only computes one plan (i.e., the optimal plan) for each setting of the problem. Further, this approach does not address discarding the unexplainable observations explicitly, as mentioned in the previous section.

4.2 Approach 2 : Diverse

Given the transformed temporal planning problem, this approach computes both the posterior probability of the plans as well as goals by running a diverse temporal planner on the transformed temporal planning problem. It then uses the following formulas to approximate the probabilities. In particular, it first computes $P(\pi|O)$ as follows:

$$P(\pi|O) = \beta P(O|\pi) P(\pi) = \beta P(O|\pi) P(\pi|G) P(G) \tag{3}$$

where β is a normalizing constant that depends on $P(O)$ only, and $P(O|\pi)P(\pi|G)$ is approximated as follows:

$$P(O|\pi) \cdot P(\pi|G) \approx 1 - \frac{\beta' V(\pi)}{\sum\limits_{\pi' \in \Pi} V(\pi')} \tag{4}$$

where β' is a positive constant, used to offset large sums, Π is a sampled set of plans that satisfy the observations and achieve at least one of the goals $G \in \mathcal{G}$, and $V(\pi) = b_1 \cdot d(\pi) + b_2 \cdot N$ is the value of the weighted factor over two objectives for the plan π that achieves goal G and satisfies the observation O: (1) the original duration of the actions in the domain, (2) the duration of the discard actions (i.e., unexplainable observations). Coefficients b_1 and b_2 are used to give weights to the importance of the original actions and the discard actions respectively; N is the number of unexplainable observations. The missing observations are not addressed separately, but rather handled implicitly by the coefficients b_1 for the duration of the original actions. Following the transformation to planning described earlier, the value of $V(\pi)$ is equivalent to duration of the plan π.

The posterior probabilities of goals given observations are then computed by a summation over all the values of $P(\pi|O)$ for the sampled set of plans, Π, that achieve G and satisfy O.

$$P(G|O) = \sum_{\pi \in \Pi} P(\pi|O) \tag{5}$$

The set of plans Π is computed using diverse planning, where the objective is to find a set of plans m that are at least d distance away from each other. 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. It is possible to also use the top-k planning approach, where a set of (possibly diverse) high-quality plans are found (Riabov, Sohrabi, and Udrea 2014). However, the top-k planning approaches do not address temporal domains. We have tried to compile away the temporal aspects of these domains, using the approach described in (Celorrio, Jonsson, and Palacios 2015), but we faced challenges regarding scalability.

4.3 Approach 3: Hybrid

In this approach, we again use the diverse temporal planner used in the previous approach, this time to compute a smaller set of plans for each of the different goals. After merging the sets of diverse plans, we are then able to compute the posterior probabilities just as we did in the previous approach, given the merged set of diverse plans. This approach forces the diverse planner to compute a set of plans for each of the goals, rather than allowing it to choose the goal that is shortest to reach, as is done in Approach 2. Thus, each of the possible goals is assigned at least one representative plan that is taken into account when computing the posterior probabilities of the different goals; in doing so, we ensure that every goal is covered and accounted for. As shown in our experiments, the results show that this approach, on average, achieves the highest coverage rates across all domains.

5 Experimental Evaluation

In this section, we evaluate our three proposed approaches, using a diverse planner, LPG-d (Nguyen et al. 2012) for the diverse planning approach and the hybrid approach, and a temporal planner LPG-TD (Gerevini, Saetti, and Serina 2004) for the delta approach. We chose these planners as we were able to run them successfully, using the transformed planning problem as input. The other planners we have tested, (e.g., POPF2 (Coles et al. 2010), OPTIC (Benton, Coles, and Coles 2012)), either timed out on most problem instances, or did not accept the transformed planning problem as the input. Note, the results for the diverse approach were obtained by running the diverse planner LPG-d once for each problem. For the hybrid approach, the diverse planner was run once for each goal, that is $|G|$ times. For the delta approach, LPG-TD was run $2 \times |G|$ times. We used a timeout of 30 minutes and ran all our experiments on dual 16-core 2.70 GHz Intel(R) Xeon(R) E5-2680 processor with 256 GB RAM. For the LPG-D planner we used two settings of (m, d), $(10, 0.2)$, $(50, 0.2)$, but report only on the $(10, 0.2)$ case because this setting performed better with respect to recognizing a goal; 10 plans that are at least 0.2 distance away from each other. For the coefficients, we set b_1 to be the maximum of all action durations in the domain, and we set b_2 to be ten times b_1; hence, we assign a higher penalty for the unexplained observations, and a lower penalty for the missing observations.

In this paper, we address a combination of elements that has not been addressed by previous research; hence, we create, for evaluation purposes, a set of novel benchmarks, based on the International Planning Competition (IPC) domains and the Competition of Distributed and Multiagent Planners (CoDMAP), namely Rovers (a domain where a collection of rovers navigate a planet surface, finding samples and communicating them back to a lander), Depots (a domain in which trucks can transport crates around and then the crates must be stacked onto pallets at their destinations), Satellites (a domain which requires planning and scheduling a collection of observation tasks between multiple satellites, each equipped in slightly different ways), and ZenoTravel (a domain which requires the transportation of people around in planes, using different modes of movement). The original domains are either temporal or are multi-agent and are not plan recognition problems. We modify the domains to create benchmark problems for the multi-agent plan recognition problem with temporal actions. In combining the multi-agent and temporal aspect of the problems, we had to address a number of issues that overlooked concurrency. For example, in the Rovers domain, two rovers are able to sample the same rock sample at the same time, and when one of the rovers is done sampling, the other can still work on the same sample, although it is depleted; in the Depots domain, the effect of a "lift" action, executed on a crate, is that the hoist is now lifting the crate. The effect should only be applied at the end of the action's execution time, yet in the original domain it is immediately applied at the start. We have used a modified version of the domains, where these issues are addressed.

To construct the plan recognition problems, we computed a number of plans that are a solution to the original planning problems. From these plans, we sample actions in order to construct O, the sequence of observations, while keeping track of the goal used in the original planning problem (i.e., ground truth goal). To evaluate how well the approaches address missing observations, we created several problems that do not have the full observation sequence (i.e., some observations are missing). We did so by randomly selecting 10%, 40%, 70% and 100% 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 added extra observations, which were not observed, to the original observation sequence.

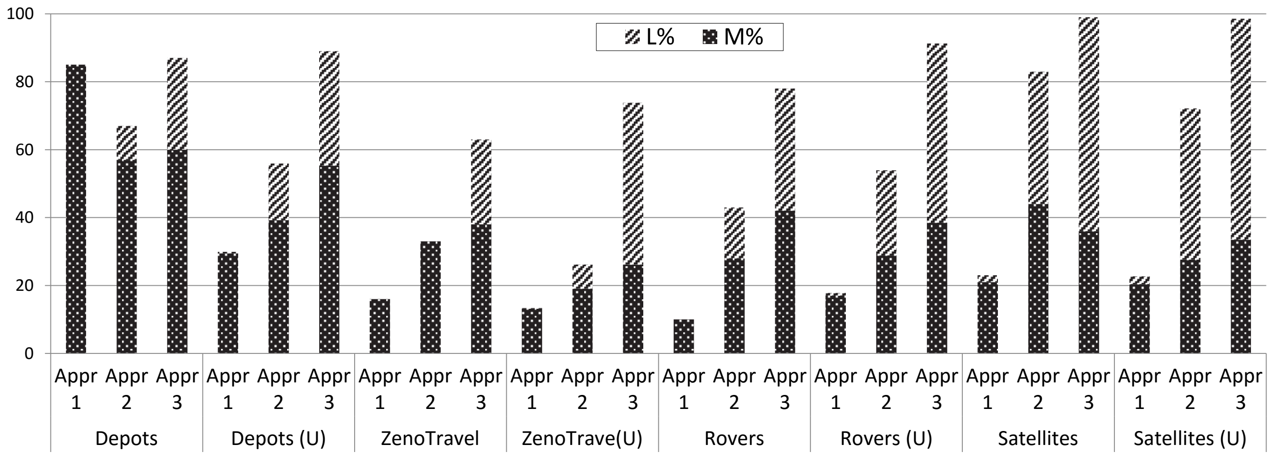

Figure 3: Comparison of our three proposed approaches for recognizing a goal: (1) delta, (2) diverse, and (3) hybrid. The comparison is made for each of the four domains, with and without the introduction of unreliable observations; (U) signifies that the results are the average over all instances where unreliable observations were introduced in that specific domain. The dotted and dashed portions of the bars show the average percentage of instances in which the ground truth goal was deemed Most and Less likely respectively, i.e., the metrics M and L. The height of each bar is the overall goal recognition coverage.

Figure 3 shows the summary of the results when evaluating our three proposed plan recognition approaches for recognizing a goal. Approach 1 is the delta approach, Approach 2 is the diverse approach and Approach 3 is the hybrid approach. For each setting of the problem, we generated two ground truth plans, and there are 20 problems in each domain; the problems vary in difficulty, i.e., number of objects, agents etc., thus the more difficult problems are computationally more complex. In addition, we have experimented by adding a number of extra observations, i.e., noise introduced to the problem; there are two levels of noise, one of which adds 12% extra, possibly unexplainable observations relative to the number of original observations, while the other adds the same percentage of extra observations, only this time relative to the size of the ground truth plan. The extra observations were generated similarly to what was done in Sohrabi et al. 2016; in this work, however, rather than introducing a fixed number of extra observations, the amount of noise is calculated based on the size of the observations sequence and the size of the ground truth plan. Thus, the level of noise is better adapted to a specific problem. The figure presents the results for each of the four domains, with and without the introduction of unreliable observations. (U) signifies that the results are an average over all cases where unreliable observations were introduced in a specific domain, and of all 20 problems in the domain.

To evaluate the coverage and accuracy of the different approaches, we compute the approximation of the posterior probability of a goal given the observations, $P(G|O)$ (not shown in the Figure), in addition to the average percentage of instances in which the ground truth goal was deemed Most and Less likely, i.e., whether or not the ground truth goal was assigned the highest probability. These values, M and L, are shown respectively in the dotted and dashed portions of the bars in the figure. The overall value of M and L, sum of the most and less likely percentages, indicates the goal recognition coverage for that method, and is expressed by the total height of each bar. 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 0.03 posterior probability.

The results in Figure 3 show that Approach 1 does best (i.e., highest M value on average) in the Depots domain when observations are reliable or no noise is introduced, yet it is very sensitive to noise and does much worse than the other approaches when observations are unreliable (i.e., missing or unexplainable). Approach 1, on average, also performs best with regard to the approximation of the posterior probability of the ground truth goal given the observations, $P(G|O)$, in the Depots domain; however, as with the M value, Approach 1 is highly sensitive to noise. Overall, when noise was introduced, Approach 3 performs best with regard to $P(G|O)$. In addition, the results show that on average, Approach 3 achieves the best coverage, i.e., the total height of the bars, across all domains. Furthermore, the results show that on average, Approach 3 achieves the best precision in detecting the ground truth goal when unexplainable observations are added, as indicated by the height of the dotted portions of the bars where (U) is present. On average, over all our problems, when unreliable observations were not introduced, Approach 1 had 34% coverage, Approach 2 had 56% coverage, and Approach 3 had 82% coverage. When unreliable observations were introduced, Approach 1 had 21% coverage, Approach 2 had 52% coverage, and Approach 3 had 88% coverage.

Note that while one would expect to see the best results in cases where unreliable observations are not introduced, i.e., sections in the figure where (U) is absent, this is not the case in general. This can be attributed to the fact that the ground truth plans from which the observations are sampled, are sub-optimal, which makes the task of recognizing the ground truth goal more challenging. For example, the system might attempt to explain an observation o that has been sampled from a sub-optimal plan that achieves goal G; observation o correlates to a part of the plan, which causes the overall duration to be higher, e.g., an unnecessary detour; in this case, the system will deem it less likely that the agent is pursuing goal G given observation o, due to the duration-sensitive nature of all computational approaches in this paper. It is also worth noting that in some cases, the sheer amount of observations caused some problem instances to become computationally challenging, and often led to the system timing out; this can explain why the results are, in some cases, worse when less unreliable observations are introduced. Additionally, since the extra observations are added randomly, in some cases the observations are unexplainable, while in other cases, it is possible for the system to explain the extra observations by computing very long and costly plans. This, combined with the fact that the ground truth plans are sub-optimal, can account for some of the unexpected results, for example in cases where introduced noise does not hurt performance. The nature of the domains, e.g., interchangeable objects or inconsequential order of action execution, causes some of them to be less sensitive to this issue than others.

6 Related work

There exists a body of work on multi-agent systems. The closest to our work is the work of (Crosby, Jonsson, and Rovatsos 2014) that is also concerned with the notion of concurrency. However, they do not model durations and the constraints are over the objects. Hence, the actions are not temporal in the sense of PDDL 2.1 (Fox and Long 2003). They also do not address the plan recognition problem or the observations. There are many other multi-agent systems that do not address durations and temporal constraints (e.g., (Kominis and Geffner 2015; Muise et al. 2015; Bisson, Larochelle, and Kabanza 2015)).

The plan recognition problem, in many variations and forms, has been addressed by previous work (e.g., (Banerjee, Lyle, and Kraemer 2015; Zhuo, Yang, and Kambhampati 2012; Kominis and Geffner 2015; Sukthankar et al. 2014)). However, the focus and the problem addressed are different. In particular they either do not address the temporal aspect of the actions, or do not use AI planning to compute the solution. The problem we address in this paper focuses on use of AI planning to address the multi-agent plan recognition problem with temporal actions and unexplainable observations. To the best of our knowledge, we are the first to propose this problem and provide a solution for it.

7 Summary and Future Work

In this paper, we address the problem of MAPR with temporal actions and unreliable observations. To this end, we first characterize the problem, and then propose a multi-step compilation technique that enables the use of AI planning for the computation of the posterior probabilities of the possible goals. In addition, we propose a set of novel benchmarks that allow for a standard evaluation of solutions that address the MAPR problem. We present results of an experimental evaluation of our approach on this set of benchmarks, using several temporal and diverse planners.

The work in this paper allows for better reasoning about the goals and plans of the different agents; our inferences enjoy much greater depth when the different elements, namely temporal actions, a multi-agent setting and unreliable observations, are addressed. However, better reasoning does not come without a price, and indeed the problem this paper addresses turns out to be quite complex, due to the different elements it consists of, which makes its computational solution expensive. Recent work (E-Martín, R-Moreno, and Smith 2015) suggests an approach that propagates cost and interaction information in a plan graph, and uses this information to estimate posterior probabilities for the different goals. Using this proposed approach, or other faster approaches, could potentially yield better results.

Furthermore, the notion of concurrency that is addressed in the current work is limited to temporal overlapping of actions. We do not explicitly set concurrency constraints over resources, as was done in (Crosby, Jonsson, and Rovatsos 2014). Moreover, in our experiments, the agents are pursing one identical common goal, whereas our approach enables us to address different agents pursuing different, possibly competing goals. Additionally, the goals pursued by the agents, as mentioned in previous sections, might be complex, i.e., a combination of other goals. Hence, as part of future work, it will be interesting to experiment with domains that offer greater interaction between agents, introducing the notion of required concurrency (Cushing et al. 2007).

There are several aspects of multi-agent planning and plan recognition that are important but are not a focus of this paper. Two such aspects are privacy and the beliefs of an agent (Kominis and Geffner 2015); regarding privacy and what is or is not accessible to the agents and the system, it will be highly interesting, as part of future work, to experiment by allowing the system and the agents only partial observability. In our experiments, the system was able to observe all effects, caused by the agents' actions, while in reality this is often not the case, and omniscience is not commonly found. Further, when considering epistemic and belief states of an agent in a plan recognition context, it is worth discussing the following; given that the system has recognized the goal of a specific agent, should this goal be broadcast to all other agents? We must consider that an agent might behave differently based on this knowledge. For example, if agent i becomes idle and knows that agent j is pursuing goal G, then it might offer its help to agent j in pursuing that goal. This might prove a basis for better cooperation between agents.

To conclude, our approach enables the application of a multi-agent plan recognition approach to previously unaddressed problems, by modeling them in temporal planning domains. By enabling the use of existing temporal planners, one can choose the planner that works best for a specific domain and quickly compute a solution to their multi-agent plan recognition problem.

Acknowledgements: We gratefully acknowledge Karin

Niemantsverdriet's assistance with the design and creation of figures; and the Laboratory for Analytic Sciences (LAS) and the Natural Sciences end Engineering Research Council of Canada (NSERC) for funding this work. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSERC, the LAS and/or any agency or entity of the United States Government.

References

Banerjee, B.; Lyle, J.; and Kraemer, L. 2010. Multi-agent plan recognition: Formalization and algorithms. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI)*.

Banerjee, B.; Lyle, J.; and Kraemer, L. 2015. The complexity of multi-agent plan recognition. *Autonomous Agents and Multi-Agent Systems* 29(1):40–72.

Benton, J.; Coles, A.; and Coles, A. 2012. Temporal planning with preferences and time-dependent continuous costs. In *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS)*.

Bisson, F.; Larochelle, H.; and Kabanza, F. 2015. Using a recursive neural network to learn an agent's decision model for plan recognition. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 918–924.

Brafman, R. I., and Domshlak, C. 2008. From one to many: Planning for loosely coupled multi-agent systems. In *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS)*, 28–35.

Carberry, S. 2001. Techniques for plan recognition. *User Modeling User-Adapted Interaction* 11(1-2):31–48.

Celorrio, S. J.; Jonsson, A.; and Palacios, H. 2015. Temporal planning with required concurrency using classical planning. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS)*, 129–137.

Charniak, E., and Goldman, R. P. 1993. A bayesian model of plan recognition. *Artificial Intelligence* 64(1):53–79.

Cohen, P. R.; Perrault, C. R.; and Allen, J. F. 1981. Beyond question answering. *Strategies for natural language processing* 245–274.

Coles, A.; Coles, A.; Fox, M.; and Long, D. 2010. Forward-chaining partial-order planning. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS)*, 42–49.

Crosby, M.; Jonsson, A.; and Rovatsos, M. 2014. A single-agent approach to multiagent planning. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, 237–242.

Cushing, W.; Kambhampati, S.; Mausam; and Weld, D. S. 2007. When is temporal planning really temporal? In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 1852–1859.

E-Martín, Y.; R-Moreno, M. D.; and Smith, D. E. 2015. A fast goal recognition technique based on interaction estimates. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 761–768.

Fox, M., and Long, D. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research* 20:61–124.

Gerevini, A.; Saetti, A.; and Serina, I. 2004. LPG-TD: a fully automated planner for PDDL 2.2 domains. In *14th Int. Conference on Automated Planning and Scheduling, system demo session (ICAPS-04)*.

Haslum, P., and Grastien, A. 2011. Diagnosis as planning: Two case studies. In *International Scheduling and Planning Applications woRKshop (SPARK)*, 27–44.

Kautz, H. A., and Allen, J. F. 1986. Generalized plan recognition. In *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI)*, 32–37.

Kominis, F., and Geffner, H. 2015. Beliefs in multiagent planning: From one agent to many. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS)*, 147–155.

Muise, C. J.; Belle, V.; Felli, P.; McIlraith, S. A.; Miller, T.; Pearce, A. R.; and Sonenberg, L. 2015. Planning over multi-agent epistemic states: A classical planning approach. In *Proceedings of the 29th National Conference on Artificial Intelligence (AAAI)*, 3327–3334.

Muise, C.; Lipovetzky, N.; and Ramirez, M. 2015. MAP-LAPKT: Omnipotent multi-agent planning via compilation to classical planning. In *Competition of Distributed and Multiagent Planners*.

Nguyen, T. A.; Do, M. B.; Gerevini, A.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012. Generating diverse plans to handle unknown and partially known user preferences. *Artificial Intelligence* 190:1–31.

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

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

Riabov, A.; Sohrabi, S.; and Udrea, O. 2014. 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)*, 10–16.

Schmidt, C. F.; Sridharan, N.; and Goodson, J. L. 1978. The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence* 11(1-2):45–83.

Sohrabi, S.; Riabov, A.; and Udrea, O. 2016. Plan recognition as planning revisited. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*.

Sukthankar, G.; Geib, C.; Bui, H. H.; Pynadath, D. V.; and Goldman, R. P. 2014. *Plan, Activity, and Intent Recognition*. Boston: Morgan Kaufmann.

Zhuo, H. H.; Yang, Q.; and Kambhampati, S. 2012. Action-model based multi-agent plan recognition. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, 377–385.