UTRECHT UNIVERSITY

Broadening the Scope of Multi-Agent Plan Recognition: Theory and Practice

by

Maayan Shvo

A thesis submitted in partial fulfillment for the degree of Master of Science in the Subject of Artificial Intelligence

in the
Department of Information and Computing Sciences

Supervisors:
Prof. Sheila McIlraith & Dr. Mehdi Dastani

July 2018

UTRECHT UNIVERSITY

Abstract

Department of Information and Computing Sciences

Supervisors:
Prof. Sheila McIlraith & Dr. Mehdi Dastani

Master of Science in the Subject of Artificial Intelligence

by Maayan Shvo

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 made various strong assumptions which have limited their applicability to a restricted set of real-world instantiations of the MAPR problem. In order to broaden the applicability of MAPR to a wider range of problems, in this thesis we characterize two novel formulations of the MAPR problem, each relaxing different assumptions made by previous work. The first formulation defines the Epistemic MAPR problem, which no longer assumes that all agents must share a common mental state. This, in turn, enables the observing agent to consider the unique perspective of each observed agent when detecting its likely plans and goals. The second formulation defines the MAPR problem with temporal actions and unreliable observations. This formulation relaxes the assumptions that (a) the agents' actions are instantaneous and (b) the observations are perfect and reliable. Importantly, the thesis proposes to conceive the computational core of the MAPR problem as an AI planning task, thus enabling the use of existing planning tools. The thesis then introduces different AI planning-based computational approaches which solve the novel formulations of the MAPR problem by solving the corresponding planning problems. Finally, the thesis illustrates the power and flexibility of the proposed computational approaches by demonstrating their applicability to various, previously unaddressed, MAPR problems.

Acknowledgements

I would like to thank the wonderful human mosaic, my support network, that helped make this thesis (and Master's) possible. My family, for (mostly remotely) being there for me, unconditionally, and supporting my crazy journeys. Tara, for being part of my life, for your feedback, for proofreading, and for our countless agent Claire & agent Tara conversations. I would also like to thank Prof. Sheila McIlraith and Dr. Mehdi Dastani, my supervisors, for giving me the opportunity to carry out this project and for providing me with invaluable guidance throughout the process. Mehdi, thank you for your much appreciated feedback. Sheila, your perspective never fails to elucidate the strengths, as well as the weaknesses, of the work and ideas brought before you; thank you for the invitation to work on my thesis in Toronto. I would like to thank Karin Niemantsverdriet for her much appreciated counsel and support with some of the visual elements in this thesis. Finally, I would like to greatly acknowledge Dr. Shirin Sohrabi, who was an extraordinary mentor to me while I took my first steps in the world of research. The joint work with Dr. Sohrabi laid the foundations for this thesis, and I am grateful for her counsel and friendship. I would like to extend my gratitude to NSERC (Natural Sciences and Engineering Research Council of Canada) for supporting my stay in Toronto.

Contents

Chapter 1

Introduction

1.1 Background

With an ever-increasing interaction between machines and humans, coupled with the ever-growing responsibilities bestowed upon our artificially intelligent companions, an important goal for AI research is the creation of machines that are able to seamlessly and naturally understand and assist humans. As part of that effort, research on Plan, Activity and Intent Recognition (PAIR) [1], which facilitates reasoning about the actions of agents (be they artificial or human) and the underlying motivation for those actions, is key if we are to achieve successful human-machine and machine-machine coordination, cooperation and communication. Within research in PAIR, Plan Recognition (PR) – 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-machine and machine-machine interactions (e.g., [2]). Interesting and impactful applications of PR abound: at IBM, researchers have demonstrated how PR might be used in malware detection [3] and enterprise risk management ([4], [5]); further, PR has been applied to many real-world problems, including a problem literally out of this world. NASA researchers enabled assistive robots aboard the International Space Station (ISS) to recognize the goals of astronauts aboard the ISS, using PR, and subsequently assist them in achieving these goals [6]; finally, PR has been successful in facilitating human-robot cooperation in search & rescue scenarios [7]: a robot, imbued with PR capabilities, was able to predict a human teammates triage-related plans and goals and help her achieve them (e.g., by proactively picking up a medical kit from a faraway location, reasoning that she would require it to perform triage on a victim).

While originally conceived in the context of single-agent plan recognition, recent research has focused on the more complex task of Multi-Agent Plan Recognition (MAPR) (e.g, [8], [9], [10]). In MAPR, the goals or plans of multiple agents are hypothesized, based upon observations of the agents, providing a richer paradigm for addressing many real-world applications of MAPR. Among these applications are wide area surveillance (where the actions performed by a large number of individuals, in a large physical area, are monitored); game-playing (where MAPR can be used to allow an AI system to reason about the strategies and tactics of multiple agents, allowing it to become more challenging/helpful to the human user [11]); cyber attacks (where the attacks, carried out and divided amongst multiple attackers, can be recognized by MAPR systems); and military domains (where MAPR can recognize threatening intentions by a group of enemy ships against friendly naval ships [12]).

1.2 Research Problems

While the different formulations of MAPR, as proposed by previous work and surveyed in Chapter 2, are effective for certain classes of problems, they do not capture important nuances that are evident in many real-world MAPR tasks. This is due, in part, to the assumptions made by previous work, with respect to the MAPR paradigm, about e.g. the nature of the agents and the environment. This thesis therefore explores whether novel formulations of the MAPR problem, addressing a myriad of properties and allowing for enhanced expressivity, are paramount to the applicability of a MAPR approach to many real-world instantiations of the MAPR problem. To this end, the main research problem addressed by this thesis is as follows:

Characterize and solve novel multi-agent plan recognition formulations which are applicable to a wide range of real-world instantiations of the MAPR problem

In order to characterize novel MAPR formulations, we first review the existing body of MAPR research, which greatly varies in terms of the computational approaches used, the assumptions made and the way in which the MAPR problem and its solution are construed. In turn, this review will help make explicit the assumptions made by previous work, which restrict their applicability to a limited set of real-world instantiations of the problem. Thus, the first sub-problem addressed by this thesis is:

Sub-problem #1 *- Review previous work in MAPR, in order to make explicit the assumptions that have been made by this body of research*

Next, after making explicit the assumptions made by previous MAPR work, we characterize and solve novel MAPR formulations, which relax some of these assumptions. Thus, the second and third sub-problems addressed by the thesis are:

Sub-problem #2 - *Characterize novel formulations of the MAPR problem which relax assumptions made by previous research*

Sub-problem #3 - *Develop computational approaches which solve the proposed formulations of the MAPR problem*

Finally, we show that these formulations are applicable to a wide range of real-world instantiations of the MAPR problem. Hence, the fourth sub-problem addressed by the thesis is:

Sub-problem #4 - *Demonstrate that the proposed MAPR formulations are applicable to various real-world scenarios*

1.3 Approach to Solving the Research Problems

In order to address *sub-problem #1*, we first enumerate a set of attributes which delineates the general case of the MAPR problem; second, we characterize previous formulations of the MAPR problem as different configurations of the enumerated attributes, thus making explicit the assumptions made by previous work.

Next, we turn to solve *sub-problem #2* by characterizing two novel formulations of the MAPR problem, which relax some of the assumptions made by previous MAPR work. By relaxing assumptions made by previous work, these formulations will allow for an enriched characterization of the problem, providing support for a more robust representation of the e.g. capabilities and mental states of the agents, as well as the nature of observations. In this thesis, we choose to relax three assumptions made by previous work.

The first assumption we relax is that agents have a shared mental model and a representation of both the plan and the world. One of the challenges which arise when transitioning from the single-agent plan recognition case to the multi-agent case (and, more generally, from a single-agent setting to a multi-agent one), is that the different agents might not possess the same representation of the plan, even if they are working together. For example, two search & rescue robots, working together in a disaster zone, might form different plans due to differing capabilities and communications received. One robot might decide not to explore a certain location, because it has not received a communication which the other robot has received. Previous MAPR work attributed identical perspectives to the different agents, however, this is not a realistic assumption and the agents may very well have different motivations, intentions and beliefs regarding the collaborative or individual plan they are executing and participating in. The agents may also differ in their perception of the world, where a certain agent might, for example, have limited perceptual access to the environment, which affects the actions it is likely to take towards achieving their goal. To truly capture the multi-agent aspect of the MAPR problem, in this work, we no longer assume that agents share an identical perspective of the world in general and of the plan which they are pursuing in particular.

The second assumption we relax is that the recognition system is given a perfect and reliable observation sequence. In many real-world applications of MAPR, observations are unreliable, i.e., unexplainable or missing; they are often over properties of the world rather than actions, and the available observations may not be explainable by the agents' goals and plans. Unexplainable observations could arise as a result of a faulty sensor or communication error. In the case of missing observations, the recognition system is given an incomplete set of observations. While some previous work in MAPR addressed incomplete observation sequences, unexplainable observations went unaddressed.

The third assumption we relax is that actions have no durations and are instantaneous. In many real-world scenarios, there is need for the actions of the agents to be temporal or concurrent. Much previous work in MAPR elected to use joint instantaneous actions in order to model concurrency of actions of agents (or did not allow for concurrency at all). However, using joint actions to model concurrent actions performed by multiple agents is restricting in that a single agent cannot perform two actions concurrently. The relaxation of this assumption and the use of temporal actions allow concurrency of a single agent's actions as well as actions of different agents. In addition, introducing action durations allows for more expressivity, as it enables to more precisely form a timeline and position each of the agents (and their actions) along it, given observations.

Next, we turn to address *sub-problem #3* by introducing different AI planning-based computational approaches which solve the proposed formulations of the MAPR problem. Following the characterization of the two novel formulations of the MAPR problem, with the aforementioned relaxed assumptions, we conceive the computational core of the problem as an AI planning task. To conceive the computational core of the problem as a planning task, we turn to the work of Ramírez and Geffner [13], who proposed the Plan Recognition as Planning (PRAP) paradigm, where the plan recognition problem is cast as an AI planning problem, thus allowing the use of off-the-shelf planners on the resulting planning task; the PRAP approach will be elaborated upon in chapters 3 and 4. Once the MAPR problem is cast as a planning problem, the latter is solved and the post-processed solutions allow us to recognize the goals and plans of the agents.

Importantly, our two novel formulations relax the three assumptions mentioned above. Specifically, our first proposed formulation of the MAPR problem, as discussed in Chapter 3, relaxes the first assumption. *In order to relax the first assumption*, our approach forms a localized perspective for the different agents, which will be incorporated into the plan recognition process. An agent's localized perspective depends on various factors, including its perceptual capabilities (e.g., what is the agent able to observe? in what manner can the agent sense the world around it?), its intentional state (e.g., what motivates the agent to execute an action?), as well as its beliefs and knowledge about the world and its relations and interactions with other agents in the environment. In order to form a localized perspective for each agent, the mental state of the agent must first be represented and reasoned about; we elect to use a rich and expressive planning framework, namely Multi-agent Epistemic Planning (MEP), as introduced in [14], and situate the proposed formulation of the MAPR problem within this framework. Using MEP, which is rooted in Dynamic Epistemic Logic (DEL), we are able to explicitly represent the disparate mental states of different agents. Importantly, reasoning about these disparate mental states can help us understand the agents' actions in the context in which they were made. In Chapter 3, the use of this framework, as well as the benefits of doing so, will be discussed and demonstrated.

Our second proposed formulation of the MAPR problem, as discussed in Chapter 4, relaxes the second and third assumptions. *In order to relax the second assumption*, our approach addresses potentially unreliable observations, including unexplainable and missing observations. To address unreliable observations, Sohrabi et al. 2016 modify the definition of satisfaction of an observation sequence by an action sequence, introduced in the original plan recognition as planning paradigm [13], to allow for observations to be left unexplained. We adapt their approach and incorporate it into our formulation of the MAPR problem. Next, *in order to relax the third assumption*, we leverage advances in temporal planning (e.g., [16]) and enable the use of temporal actions with durations, thus allowing for concurrency of a single agent's actions as well as actions of different agents. Differently than the original plan recognition as planning paradigm, our approach focuses on the duration of plans, rather than their cost.

To solve *sub-problem #4*, we illustrate the power and flexibility of our proposed computational approaches, specifically testing their applicability to previously unaddressed real-world scenarios. To do so, domains and scenarios which specifically require the relaxation of the aforementioned assumptions are designed and used. For example, in an environment with a high likelihood of faulty sensors, we would like to relax the assumption that the observations provided by the sensors are perfect and complete. Further, in an environment where communication is limited or unreliable, agents might form different beliefs regarding their environment, due to discrepancies in the communication they have received or in their ability to perceive their environment. Such environments serve as appropriate test beds for the applicability of our approaches to different real-world scenarios and demonstrate the need to relax various assumptions made by previous work. Various off-the-shelf planners are used to solve MAPR problem instances within these environments, and results are gathered. Analysis of these results is done based on a set of metrics aimed at evaluating the plan and goal recognition capabilities of our computational approaches.

1.3.1 Summary and Contributions

To summarize, we outline the contributions made by this thesis by solving the sub-problems listed above: (i) a characterization of two novel formulations of the MAPR problem which relax various assumptions made by previous work; (ii) introduction of different compilation processes that transform the MAPR problem to a planning problem and enable the use of powerful AI planning tools on the transformed planning problem; (iii) introduction of various AI planning-based computational approaches to computing the probability distributions of the goals and plans of agents given the observations; (iv) experimental evaluation of our proposed techniques on a set of novel benchmarks and scenarios in a number of different settings, using several types of off-the-shelf planners.

Finally, by solving the four sub-problems, we demonstrate that our approaches manage to address new classes of important MAPR problems, which were previously unsolvable. This, in turn, demonstrates the need for more general and expressive formulations of the MAPR problem, as advocated for and characterized by the thesis.

Chapter 2

Characterization of the Multi-Agent Plan Recognition Problem

In this chapter, we address *sub-problem #1*, as outlined in section 1.2. We begin by introducing a basic formulation of the multi-agent plan recognition problem and its solution; next, we enumerate a set of attributes which delineates the general case of the MAPR problem; finally, we survey previous work in the field and characterize the surveyed body of work as different configurations of the enumerated attributes.

2.1 Overview of the Multi-Agent Plan Recognition Problem

In this section, we review basic definitions, including necessary planning background, and introduce a basic formulation of the multi-agent plan recognition problem and its solution. We start by formally defining the classical planning problem. In AI planning ([17], [18]), the objective is to generate a partially or totally ordered sequence of actions, a plan, which transforms some initially specified state of the world to a desired (goal) state. Research in the field, spanning many decades, has strived to produce general solutions to this problem, leading to planners which are not domain-specific. That is, general-purpose planners have been created, which are agnostic to problem and domain specific peculiarities, and return a solution, a plan, given some input specified in a generic and standard format (e.g., the Planning Domain Definition Language (PDDL)

[19]). Research in AI planning has led to the creation of powerful and scalable planners, capable of solving problems with trillions of states in fractions of a second [20].

As will be seen in Chapters 3 and 4, and as mentioned previously, we elect to conceive the computational core of the MAPR problem as an AI-planning problem, thus enabling the use of existing planners to solve the MAPR problem and recognize the goals and plans of the agents. We first define the classical planning problem as follows:

Definition 1 *[**Planning Problem**] A planning problem is a tuple of the form $P^c = (F, A, I, G)$, where F is a finite set of fluent symbols, A is a set of actions, $I \subseteq F$ defines the initial state, and $G \subseteq F$ defines the goal state. Each action $a \in A$ is associated with a precondition, Pre_a, add effects, eff_a^+, delete effects, eff_a^-, and non-negative action costs, $COST(a)$.*

A state, s, is a set of fluents that are true. An action $a \in A$ is *executable* in a state s if $Pre_a \subseteq s$. The successor state is defined as $\delta(a, s) = ((s \setminus eff_a^-) \cup eff_a^+)$ 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 I and $G \subseteq \delta(a_n, \delta(a_{n-1}, \ldots, \delta(a_1, I)))$.

Not all realistic problems, however, can be modeled as a classical planning problem. In some real-world problems, involving multiple agents, each possibly having its own goal, we do not wish (and often, it is not possible) to centrally plan for all agents (see further discussion in section 2.2.8). For instance, different robots of different makes could be operating in the same shared environment, each pursuing different goals and wishing to individually plan how to achieve their goals. However, without taking into consideration the dependencies between the tasks of the agents, a conflict might arise when the agents execute their independently planned actions. Thus, in some cases, it is not not possible for each agent to independently solve its planning problem, given the dependencies between the different agents. To address the deficiencies of individual planning in the presence of multiple agents, research in Multi-Agent Planning (e.g., [21], [22], [23]) has focused on addressing the dependencies between agents by enabling them to coordinate their actions and plans. Hence, De Weerdt et al. [24] define multi-agent planning as planning and coordination, combined. They define the multi-agent planning problem as follows:

Definition 2 *[**Multi-Agent Planning Problem (following De Weerdt et al. [24])**] Given a description of the initial state, a set of global goals, a set of (at least two) agents, and for each agent a set of its capabilities and its private goals, find a plan for each agent that achieves its private goals, such that these plans together are coordinated and the global goals are met as well.*

For the purposes of the thesis, we define a simplified Multi-Agent Planning Problem, based on the planning problem defined above.

Definition 3 *[Multi-Agent Planning Problem] A Multi-Agent Planning Problem is a tuple $P^m = (F, \{A_i\}_{i=1}^N, I, G)$, where F and I are defined as in Definition 1, G is the goal of the multi-agent problem, achieved by N agents ($N \geq 2$), each with their own set of action descriptions, A_i, $1 \leq i \leq N$.*

The solution to the multi-agent planning problem, P^m, is a plan $\pi = a_1, ..., a_n$, where a_i is an action belonging to one of the agents, if π is executable from I and $G \subseteq \delta(a_n, \delta(a_{n-1}, \ldots, \delta(a_1, I)))$.

We return now to the main focus of the thesis, the multi-agent plan recognition problem. In plan recognition (and its multi-agent variant) we do not know, a priori, the goals the agents are trying to achieve, nor do we know how they plan to achieve those goals. This stands in contrast to the planning problems defined thus far in this section, where the goal of the agent(s) is known to us (i.e., G). Thus, the objective in plan recognition, given observations about the actions of the agents or the state of the world as affected by the agents' actions, is to recognize the goals and plans of the agents. To this end, the goal state, G, is extended to a set of possible goals, $\mathcal{G}$, which serves as a set of hypotheses for the different goals the agent or agents might be trying to achieve. A more elaborate discussion of $\mathcal{G}$ will be had in Chapters 3 and 4. The plan recognition problem is formally defined as follows:

Definition 4 *[**Plan Recognition Problem**] A plan recognition problem is a tuple of the form $P^r = (F, A, I, O, \mathcal{G}, \textsc{prob})$, where (F, A, I) is the planning domain as defined in Definition 1, $O = [o_1, ..., o_m]$, where $o_i \in A$, $i \in [1, m]$ is the sequence of observations, $\mathcal{G}$ is the set of possible goals G, $G \subseteq F$, and* PROB *is a probability distribution over $\mathcal{G}$, specifying the prior probability of a goal, $P(G)$.*

The solution to the plan recognition problem, P^r, is defined as two probability distributions: the probability of plans given observations, $P(\pi|O)$, and the probability of goals given observations, $P(G|O)$. In previous work, AI planning was used to approximate these probabilities and compile away the sequence of observations, O. We will discuss these techniques in Chapters 3 and 4. PROB is a probability distribution over $\mathcal{G}$ which assigns each possible goal $G \in \mathcal{G}$ a prior probability, $P(G)$. Throughout the thesis, this probability distribution is assumed to be uniform. This is, of course, not always the case; in some instances, some possible goals are known to be more likely than others, apriori. The prior probability could be determined by a myriad of factors, and further exploration is left to future work. The prior probability of a goal, $P(G)$, is used in the following chapters to compute the goal's posterior probability, given the sequence of observations, O.

As with multi-agent planning, which required taking into consideration the dependencies between the different agents, so does multi-agent plan recognition require a similar consideration. Specifically, to recognize the goals and plans of multiple agents, the recognition system must take into consideration all manner of possible interaction and dependencies between agents, in addition to the characteristics of the domain and the given observations.

In the next section, we will enumerate a set of attributes which delineates the general case of the MAPR problem and make explicit the various assumptions made by previous work in the field. As mentioned, to relax a number of these assumptions, we will define two novel formulations of the MAPR problem in in Chapters 3 and 4. First, however, we define a basic formulation of the multi-agent plan recognition problem, which does not relax the aforementioned assumptions, as well as its solution.

Definition 5 *[**Multi-Agent Plan Recognition Problem**] A Multi-Agent Plan Recognition Problem problem is a tuple of the form*
$P^{MAPR} = (F, \{A_i\}_{i=1}^N, Ag, I, O, \mathcal{G}, \textsc{prob})$, *where* F, I, $\{A_i\}_{i=1}^N$ *and* Ag *are defined as in Definition 3,* $O = [o_1, ..., o_m]$, *where* $o_j \in \mathcal{A}$, $1 \leq j \leq m$, *is the sequence of observations,* $\mathcal{G}$ *is the set of possible goals,* $G \in \mathcal{G}$, *and* $\textsc{prob}$ *is a probability distribution over* $\mathcal{G}$, *specifying the prior probability of a goal,* $P(G)$.

Definition 6 *[**Solution to the Multi-Agent Plan Recognition Problem**] Given a Multi-Agent Plan Recognition problem,*
$P^{MAPR} = (F, \{A_i\}_{i=1}^N, Ag, I, O, \mathcal{G}, \textsc{prob})$, *the solution is given as the following probability distributions: i.* $P(G|O)$, *the probability of goals given observations, and ii.* $P(\pi|O)$ *the probability of plans given observations.*

Thus, given a multi-agent domain description and a sequence of observations, the task is to infer the goals and plans of the agents.

The solution to the MAPR problem may vary, depending on the properties of the MAPR formulation in question, as will be discussed in the following sections. For example, we note that some previous work in MAPR has, as part of the solution to the problem, identified the teams to which the different agents belong. A team was typically identified by this body of work as a group of agents working together towards a common goal. In this thesis, as will be seen in chapters 3 and 4, we assume that all agents share a higher-level common goal. In addition, we choose to focus in this work on various important aspects of the MAPR problem, including the relaxation of assumptions made by previous work. As such, exploring team identification is out of the scope of this thesis and is left to future work.

Further, in this work we assume that all agents are pursuing a common goal and so the goals $G \in \mathcal{G}$ in the formulations presented in the following chapters, are shared by all agents. Thus, as a solution to the MAPR problem, we assign probabilities to different goals the *group* of agents might be pursuing, rather than the goals of individual agents. However, as will be discussed in section 2.2.9, agents might have different, possibly conflicting goals, or they might be independently pursuing separate goals. Alternatively, we might want to recognize the goals pursued by the identified teams of agents or focus on a specific subset of agents and recognize their goals. In these cases, where there is not one common goal shared by all agents, $\mathcal{G}$ could represent the possible goals of each agent, each subset of agents, or whichever level of resolution is needed for the particular MAPR formulation.

2.2 Characterization of the MAPR Problem

Next, we would like to enumerate a set of attributes which delineates the general case of the MAPR problem. In [25], a comprehensive survey was performed, laying out and comparing different approaches and paradigms to autonomous agents modelling other agents. The authors performed a thorough comparison of previous research, within each modelling approach, based on the underlying assumptions made by the featured body of work. One of the surveyed modelling approaches the authors chose to focus on is Group Modelling, where the properties of multiple agents are modelled. Within this class of modelling approaches, the authors include the ever-growing body of work on MAPR and compare past research. We elect, in this thesis, to utilize the set of assumptions outlined by [25] and elaborate upon it, introducing additional important assumptions and attributes. The attributes we introduce in this section pertain mostly to the MAPR problem and are therefore interesting in the context of the characterization of the MAPR problem. We use the following assumptions from [25]: *Fixed or Changing Behavior*; *Common Goals?*; *Agent Interaction*; *Observations - Observability*. The other assumptions mentioned in this section were not adopted from the comparison done in [25].

In what follows, we list the set of assumptions and attributes which delineates the general case of the MAPR problem and proceed to present a comparison between the different MAPR approaches which were presented in section 2.3.1.

2.2.1 Fixed or changing behavior?

The question raised here is whether or not, and to what degree, an agent is able to change its decision making mechanism. A non-learning agent with fixed behavior is not able to adapt its behavior in light of its actions and the history of the environment. For example, in plan recognition approaches which utilize a plan-library, the behavior of the agents is assumed fixed and not adaptable. In contrast, a learning agent might try to constantly update its model of other agents and of the environment, as new information and evidence becomes available. For example, human agents constantly adapt to their environment and (usually) update their decision making mechanisms according to perceived changes to it. Fixed behavior is often assumed in order to alleviate the immense computational price of tracking and predicting possible changes to the agents' behavior.

2.2.2 Action Durations

Much previous work in MAPR assumed that actions have instant effects and chose to use joint instantaneous actions (or did not allow for concurrency at all) in order to model concurrency of actions of agents. However, in many real-world scenarios, the effects of the agents' actions are not instantaneous; for example, when passing a soccer ball to a fellow player, some amount of time will have elapsed before the ball reaches its destination.

2.2.3 Shared, isolated or conflicting goals?

In a multi-agent setting in general, and in the MAPR context in particular, it is important to consider the nature of the agents' goals. We define a shared goal as one that is shared by all agents. While, in a cooperative setting, different agents could adopt different sub-goals which accomplish the shared goal, the overarching goal is identical for all agents. In contrast, agents could be adversaries or in need of the same limited resources. In such cases, the goals of the agents might conflict. In this work we assume that one common goal is shared by all agents and therefore do not address conflicting goals. However, addressing the possibility of conflicting goals is important and is left to future work. Finally, agents may not be interacting directly and their goals might be isolated; for instance, when agents are independently interacting with one another, as is described in section 2.2.9.

2.2.4 Relationship between the observing agent and the observed agent(s)

Cohen et al. [26] categorize plan recognition methods into 'keyhole' and 'intended' methods: in keyhole plan recognition, the observed agent or agents are not aware of the observing agent, who is observing them as if through a keyhole. Furthermore, in a keyhole setting the agents do not attempt to assist or sabotage the observing agent's recognition process, since they are not aware of it. In contrast, in 'intended' plan recognition, the observed agent is aware of the observing agent and might wish to communicate its plan to it. Under the paradigm of intended plan recognition, the observer can interact with the agents and directly interfere with their plans, either assisting or undermining them. For example, Freedman and Zilberstein [27] integrated planning and plan recognition by allowing the observing agent to act on the recognized goals and plans of the observed agent. To this end, they define three forms of possible interaction types between the observing agent and the observed agent: (1) **Assistive Interaction**, where the observing agent adopts the observed agent's goal as its own and assists the agent in achieving it; (2) **Independent Interaction**, where the observing agent has its own goal, but avoids preventing the observed agent from achieving its own goal and (3) **Adversarial Interaction**, where the observing agent actively tries to prevent the observed agent from achieving its goal. For example, an AI player became a more challenging opponent after utilizing a recognition system which identified the human player's strategies and tactics [28]. As will be discussed in section 5.2, while previous work in MAPR has assumed keyhole plan recognition, relaxing this assumption is an important avenue of future work for the field.

2.2.5 Heterogeneous or Homogeneous group of agents?

In a multi-agent setting, it is important to consider the capabilites of the different agents. For example, in the RoboCup domain [29] each team is a homogeneous set of agents, with identical capabilities. In contrast, in a search & rescue environment we can expect to find a heterogeneous group of agents, comprised of both heavy robots capable of carrying injured people to safety, and lighter scout robots, tasked with locating victims. As discussed earlier, the heterogeneity of a group of agents can lead to different agents forming different perspectives on their environment.

2.2.6 Observability - Observations

Here, we are concerned with what the observing agent is able to observe about the environment and, importantly, about the observed agents. Much previous work in MAPR assumes perfect and complete observation sequences; however, in many real-world applications of MAPR, observations are unreliable, i.e., unexplainable or missing. In the case of unexplainable observations, available observations may not be explainable by the agents' goals and plans; these observations could be a result of a faulty sensor or communication error. In the case of missing observations, the recognition system is given an incomplete set of observations. Beyond unreliable observations, the degree of observability depends both on the environment (including the observed agents) and on the observing agent. The observing agent's perception determines what it is able to observe about the agents' actions; for example, if we do not assume omniscience on behalf of the observing agent, then it cannot observe the actions of agents outside of its line of sight (or outside the range of the deployed sensors). Further, some of the agents' actions might be private or parts of the environment might be concealed, with the observing agent not having access to them. One such example is a Poker game, where no player can see the cards held by her opponents; another such example is an action performed by an agent, which affects its internal state but is not communicated or shared, and has no external manifestations.

2.2.7 Variability in the agents' perspectives

While previous MAPR work attributed identical perspectives to the different agents, in many real-world scenarios the agents might have a different perspective on the environment, due to variability caused by the following factors:

Variability in the agents' perceptual capabilities: the agents may differ in their perception of the world, where a certain agent might, for example, have limited perceptual access to the environment, which affects the actions it is likely to take towards achieving its goal(s). For instance, a bat and a human navigate in drastically different ways, due to the biologically determined disparities of their respective perceptual capabilities.

Variability in the agents' mental state: in many real-world scenarios, different agents hold different motivations, intentions and beliefs regarding the collaborative or individual plan they are executing and participating in. Further, different agents might not possess the same representation of the environment and the plan, even if they are working together. For example, two search & rescue robots, working together in a disaster zone, might form different plans due to differing capabilities and communications received. One robot might decide not to explore a certain location, because it has not received a communication which the other robot has received. Additionally, different agents may prioritize different aspects of a common goal, where, for example, two agents might differ on the order in which to perform triage on two people in need of care.

In the simplest case, all agents have full access to the entire environment, with zero perceptual and mental state variability. Most previous work in MAPR either assumed such a setting, or did not explicitly address the implications of such variability in the plan recognition context. However, as will be demonstrated in Chapter 3, these assumptions are too strong as they do not allow us to capture important nuances of the recognition problem. To relax these assumptions, a localized perspective of each agent will be formed. This way, actions can be reasoned about in the context in which they were made.

2.2.8 Decentralized or Centralized Planning?

Since multi-agent plan recognition is done in the presence of multiple agents, it is important to make clear the assumptions made about the de-centralized or centralized manner in which the agents' plans are made. First, we distinguish between two types of agents: *planning agents* and *executing agents*. The former is involved in the planning process, while the latter is involved in plan execution.

In centralized multi-agent planning, one planning agent devises a plan which is executed by one or more executing agents. For example, in a factory setting there is likely a centralized planning system, acting as the planning agent, which plans for all the executing agents working in the factory. In decentralized planning, there are multiple planning agents, and decisions are typically made locally and autonomously by the multiple planning agents. For example, in a search and rescue environment, multiple agents (who could be either artificial or human) might navigate a complex environment in search of victims and with the objective of saving as many lives as possible. In such a setting, it is often impossible to have a central system plan for all agents, due to, for example, partial observability and sub-par communication; thus, the various agents must, autonomously and locally, deliberate and plan in order to achieve the goals of the group, given their own perspective of the environment.

Different approaches to MAPR employed different evaluation techniques; most previous work, in order to generate the ground truth plans and goals of the agents, assumed a centralized planning system which computes an optimal plan for a group of executing agents. This assumption is, once again, too strong for many real-world scenarios; in Chapter 3, in order to simulate the various scenarios and generate ground truth plans, we form a plan from each agent's unique perspective.

2.2.9 Agent Interaction Types

It is important to distinguish between different types of relationships and attitudes the agents might have with and towards one another. It is especially important for MAPR approaches to capture correlations between the agents' actions when the agents do not act independently of one another, as is often the case in real-world scenarios (e.g., coordination and cooperation between teammates or an adversary attempting to sabotage a fellow agent's plan). Following [30] and [31], we define three different types of possible agent interactions.

Independent Agents: The agents share a common environment, however, they have isolated goals and do not directly interact. Agents might still interfere with one another's plans, as the agents are sharing a common environment. For example, in the presence of shared resources, two independent agents might require the same resource, which might prevent one or both of the agents from achieving their goal.

Coordinated Agents: In contrast to independent agents, coordinated agents actively avoid interfering with one another's plans by communicating and coordinating. These agents communicate during the planning process in order to avoid interferences. Further, the agents prioritize the optimality of the joint plan over individual plans. There are different approaches to computing plans for coordinated agents, some involve the merging of individual plans, while others attempt to maximize the efficiency of a joint plan.

Cooperative Agents: Cooperative agents go a step further and may proactively adopt sub-goals which assist their teammates to achieve their goals. As a result of adopting sub-goals, planning with multiple goals may lead to interleaving actions, each one supporting a different goal. In the context of recognition systems, this adds another layer of complexity: since agents may be pursuing different goals, each observed action may be contributing to a different goal.

Adversarial Agents: Finally, adversarial agents intentionally work against one another, where each agent tries to prevent the other from accomplishing their goal. In game playing, agents are typically seen as adversarial.

2.3 Review of previous work

In this section, we survery previous work in MAPR and characterize this body of work as different configurations of the set of attributes enumerated in section 2.2.

2.3.1 Past Research in Multi-Agent Plan Recognition

Early work in MAPR 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. For example, using footage from American football games, Intille and Bobick [32] developed an activity recognition system for multi-agent systems. This system recognized plays in a library, based on the pose of a player, in addition to speed sensor data which tracked individual players. In their work, all agents are executing a single plan as one team and so the observations can be concatenated and matched against the library of plays.

Saria and Mahadevan [8] also focused on the detection of coordinated behavior in a sports domain. The approach presented in their work, based on Multi-agent Markov Decision Processes, enables the recognition of multi-agent plans in soccer games. Their approach uses a hierarchical policy structure which includes joint policies, compounded of concurrent single-agent policies.

Banerjee et al. [9] proposed to formalize MAPR with a new model: provided a fully observed team trace and a plan library of complete team plans, they used a first-cut approach to find the most likely team plan given the trace. In this work, the plan library is flattened, and the observations are matched against it; finally, various existing algorithms are extended in order to detect patters over all combinations of agents. The authors later extended their work to allow for interleaved plan execution and incomplete observation traces [33].

In [34], the authors introduce a template-based approach to multi-agent plan recognition. In their work, the recognition is facilitated by combining a top-down approach with a bottom-up approach. First, observations about the state of the system are used to reason about possible high-level naval military goals, and those goals are then decomposed into their component actions. Second, the individual agent actions are recognized and combined into plans; these plans are then used to determine the most likely overall high-level goal of the group.

Previous work has also leveraged complex coordinated agent behavior in the recognition process; Sadilek and Kautz [35], using GPS trackers attached to the players, tracked multiple agents in a game of Capture the Flag. Due to the limited accuracy of the GPS, tagging (the act of 'freezing' an enemy player in your territory by touching them) could not be recognized by detecting temporal co-location of two players. Rather, tagging was recognized when a tagged player remained stationary. This approach offers a possible solution for handling unobservable actions or dropped observations, by detecting the effects of the these missing observations

Further, coordinated agent actions were used by Sukthankar and Sycara [36] to prune the multi-agent plan library by matching observed structure within coordinated traces to structures within the library. Their framework utilized both temporal information and coordination information to constrain the plan recognition process.

Avrahami-Zilberbrand and Kaminka [37] chose to use a plan library comprised of single-agent plans, rather than one with team plans; in their work, they identified dynamic teams based on the assumption that all agents in a team execute the same joint plan under the temporal constraints of that plan. However, assuming all agents in a team follow the same plan puts a constraint on the activities of the agents in the team, and can be limiting when observing team-mates which execute coordinated but different behaviors. Their approach can also identify suspicious behavior from the observed information, by detecting deviations from an expected plan.

In [38], knowledge of the social structures of teams of agents is exploited in order to monitor teams of cooperating agents; their approach monitors the communication between the different members of the team, where the plans and goals are subsequently inferred based on the content of the communication. The authors' proposed approach can scale well to very large instances of the MAPR problem; however, in order to achieve this, their approach trades expressivity for scalability, electing to represent only certain useful monitoring hypotheses.

In [12], the agents' low level actions are first detected using Hidden Markov Models; then, the higher level intentions, possibly involving multiple agents, are detected using activation networks in which the activation spreads between nodes based on the detected low level actions of the agents. This way, both the intentions of a single agent, as well as joint intentions of a group of agents, can be recognized. The authors validated their approach within an open source naval ship simulator, where the system showed promising results when recognizing threatening intentions against friednly naval ships.

Zhuo et al. did away with plan libraries and instead developed an action model-based MAPR tool called DARE (**D**omain-model based multi-**A**gent **RE**cognition), which jointly inferred goals and team compositions [10]. Their approach casts the MAPR problem as a Constraint Satisfaction Problem (CSP), which it then solves using a MAX-SAT solver. Hard constraints are created by analyzing the action model, finding dependencies between actions; observations are also converted to hard constraints, forcing the solution to the CSP to satisfy both the domain specification and the order in which the observations were made. Earlier work by the authors, rather than using action specifications in the STRIPS formalism, used matrix-based plan libraries in conjunction with casting the MAPR problem as a satisfiability problem [39].

Finally, in [40], an approach to the MAPR problem is proposed which also eschews plan libraries and instead makes use of AI planning; they propose an extension to the Plan Recognition as Planning paradigm, proposed by Ramirez & Geffner [13], and transform the MAPR problem into a planning problem. Their approach includes an online recognizer, which updates the likelihoods assigned to goals and plans given the latest incoming observation. While this work is the most closely related to ours, it still makes a number of strong assumptions which are relaxed in this thesis. In their work, unexplainable observations are not addressed, actions are assumed to be instantaneous, and the agents are assumed to be sharing a common mental state and representation of the world.

2.3.2 Comparison of Previous MAPR Research

	Changing Behaviour?	Action Durations	Common Goals?	Observed-Observing Relationship	Observability - Observations	Agents' Mental State	Heterogeneous Agents?	(De)Centralized Planning	Agent Interaction
(Argenta and Doyle, 2017)	no	inst.	no	keyhole	partial	shared	yes	cent.	coor.
(Saffar et al., 2015)	no	inst.	no	keyhole	full	shared	yes	dec.	mixed
(Zhuo et al., 2012)	no	inst.	no	keyhole	partial	shared	yes	dec.	mixed
(Banerjee and Kraemer, 2011)	no	inst.	no	keyhole	partial	shared	yes	dec.	coor.
(Zhuo and Li, 2011)	no	inst.	no	keyhole	partial	shared	yes	dec.	mixed
(Banerjee et al., 2010)	no	inst.	no	keyhole	full	shared	yes	dec.	coor.
(Sadilek and Kautz, 2010)	no	inst.	yes	keyhole	partial	shared	no	dec.	coor.
(Avrahami-Zilberbrand and Kaminka, 2007)	no	inst.	_*	keyhole	partial	shared	no	dec.	ind.
(Sukthankar and Sycara, 2007)	no	inst.	no	keyhole	partial	shared	no	dec.	coor.
(Saria and Mahadevan, 2004)	no	inst.	yes	keyhole	partial	shared	yes	mixed**	coop.
(Kaminka et al., 2002)	no	inst.	yes	keyhole	partial	shared	yes	dec.	coop.
(Intille and Bobick, 1999)	no	inst.	yes	keyhole	partial	shared	yes	cent.	coor.
(Azarewicz et al., 1989)	no	inst.	yes	keyhole	partial	shared	yes	cent.	coor.

Table 2.1: Comparison of previous MAPR approaches with different configurations of the various attributes enumerated in section 2.2
*Goals are not recognized by approach/discussed in paper
**Centrally planned at a higher level and individually planned at lower levels

Table 2.1 presents a comparison of previous MAPR approaches, surveyed in the previous section, with different configurations of the various attributes and assumptions. As has been claimed, it is clear that previous work in MAPR has operated under strong assumptions, limiting the classes of problems to which they are applicable. Importantly, previous work has assumed that all agents have a shared mental state and thus did not consider the unique perspective of each agent when recognizing the plans and goals of multiple agents; additionally, previous work has assumed that actions are instantaneous and have no duration; further, as has been mentioned, MAPR research has not yet addressed a different relationship between the observing agent and the observed agents, other than keyhole recognition, in which the observed agents are not aware that they are being observed. In chapters 3 and 4, we will present novel formulations of the MAPR problem, which relax a number of these assumptions and allow us to solve new classes of problems. Specifically, in Chapter 3 we no longer assume that all agents must share a common mental state, thus enabling the observing agent to consider the unique perspective of each agent when recognizing its plans and goals. In Chapter 4, we relax the assumptions that (a) the agents actions are instantaneous and (b) the observations are perfect and reliable.

Chapter 3

Epistemic MAPR

In this chapter and the following chapter we will address *sub-problem #2*, *sub-problem #3*, and *sub-problem #4* by characterizing and solving novel formulations of the MAPR problem and by illustrating the power and flexibility of our approaches by addressing previously unaddressed classes of MAPR problems.

The structure of this chapter is as follows: first, we present a number of motivating examples; second, we define the main problem addressed in this chapter, the Epistemic MAPR problem, which relaxes the assumption, made by previous MAPR research, that agents share a common mental state; third, we show how to transform the epistemic MAPR problem to a classical planning problem; fourth, we propose an AI planning-based computational approach which allows us to utilize powerful planning tools in order to solve the epistemic MAPR problem; finally, we show how this approach can be used to solve an important and previously unaddressed class of MAPR problems, by illustrating the power and flexibility of our approach using a number of different scenarios.

3.1 Motivation

To motivate the need to characterize and solve this formulation of the MAPR problem, let us consider the Urban Search & Rescue (USAR) domain (e.g., [41] [42] [7]). The objective in USAR is the location, extrication, and initial medical stabilization (triage) of victims trapped in structural collapse due to natural disasters, mines and collapsed trenches. Much research has been aimed at investigating the role of artificial agents, mostly robotic, in such scenarios. The robots and humans form a team on-site and collaborate in order to better perform the task at hand. Often, risk to humans can be mitigated or even entirely avoided by deploying robotic rescuers. This setting introduces

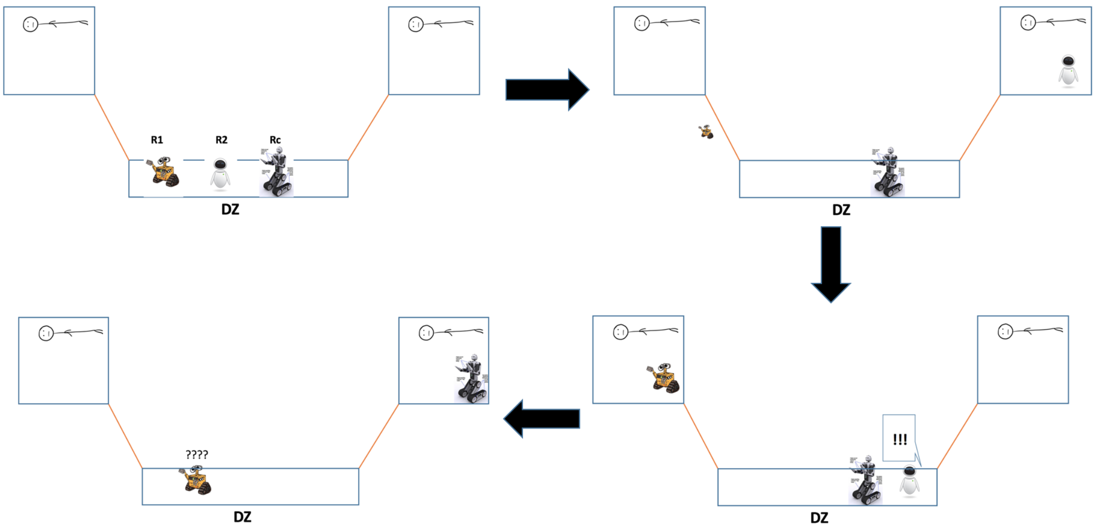

FIGURE 3.1: An illustration of the 2nd USAR scenario (described in section 3.1): in it, we have three robots - two small (R_1 and R_2) and one large (R_c); R_1 and R_2's shared objective is locating a victim (the locations of the victims are not known in advance) and transporting them to the Drop Zone (DZ).

great complexity, since there are usually multiple deployed robots on site, and they must collaborate and coordinate their actions in order to achieve their goals. To this end, much work has been done on team formation and management (e.g., [43]), agent communication (e.g., [44]) and AI planning (e.g., [45]) in the USAR domain. In the context of plan recognition, Talamadupula et al. [7] propose an approach which allows a robot in a USAR environment to predict a teammate's behavior given a high-level mental model of the observed agent and observations of its actions; they do so by using AI planning and by modelling the mental states of the observed teammate. We will see in section 3.6 how our approach can address the scenario proposed in their work.

We consider two motivating scenarios, situated in the USAR domain, which illustrate the importance of solving the MAPR problems they present. After describing these examples, we will discuss the desiderata of any approach that aims to address these problems.

Example 3.1 *[Motivating Scenario #1] Let us consider two small robots, R_1 and R_2, initially located at a drop zone (DZ), that set out to search for victims in a building, each headed towards a different room. The scenario is illustrated in Figure 3.1. The locations of the victims are not known in advance, and the general objective of the robots is: "Locate *a* victim and transport them to the DZ". In such scenarios, there is typically limited communication, however, we assume that the robots' movements or location can be tracked using sensors and other monitoring devices. Since R_1 and R_2 are small search robots, they cannot transport the victims they find, and must venture back to the DZ to* *alert a more capable robot, R_c, to the whereabouts of the victim they have found. Upon being alerted, R_c will leave the DZ to retrieve the located victim. Both agents believe that R_c is at the DZ and know to return to it in order to report what they have found. The issue arises, then, when R_1 finds a victim, heads back to the DZ and alerts R_c, causing R_c to leave the DZ before R_2 has reached it. Without further mechanisms in place, R_2 would be 'stuck' at the DZ, not knowing what to do and wasting precious time. The problem is caused since, while the agents are teammates and share a common high-level goal, i.e. transporting a victim to the DZ, their perspectives on how to achieve this goal may differ. In this case, since each agent locates a different victim, their grounded version of the common goal becomes "transport *the located* victim to the DZ". As a solution to this problem, R_c could 'leave a message' for R_2, explaining where it had gone; alternatively, R_2 could be alerted to the new situation or backup could be sent its way.*

Example 3.2 *[Motivating Scenario #2] In this scenario, the initial conditions, as well as the common goal of the agents, are identical to those presented in the first scenario. However, in this scenario two small robots can join forces and together lift and transport the victim. For this to happen, a small robot, upon finding a victim, must venture to find another small robot and ask for its assistance. Upon being alerted, the fellow small robot, if it is not otherwise occupied, will accompany its fellow teammate to the location of the victim, where, together, they will carry the victim and get them to the DZ. Both agents know where their teammate is headed and will therefore make their way there in order to seek assistance. The issue arises, then, when R_1 and R_2 both find a victim, and start making their way towards the other, in search of assistance. This, in turn, could cause the agents to 'get lost' within the potentially complex building layout, and fail to find their fellow teammate. Once again, as mentioned, while the agents are teammates and share a common high-level goal, their perspectives on how to achieve this goal may differ. As a solution to this problem, a recognition system could reason about the plans of each agent, from their individual perspective, and alert them to the change that had occurred in the other agent's location; alternatively, the recognition system could preemptively send backup or offer (if it is embedded in another robot) to physically assist the agents in lifting and transporting the located victims; the preemptive assistance is predicated on the recognition system's understanding of the agent's plans. Thus, both solutions to the problem require a plan recognition approach.*

The examples above can be cast as two separate problems, a multi-agent plan recognition problem, and a problem which requires intervention and assistance based on the recognized plans of the agents. In both examples, the recognition system must reason about the plans of each agent, from their individual perspective, and subsequently act upon the recognized plans. We note that in order to apply a solution to a real world scenario such as those mentioned above, a gap must be bridged between the recognition of the agents' plans, and the relevant intervention and assistance. We leave this integration for future work and discuss integration attempts made by previous work in section 5.2. In this thesis, we will focus on the plan recognition aspect of the problem, and cast it as a multi-agent plan recognition problem. To successfully address the MAPR problems which arise in both scenarios, let us consider what must be taken into consideration by a MAPR approach: i. a MAPR approach must be able to model the agents' beliefs and update them according to changes in the environment and the actions of the agents; ii. a MAPR approach must infer that both R_1 and R_2 will seek out R_c (or one another, in the second scenario) in order to achieve their goal; iii. given the observations about the state of the world and the actions or movements of the agents, each agent's plan, from their unique perspective, must be recognized, while taking into consideration the interaction between the agents and their held beliefs about one another.

As demonstrated in Chapter 2, all previous work in MAPR either assumes the agents possess a shared mental state, or do not explicitly reason about the beliefs of the agents and the changes to their beliefs. Thus, the requirements to solve the proposed scenarios, as elucidated above, cannot be met by previous work, due to the strong assumptions they make, as well as their lack of a sufficiently rich and expressive representational framework. In what follows, we propose a sufficiently rich and expressive formulation of the MAPR problem which relaxes a few key assumptions and allows us to address the above examples, as well as a new and important class of MAPR problems.

3.2 Preliminaries

3.2.1 Assumptions

In the previous chapter, we have characterized the MAPR problem by enumerating a set of attributes and assumptions which delineate the problem's general case. Before introducing the epistemic multi-agent plan recognition problem, we consider the assumptions made in this chapter, when addressing this variation of the MAPR problem.

Agents act rationally: agents choose to perform actions which lead to an expected optimal result, where optimality is defined relative to some objective metric. In this work, the objective metric is the cost of a plan, defined as the number of actions it contains; in Chapter 4, optimality will be determined by the durations of the actions.

Multiple planning agents: we do not assume a central planning system with multiple executing agents (as defined in section 2.2.8). In so doing, we allow for a de-centralized planning environment, where each agent is deliberative and involved in the planning process determining its actions. We note that this assumption does not preclude the possibility of a centralized planning system, with one planning agent and multiple executing agents.

Agents are not competing and share a common higher-level goal: we assume that agents are not adversarial and do not intentionally hinder other agents' plans. We further assume that all agents share a common goal (for example, locating a victim and transporting them to the drop zone). Importantly, in a de-centralized environment different agents might form their own perspective on how to achieve the common goal, and may pursue different plans to achieve the goal.

Agents do not share a common mental state and may have a unique perspective on the environment: in this chapter, we assume that each agent (be it human or artificial) has its own perspective on the world in which it operates. The potential dispartiy between the different agents' perspectives can be attributed to many factors, including: different agents being exposed to different information (e.g., different communications received or an encounter with a particular agent), resulting in disparate mental representations of the environment; agents with different perceptual capabilities (e.g., a human compared to a robot or a simple robot compared to an advanced robot) will perceive the world differently. We therefore allow for different agents to hold different, possibly contradicting, beliefs about the environment and the way in which to achieve the common goal, as will be seen and discussed in the following sections.

Heterogeneous group of agents: we offer, in this chapter and the next, an expressive framework in which to represent the heterogeneous nature of a group of agents. Different sets of capabilities, as well as unique mental states, can be associated with the different agents in the domain.

Nature of the observations available to the observing agent: in this chapter, we assume that the sequence of observations available to the observing agent consists of the actions of the observed agents, as well as public communications received by the agents.

3.3 Problem Definition

In this section we define the epistemic multi-agent plan recognition problem addressed in this chapter, and its solution. First, recall the multi-agent planning problem, as defined in Definition 3. While this formulation of the multi-agent planning problem is sufficient to solve various classes of problems, it implicitly assumes that all agents operate in their environment with identical knowledge and beliefs about it. In the general case, where agents have differing, possibly incomplete knowledge of the world, the problem is more challenging, as elucidated in the previous chapter and in Examples 3.1 and 3.2. Thus, we turn to discuss the knowledge and beliefs of agents and how to reason about the two. Often in the real world, there is a need to represent and reason about the mental state of an agent, including their beliefs and knowledge about their environment and about other agents. As humans, we do this type of representation and reasoning frequently and extremely well by employing our Theory of Mind (ToM) (e.g., [46], [47]). ToM allows humans to represent the mental states - including intentions, beliefs, desires, and knowledge - of others, via observations about their behavior and by utilizing an internal mental model. In this way, by constructing models of the mental state of others, we are able to better reason from the perspective of fellow humans, enabling us to e.g. empathize with and better understand them.

For humans, thus, expressions such as 'John believes that ϕ' or 'Jane knows that ψ' sound natural due to our ability to utilize ToM. Logicians and philosophers, as early as the 1950s (e.g., [48], [49]), realized that these expressions have systemic properties that can be formally studied. Hence, formal frameworks of epistemic (pertaining to knowledge) and doxastic (pertaining to belief) logics, were conceived. Hintikka [49] was the first to propose modelling belief and knowledge in terms of possible states of affairs, or possible worlds, and analyze the two with tools of modal logic. Under such a paradigm, the key notion of epistemic logic becomes that of an *indistinguishability* relation between possible worlds. Knowing a fact, ϕ, is then dependent on an agent's ability to *distinguish* between possible worlds where ϕ is true, and possible worlds where ϕ is false. The appeal of epistemic and doxastic logics was not lost on researchers from various fields, who extended and applied the original formal formulations; these included: computer scientists [50], game theorists [51] and cognitive scientists [52].

Next, it is often the case that an agent may wish to represent and reason about the mental state of several agents, including, importantly, modelling the nature of changes that occur in their mental states as a result of changes in their environment. Thus, Dynamic Epistemic (or Doxastic) Logic (DEL) was conceived to capture the effects of these changes. In other words, DEL *"studies the evolution of knowledge and belief in the context of change"* [53].

Formally, the set of well-formed formulae in DEL, $\mathcal{L}$, is defined inductively by the following grammar:

$$\phi ::= p \mid \phi \wedge \phi' \mid B_i\phi \mid [\alpha]\phi \mid \neg\phi$$

where $Ag = \{1, ..., n\}$, $\mathcal{A}$, and $\mathcal{P}$ are finite sets of agents, actions, and propositions, respectively. $p \in \mathcal{P}$ are atoms, $i \in Ag$ are agent indices, $[\alpha]$ is the action modality, and $B_i\phi$ should be interpreted as *"agent i believes ϕ."* The semantics is given by Kripke structures [54], which are triplets, $M = \langle W, R, V \rangle$, containing a set of worlds, accessibility relations between the worlds for each of the agents ($R = \{R_i \mid i \in Ag\}$), and a function, V, that maps from the worlds $w \in W$ into truth valuations $V(w)$ ($V : W \to 2^{\mathcal{P}}$), thus defining what propositions are true in each world. When an agent i is at world $w \in W$, M determines, given the accessibility relations in R_i that pertain to w, what worlds the agent considers possible. An arbitrary formula ϕ is true in a world w of a Kripke structure $M = \langle W, R, V \rangle$, written $M, w \vDash \phi$, under the following, inductively-defined, conditions:

- $M, w \vDash p$ for an atom p, if p is true in $V(w)$,
- $M, w \vDash \phi \wedge \psi$, if both $M, w \vDash \phi$ and $M, w \vDash \psi$,
- $M, w \vDash \neg\phi$, if $M, w \nvDash \phi$,
- $M, w \vDash B_i\phi$, if $M, w' \vDash \phi \quad \forall w' \in W$ s.t. $R_i(w, w')$, and
- $M, w \vDash [\alpha]\phi$, if ϕ holds after applying action α to (M,w)

An action, α, is defined as $\langle \pi, effects \rangle$, where π is the set of preconditions. The application of α to (M,w) is defined in terms of standard precondition and successor state axioms, following [55]. An action describes how the mental model of an agent, as well as the state of the world, change. Next, following the multi-agent epistemic planning formulation in [14], which we adopt later in this section, we assume certain constraints on the Kripke structure, which lead to particular properties of belief, as discussed in [54]. Namely, we assume that the Kripke structure is *serial* ($\forall w\, \exists v\, R(w\,, v)$), *transitive* ($R(w\,, v) \wedge R(v\,, u) \Rightarrow R(w\,, u)$) and *Euclidian* ($R(w\,, v) \wedge R(w\,, u) \Rightarrow R(v\,, u)$), with the resulting properties of belief:

$$K \quad B_i\phi \wedge B_i(\phi \Rightarrow \psi) \Rightarrow B_i\psi \qquad \text{(Distribution)}$$
$$D \quad B_i\phi \Rightarrow \neg B_i\neg\psi \qquad \text{(Consistency)}$$
$$4 \quad B_i\phi \Rightarrow B_iB_i\psi \qquad \text{(Positive Introspection)}$$
$$5 \quad \neg B_i\phi \Rightarrow B_i\neg B_i\psi \qquad \text{(Negative Introspection)}$$

These axioms, together, form the $KD45_n$ system, where n specifies that there are multiple agents in the environment.

Next, recall the limitations of the basic formulation of the multi-agent planning problem, namely the assumption that all agents operate with identical knowledge and beliefs about their environment. *Epistemic Planning*, to this end, marries AI planning and epistemic (or doxastic) logic, thereby leveraging the fortes of both worlds and enabling the relaxation of the aforementioned assumption. Epistemic planning has been the subject of ever-growing interest recently, culminating in a Dagstuhl workshop which brought together researchers from the fields of AI planning, epistemic logic, and knowledge representation & reasoning [56], to address fundamental problems in the field and to strengthen the marriage between the different fields of research. In general, epistemic planning can help answer the following question: *"Given some initial state of knowledge (or belief), and a desirable state of knowledge (or belief), how do I get from one to the other?."* That is, epistemic goals, as well as ontic ones, can be specified as the desired state to be achieved by the plan. This can be achieved by ontic actions (which affect the state of the world) and epistemic actions (which only affect the mental state of the agents regarding the non-changing state of the world). Importantly, by utilizing the formal framework of epistemic logic, epistemic planning is able to address a variation of multi-agent planning that can deal with non-determinism, partial observability, sensing actions, and, crucially, allow agents to reason about the knowledge, beliefs, uncertainty and capabilities of the other agents.

In the literature, different formulations of epistemic planning have been proposed, and especially of interest to us in the context of MAPR is the body of work on multi-agent epistemic planning, which explicitly addresses the strong assumptions made by multi-agent planning research in the past. For instance, Huang et al. [57] reason efficiently in the multi-agent KD45 axiomatic system by using alternating cover disjunctive formulas (ACDFs). This normal form induces formulas which are modified by belief revision and update algorithms that adapt the PrAO algorithm originally developed for contingent planning [58]. Le et al. [59] present two epistemic forward planners, EFP and PG-EFP, which make use of an epistemic planning graph in order to address unlimited nested beliefs, common knowledge, and are able to generate plans with both knowledge and belief goals.

Many proposed formulations have elected to use DEL as the underlying theoretical framework for multi-agent epistemic planning. As discussed, DEL addresses knowledge and belief in the context of change; in AI planning, actions drive change in the world with the objective of transforming some initial state into a desired goal state. DEL, thus, is a natural framework in which to model the effects of changes as affected by both the actions of the agents and the events occurring in the world. For instance, Kominis and Geffner [60] compile the multi-agent epistemic planning problem to a classical planning problem, thus allowing the computation of linear multi-agent plans; in their work, both arbitrarily long formulae, as well as disjunctive knowledge. Engesser et al. [61] introduce a new cooperative, decentralized planning concept that eliminates the need for the different agents to explicitly coordinate or negotiate. They do so by enabling implicit coordination between the agents, which relies on the agents' ability to take on the perspective of their fellow agents. Rather than compile the problem into classical planning, they introduce a planner that searches the space of epistemic states to find a solution.

In this chapter, we adopt the formulation of multi-agent epistemic planning as proposed by Muise et al. [14] (for the sake of convenience, we shall notate their formulation of multi-agent epistemic planning as MEP in the remainder of the thesis). We adpot their formulation for a number of reasons. First, their proposed formulation focuses on the beliefs of the agents, thus operating in a **doxastic** system (in contrast to an epistemic system of knowledge). This, importantly, allows us to model the false beliefs of agents, rather than assume that all agents start with common initial knowledge or beliefs, as is done in [60]. In MAPR, as mentioned, in order to understand the actions of the agents in the context in which they were performed, it is important to model agents possibly holding false beliefs, that might differ from those of the observing agent. Second, the MEP framework has been successfully applied to the team formation problem [62], in which an initiator agent must find potential team members, form a team, and formulate a joint plan which should be executed by the team and achieve some goal. The authors encode the team formation problem as a MEP problem and reason from the perspective of the initiator agent. Further, the authors demonstrate the benefits of casting the team formation problem as a planning problem, and make use of some of the powerful features offered by the MEP framework. Parallels can be drawn between the desiderata in the team formation work and ours, including: reasoning from the perspective of a 'special' agent (initiator vs. observing agent); need for a sound and complete compilation to a planning problem (which is adapted in [62] to accommodate non-determinism); need to explicitly represent and reason about the beliefs of multiple agents. Hence, the successful application of the MEP framework to the team formation problem inspired us to utilize it in the MAPR context. As part of future work, it will be interesting to explore the integration of our approach to MAPR with different formulations of multi-agent epistemic planning.

Finally, the multi-agent epistemic planning problem is defined as follows, with an underlying Kripke structure $M = \langle W, R, V \rangle$, the aforementioned semantics for Kripke structures and KD45 axioms:

Definition 7 *[**Multi-Agent Epistemic Planning Problem (following Muise et al. [14])**] A multi-agent epistemic planning (MEP) problem is a tuple of the form* $P^e = (\mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G})$*, where* $\mathcal{P}$*,* $\mathcal{A}$ *and* Ag *are defined as in the above definition of DEL,* $\mathcal{I}$

defines the initial theory, and $\mathcal{G}$ is the goal condition. Each action $a \in \mathcal{A}$ is assumed to be of the form $\langle \pi, \{(\gamma_1, l_1), ..., (\gamma_k, l_k)\}\rangle$, where π is called the precondition of a, γ_i is called the condition of a conditional effect, and l_i is called the effect of a conditional effect. Finally, we assume that $\mathcal{G}, \mathcal{I}, \pi, \gamma_i$, and l_i are all in $\mathcal{L}$, the set of well-formed formulae in DEL, excluding the $[\alpha]$ modality.

Definition 8 *[**Solution to the Multi-Agent Epistemic Planning Problem (following Muise et al. [14])**] Given a multi-agent epistemic planning problem $P^e = \langle \mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, \mathcal{G}\rangle$, the sequence of actions $a_1, ..., a_k$ achieves $\mathcal{G}$ if and only if for any (M, w) such that $M, w \vDash \mathcal{I}$, we have $M, w \vDash [a_1]...[a_k]\mathcal{G}$. The plan synthesis task, then, is one of finding a sequence of actions $a_1, ..., a_k$ that achieves the goal condition $\mathcal{G}$.*

Since reasoning in complex logical frameworks such as DEL is computationally expensive (e.g., [63], [64] and see discussion in section 3.6.1), the proposed formulation of multi-agent epistemic planning in [14] operates within a fragment of DEL. The authors, thus, introduce the following restrictions: i. reasoning is done from the perspective of a single *root agent* (defined in [14] as the agent from whose perspective the reasoning is done) ; ii. disjunctive formulae are not allowed as agent beliefs; and iii. a bounded depth of nested reasoning is defined. To reflect this, following the work of Lakemeyer and Lespérance [65], they define a *restricted modal literal* (RML) which is obtained from the following grammar:

$$\phi ::= p \mid B_i\phi \mid \neg\phi$$

The depth of a RML is defined as the maximum number of nestings of modal operators within it. In the MEP context, it is the maximum depth of nested belief modalities. Thus, $B_i\phi$, has a depth of 1 in addition to the depth of ϕ (which could contain more nested belief modalities). Formally, Muise et al. [14] define the depth of an RML as: $depth(p) = 0$ for $p \in \mathcal{P}$, $depth(\neg\phi) = depth(\phi)$ and $depth(B_i\phi) = 1 + depth(\phi)$. A conjunction of RMLs is viewed as a set, and the set of all RMLs with bounded depth d for a group of agents Ag is denoted as $\mathcal{L}^{Ag,d}_{RML}$. The state of the world, then, is some set of RMLs. Muise et al. [14] then define a *Restricted Perspectival Multi-agent Epistemic Planning problem* (RP-MEP problem) as a multi-agent epistemic planning problem, as defined in Definition 7, with a depth bound d and some root agent $\star \in Ag$. For future work, their hope (and ours) is to lift these restrictions, however, their proposed restricted framework offers an abundance of features, including, importantly, expressivity and a sound and complete transformation to classical planning. Note that in what follows, while we use the terms MEP and RP-MEP interchangeably, the formulation we are using is RP-MEP, where the observing agent is the root agent, and the depth, d, is

3 in the scope of our experiments. Such a depth is sufficient to address the scenarios presented in section 3.1 and could be increased as part of future work. Note that every RML is from the perspective of the root agent and so can either be $B_\star\phi$ or $\neg B_\star\phi$, where $\phi \in \mathcal{L}^{Ag,d}_{RML}$. Since $depth(B_\star\phi) \geq 1$, we allow for up to two more levels of nested belief. For example, a maximally nested RML in Example 3.1 could be $B_\star(B_{R_1}(B_{R_2}\phi))$, where the observing agent believes that R_1 believes that R_2 believes ϕ.

Finally, the perspectival variation of RP-MEP is appropriate in the plan recognition context, since, as mentioned, the recognition problem is reasoned about from the perspective of the observing agent. When we define the main problem addressed in this chapter, we implicitly cast the observing agent as the root agent in our MAPR formulation, and reason from its perspective.

We return now to the main focus of the thesis, the multi-agent plan recognition problem. Recall that in the basic formulation of MAPR, as defined in Definition 5, strong assumptions are made. These assumptions, include, importantly, the assumption that all agents share a common mental state. In order to address the motivating examples introduced in section 3.1, and take into consideration the unique perspective of each agent, this assumption must be relaxed. To relax this assumption, we define the Epistemic MAPR formulation, which leverages the multi-agent epistemic planning paradigm, discussed earlier. As previously mentioned, since multi-agent epistemic planning is rooted in epistemic logic, we are able to explicitly represent different agents having disparate mental representations of the environment. Importantly, in the MAPR context, reasoning about these disparate mental states can help the recognition system understand the agents' actions, i.e. the observations, in the context in which they were made.

Finally, we put everything together and define the problem we address in this chapter:

Definition 9 *[**Epistemic Multi-Agent Plan Recognition Problem**] An Epistemic Multi-Agent Plan Recognition Problem problem is a tuple of the form* $P = (\mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, O, \mathcal{G}, \text{PROB})$*, where* $\mathcal{P}$*,* $\mathcal{I}$*,* $\mathcal{A}$ *and* Ag *are defined as in Definition 7,* $O = [o_1, ..., o_m]$*, where* $o_j \in \mathcal{A}_{observable}$*,* $1 \leq j \leq m$*, is the sequence of observations,* $\mathcal{G}$ *is the set of possible goals,* $G \in \mathcal{G}$*, and* PROB *is a probability distribution over* $\mathcal{G}$*, specifying the prior probability of a goal,* $P(G)$*.*

Definition 10 *[**Solution to the Epistemic Multi-Agent Plan Recognition Problem**] Given an epistemic multi-agent plan recognition problem,* $P = (\mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, O, \mathcal{G}, \text{PROB})$*, the solution is given as the following probability distributions: i.* $P(G|O)$*, the probability of goals given observations, and ii.* $\{P(\pi_i|O) \mid i \in Ag\}$*, where* $P(\pi_i|O)$ *is the probability of plans given observations, when taking the perspective of agent* i*, as will be defined in section 3.5.*

$\mathcal{A}_{observable}$ is defined as the set of observable actions, $\mathcal{A}_{observable} \subseteq \mathcal{A}$. Since it is often not realistic to assume omniscience on behalf of the observing agent, it should therefore not be able to observe the actions of agents or events outside of its line of sight (or outside the range of the deployed sensors). Further, some of the agents' actions might be private (or purely epistemic, like a sensing action which only changes the mental state of the agent) or parts of the environment might be concealed, with the observing agent not having access to them (possibly including the internal state of the observed agents). In our experimentation, we include in $\mathcal{A}_{observable}$ public actions made by the agents, as well as public communications received by them (which are modelled as actions in our implementation). As part of future work, the notion of observability, especially in the epistemic realm, should be formalized and elaborated upon.

The set of possible goals, $\mathcal{G}$, can be seen as a set of alternative hypotheses for the common high-level goal shared by all agents (as mentioned in section 3.2.1), which are each assigned a posterior probability, given the observations. Note that each goal $G \in \mathcal{G}$ is a conjunction of predicates which must hold in the final state. In the most general case, and in the absence of prior knowledge, it is the set of all possible goals the agents might be pursuing. That is, $\mathcal{G}$ could contain all possible combinations of all possible grounded instances of the domain predicates. In some cases, however, we have at our disposal some domain knowledge and by using this knowledge, $\mathcal{G}$ can be populated only with alternative possible goals, given the domain and problem specifications. For instance, in the USAR domain, the set of possible goals, given the domain specifications, might resemble the following: $\{triagePerformedOnAllVictims, allVictimsLocated, medicalKitLocated\}$. In some instances, the high-level goal of the agents is known to us; for instance, in the motivating scenarios in section 3.1 (Examples 3.1 and 3.2), we know that the common goal shared by the agents is locating a victim and transporting them to the drop zone, represented by *victimAtDZ*. However, note that even when the goal is known, with $|\mathcal{G}| = 1$, there might be a set of possible plans that the agents can execute, which achieve the goal. For instance, a higher level shared goal of the agents might be *victimAtDZ*, however, since the goal is not fully specified (due to the locations of the victims not being known in advance), the agents may follow any one of a number of plans which ensure that *a* victim is located and transferred to the drop zone. For example, when R_1 locates $victim1$, it will choose to achieve *victimAtDZ* by satisfying *victim1AtDZ*, which in turn satisfies *victimAtDZ*.

To illustrate, we consider Example 3.1 from section 3.1; let us partially model the problem according to definition 9:

- $Ag = \{R_1, R_2, R_c\}$

- $O = [move\_R_1\_DZ\_Loc_1, move\_R_2\_DZ\_Loc_2, move\_R1\_Loc1\_DZ]$
- $\mathcal{G} = \{victimAtDZ\}$

According to the observations in O, R_1 and R_2 moved, respectively, from the DZ to Loc_1 and Loc_2; following this, R_1 started making its way back to the DZ. Recall that we would like R_c to infer that it should 'leave a message' for R_2, explaining where it had gone; alternatively, R_2 could be alerted to R_c's new location. The solution to the epistemic MAPR problem is given as the set of probability distributions $\{P(\pi_i|O) \mid i \in Ag\}$. By computing $P(\pi_{R_1}|O)$ and $P(\pi_{R_2}|O)$, the recognition system would be able to reason about the different ways in which R_1 and R_2 are planning to achieve the common higher-level goal, and provide the necessary information for a timely intervention.

In the next two sections, we will present an approach which transforms the epistemic MAPR problem into a classical planning problem and solves the transformed problem in order to compute the aforementioned probability distributions, $P(\pi_{R_1}|O)$ and $P(\pi_{R_2}|O)$.

3.4 Transformation to Classical Planning

In this section, we describe a two-step compilation technique, formalized in Algorithm 1, that allows the use of classical planning to solve the epistemic MAPR problem. That is, we first transform the given epistemic MAPR problem as defined in Definition 9 into a plan recognition problem, as defined in Definition 4; the first step is done using the compilation proposed in [14], which encodes the RP-MEP problem as a classical planning problem. Second, we transform the plan recognition problem into a classical planning problem, using the plan recognition as planning paradigm, proposed in [13]; finally, given the tranformed planning problem, we are able to compute the solution to the epistemic MAPR problem, namely the probability distributions of plans and goals given observations, in keeping with the previous plan-recognition-as-planning approaches; the planning-based computational approaches will be discussed in section 3.5. The compilation process is illustrated in Figure 3.2.

Algorithm 1: Transforming the Epistemic MAPR Problem to a Classical Planning Problem

Input: An Epistemic MAPR problem, $P = (\mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, O, \mathcal{G}, \textsc{prob})$, as defined in Definition 9
Output: A classical planning problem, $P' = (F', A', I', G')$, as defined in Definition 1
1: $P^r = \textsc{TransformToSingleAgentPRProblem}(P)$
2: $P' = \textsc{TransformToClassicalPlanningProblem}(P^r)$
3: RETURN P'

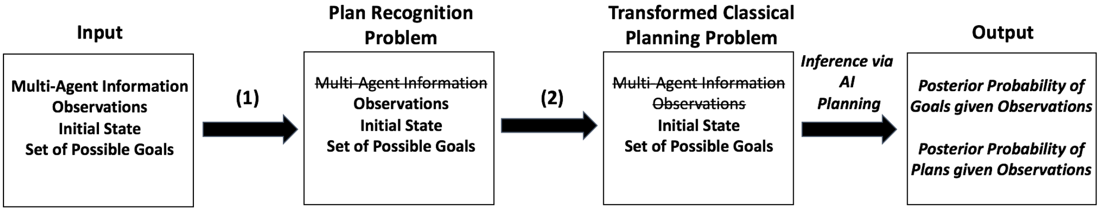

FIGURE 3.2: A pipeline showing our proposed compilation approach: transforming the original Epistemic MAPR problem into a plan recognition problem (1), a transformation step that compiles away the observations (2), allowing the use of planning tools to compute a solution to the MAPR problem (3).

3.4.1 Step 1 of Algorithm 1 - Transformation to a Single-agent Plan Recognition Problem

In this section, we describe the first step of Algorithm 1, where we transform an epistemic MAPR problem, $P = (\mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, O, \mathcal{G}, \textsc{prob})$, to a single-agent plan recognition problem, $P^r = (F, A, I, O, \mathcal{G}', \textsc{prob})$. To do so, we make use of the transformation proposed in [14], which encodes the RP-MEP problem as a classical planning problem. Following the encoding, every RML in $\mathcal{L}^{Ag,d}_{RML}$ will correspond to a fluent, $f \in F$, in the classical planning domain. The operators in the planning domain will describe how both ontic and epistemic changes occur in the world. The encoding, adapted to the epistemic MAPR formulation, is formally defined in [14] as follows:

Classical Encoding of the RP-MEP Problem (following Muise et al. [14])

Let $\mathcal{B}_i$ and $\mathcal{N}_i$ be functions that map agent i's positive and, respectively, negative beliefs from a set of RMLs, or Knowledge Base (KB), to the respective fluents in the classical planning domain:

- $\mathcal{B}_i(\text{KB}) = \{l_\phi \mid B_i\phi \in KB\}$
- $\mathcal{N}_i(\text{KB}) = \{l_\phi \mid \neg B_i\phi \in KB\}$

Given an epistemic MAPR problem, $P^e = (\mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, O, \mathcal{G}, \textsc{prob})$, we define the result of the transformation as the tuple $(F, A, I, O, \mathcal{G}', \textsc{prob})$ such that:

- $F \stackrel{def}{=} \{l_\phi \mid \phi \in \mathcal{L}^{Ag,d}_{RML}\}$
- $I \stackrel{def}{=} \mathcal{B}_\star(\mathcal{I})$

- $\mathcal{G}' \stackrel{def}{=} \{\mathcal{B}_\star(G) \mid G \in \mathcal{G}\}$

and for every action $\langle \pi, effects \rangle$ in $\mathcal{A}$, where $effects = \{(\gamma_1, l_1), ..., (\gamma_k, l_k)\}$, we have a corresponding operator $\langle Pre_a, eff_a^+, eff_a^- \rangle$ in A such that:

- $Pre_a \stackrel{def}{=} \mathcal{B}_\star(\pi)$
- $eff_a^+ \stackrel{def}{=} \{(\langle \mathcal{B}_\star(\gamma_i), \mathcal{N}_\star(\gamma_i) \rangle \rightarrow l_\phi) \mid (\gamma_i, \mathcal{B}_\star \phi) \in effects\}$
- $eff_a^- \stackrel{def}{=} \{(\langle \mathcal{B}_\star(\gamma_i), \mathcal{N}_\star(\gamma_i) \rangle \rightarrow l_\phi) \mid (\gamma_i, \neg\mathcal{B}_\star \phi) \in effects\}$

We adapt this encoding to the epistemic MAPR problem by addressing the set of possible goals, $\mathcal{G}$. While $\mathcal{G}$ in Definition 7 is the goal condition, i.e. a set of RMLs, in the MAPR context it is a set of sets of RMLs. Note that after step 1 of Algorithm 1 we are left with a single-agent plan recognition problem, $P^r = (F, A, I, O, \mathcal{G}', \textsc{prob})$, which includes the sequence of observations, O, and the set of possible goals $\mathcal{G}'$.

3.4.2 Step 2 of Algorithm 1 - Transformation to a Classical Planning Problem

In this section, we describe the second step of Algorithm 1, where we transform a single-agent plan recognition problem, $P^r = (F, A, I, O, \mathcal{G}', \textsc{prob})$, to a classical planning problem, $P' = (F', A', I', G')$. To do so, we compile away the sequence of observations, O, and incorporate it into the classical planning domain. We compile away the observations by utilizing the Plan Recognition as Planning (PRAP) paradigm, as proposed by Ramírez and Geffner [13], which views plan recognition as an inverse planning problem. While in a planning problem, as described in Definitions 1 and 7, the goal (or goals) of the agent(s) are known to us, in plan recognition they are not and the planning task becomes the satisfaction of the observations available to the recognition system. Importantly, in contrast to most plan recognition approaches, the PRAP approach does not require plan libraries, which are often difficult to specify and may be incomplete due to the observed agent going 'off-script' and using a plan which is not specified by the library; instead, PRAP uses planning domain models and computes plans which are then post-processed to correctly recognize the plans and goals of an agent. PRAP preserves the order of observations by forcing the planner to only generate plans which satisfy the observations in the correct order. This is achieved by constraining the plan generation process and forcing the planner to only compute plans which align with the given observations. Lastly and importantly, the PRAP approach assumes rationality on behalf of the agents, in that they are assumed to be executing optimal (or close to optimal) plans with respect to a known domain model and objective evaluation metric (e.g., the overall duration or cost of a plan). The approach has been extended since its initial introduction to accommodate sub-optimal observation sequences [15], varying degrees of observability [66], incomplete domain models [67] and other variations of the plan recognition problem.

We first define the satisfaction of an observation sequence, O, by a plan, π, following [13]:

Definition 11 *[Satisfaction of a Sequence of Observations (following Ramírez and Geffner [13])] A plan, $\pi = a_1, ..., a_n$, satisfies the observation sequence $O = o_1, ..., o_m$, as is defined in Definition 4, if there is a monotonic function f mapping the observation indices $j = 1, ..., m$ into action indices $i = 1, ..., n$ such that $a_{f(j)} = o_j$.*

Next, in order to solve the transformed planning problem using off-the-shelf planners, the observations must be compiled away; this is done by mapping the plan recognition problem $P^r = (F, A, I, O, \mathcal{G}', \text{PROB})$, which we are left with after the first step of Algorithm 1, to $P' = (F', A', I, O', G')$. Ramírez and Geffner [68] define a correspondence between the plan recognition problem and the planning problem, which compiles away the observations; Sohrabi et al. [15] extend this correspondence to allow for the use of diverse and Top-k planners, by introducing an extra fluent *done* to the goal condition of the transformed planning problem. The fluent *done* ensures that at least one of the goals $G \in \mathcal{G}$ is achieved, which stands in contrast to the original compilation in [68], that generates a different planning problem for each goal $G \in \mathcal{G}$. Thus, inspired by both Ramírez and Geffner [68] and Sohrabi et al. [15], we define the correspondence between P^r and P' as follows:

- $F' = F \cup \{p_a \mid a \in O\} \cup \{done\}$,
- $A' = A \cup A_{goal}$
  - $A_{goal} = \{a_G \mid G \in \mathcal{G}'\}$
- $G' = done \wedge p_{last}$,
- O' is empty.

A' includes all original actions from A, as well as A_{goal}. The precondition of each action $a_G \in A_{goal}$ is the respective goal G (recall that G is a conjunction of fluents) and its effect is the newly introduced fluent *done*. By adding *done* to the goal condition G', we ensure that all plans that achieve G' satisfy at least of one of goals $G \in \mathcal{G}'$. This is necessary for our Top-k planning based approach, which will be presented in the next section. We also modify the actions in A' that appear in the observation sequence O. To each of these actions, o_a, we add an extra fluent p_a to $eff^+_{o_a}$ and add p_b to the precondition Pre_{o_a}; p_b corresponds to the action b that immediately precedes a in O. If a is the first observation in O, we leave Pre_{o_a} unchanged. Finally, p_{last} is added to $eff^+_{o_a}$ if o_a is the last observation in O.

Importantly, we ensure that the order of the observation sequence O is respected by all plans that solve the transformed planning problem, P'. The order is enforced by the precondition p_b which only allows an action a, which appears in O, to be executed after all the observations in the sequence O which precede it have been executed. Thus, the plans that achieve the modified goal G' are only those that achieve some goal $G \in \mathcal{G}'$ (due to the *done* fluent, see explanation above) and *satisfy* the observation sequence O (as is defined in Definition 11). Since the modified goal, G', can only be achieved by making p_{last} hold, and since p_{last} can only hold after the last action in O has been executed, we ensure that all actions in O are accounted for in the correct order.

Following the two steps of Algorithm 1, we have a transformed classical planning problem, $P' = (F', A', I', G')$. Next, we prove that our compilation process is sound and complete, thus proving that Algorithm 1 is solution preserving.

Theorem 1 *[Soundness and Completeness] Given an Epistemic Multi-Agent Plan Recognition problem, $P = (\mathcal{P}, \mathcal{A}, Ag, \mathcal{I}, O, \mathcal{G}, \text{PROB})$, and the corresponding classical planning problem, $P' = (F', A', I', G')$, which is the result of applying Algorithm 1 to P, we have the following:*

(1) For all $G \in \mathcal{G}$, if π is a sequence of actions that achieves the goal condition G and satisfies O in P (where $\pi = a_1, ..., a_k$ and $M, w \vDash [a_1]...[a_k]G$) then there exists a plan π' that achieves G' in P' where π can be constructed straightforwardly from π' by removing the extra goal action, $a_g \in A_{goal}$, from π'.

(2) If π' is a plan that achieves G' in P', then there exists a sequence of actions π that achieves some goal condition $G \in \mathcal{G}$ and satisfies O in P (where $\pi = a_1, ..., a_k$ and $M, w \vDash [a_1]...[a_k]G$), where π can be constructed straightforwardly from π' by removing the extra goal actions, $a \in A_{goal}$, from π'.

Proof:

(1)

Let G be a goal in $\mathcal{G}$. Assume that π is a sequence of actions that achieves the goal condition G and satisfies O in P. We prove that there exists a plan π' that achieves G' in P' where π can be constructed from π'. First, in [14], it is shown that the encoding of the MEP-RP problem as a classical planning problem, which we use here, is sound and complete. Specifically, a plan π_G will be found for a goal G from an initial state I, using the aforementioned encoding, if and only if $M, w \vDash \mathcal{I}$ implies $M, w \vDash [\vec{a}]G$ for any (M, w), where M satisfies $KD45_n$ and $\vec{a}$ is the action sequence corresponding to π_G. Thus, we know that π is a plan for P' with the goal G. Next, we know that the goal G is achievable since π achieves it (if G is not achievable then (1) holds trivially). Thus, one possible solution for the planning problem P', is a plan, π', that achieves G and satisfies the observation sequence, O. Since *done* holds in π' and G is achieved, we know that the action $a_G \in A_{goal}$, corresponding to G, is included in π'. Further, we know that any plan that solves P' satisfies the observation sequence, O, such that there is a monotonic function f mapping the observation indices $j = 1, ..., m$ into action indices $i = 1, ..., n$ such that $a_{f(j)} = o_j$. Note that π' achieves G' since it is a solution for P'. Finally, in order to construct π from π', we remove the goal action a_G from π'.

(2)

Assume that π' is a plan that achieves G' in P'. We prove that there exists a sequence of actions π that achieves some goal condition $G \in \mathcal{G}$ (*) and satisfies O (**) in P. Since π' achieves G', we know that *done* holds. Therefore, following the definitions in section 3.4.2, it follows that at least one of the goals $G \in \mathcal{G}$ is satisfied and that some action $a_G \in A_{goal}$ is included in π'. Hence, we construct π from π' by removing the extra goal action $a_G \in \pi'$. After removing a_G, we know that the goal G is still satisfied in the final state, after the execution of π (since a_G has no effect on the state of the world), thereby proving (*). Next, since π' achieves G' we also know that p_{last} holds after the last action in π' is executed (possibly before, but it is guaranteed to hold after). The second step of the compilation is designed such that p_{last} holds if and only if the observation sequence O has been satisfied and its order respected. Thus, π also satisfies O, thereby proving (**). Finally, we use the soundness and completeness of the MEP encoding to show that since π is a plan for P and goal G, $M, w \vDash [\vec{a}]G$ for any (M, w). $\vec{a}$ is the sequence of actions which corresponds to π. $\square$

Theorem 2 *[Algorithm 1 is solution preserving]*

Proof: This follows directly from Theorem 1, where we have shown that no solution to the input, P, is lost after applying Algorithm 1 to it. $\square$

We have shown, then, that our compilation of the epistemic multi-agent plan recognition problem to a classical planning problem is sound and complete, and that plans that solve the transformed planning problem correspond to plans that solve the various planning problems within the original MAPR problem. Using this correspondence, we will see in the next section how the costs of the plans that solve the transformed planning problem are used to approximate $P(O|\pi)P(\pi|G)$. This approximation, in turn, will allow us to compute the probability distributions, $P(G|O)$ and $\{P(\pi_i|O) \mid i \in Ag\}$, which are the solutions to the original epistemic MAPR problem, as defined in Definition 9.

3.5 Computation

In this section, we describe how to compute a solution to the epsitemic MAPR problem, as introduced in Definition 9, namely the probability distribution of goals given observations, $P(G|O)$, and the probability of plans given observations, $P(\pi|O)$. We follow the approach in [15], which uses a Top-k planner to compute the probability distribution of goals and plans. This is done by instructing the planner to find a set of plans, which serves as a representative approximation of the distribution of plans that satisfy the observations and achieve one of the possible goals in $\mathcal{G}$. Finally, we show how to compute the probability distribution of plans given observations from the perspective of an agent i, $P(\pi_i|O)$. We do so by projecting to reason as agent i, following the work in [14].

3.5.1 Computing $P(\pi|O)$ and $P(G|O)$ using Top-k Planning

Given the transformed classical planning problem we compute an approximation to the probability distribution of plans as well as goals, given the observations, by running a top-k planner on the transformed planning problem. We first define $P(\pi|O)$, the posterior probability of a plan given observations:

$$
\begin{aligned}
P(\pi|O) &= \beta P(O|\pi)P(\pi) \\
&= \beta P(O|\pi) \sum_{G \in \mathcal{G}} P(\pi|G)P(G) \\
&= \beta P(O|\pi)P(\pi|G)P(G)
\end{aligned}
\tag{3.1}
$$

where β is a normalizing constant that depends on $P(O)$ alone, and $P(G)$ is $\textsc{prob}(G)$, the goal prior. Note, we assume that only one goal $G \in \mathcal{G}$ is being pursued and $P(\pi|G)$ is 0 for the action sequences π that are not plans for G. Following [15], we approximate the value of $P(O|\pi) \cdot P(\pi|G)$ by taking into account the cost of other plans π' that satisfy O and achieve G, as well the cost of the plan π. Recall that all plans that solve the transformed planning problem, P', satisfy the obsrvation sequence and achieve some goal $G \in \mathcal{G}$. $P(O|\pi) \cdot P(\pi|G)$ is approximated as follows:

$$P(O|\pi) \cdot P(\pi|G) \approx 1 - \frac{\beta' \, C(\pi)}{\sum_{\pi' \in \Pi} C(\pi')} \tag{3.2}$$

where β' is a positive constant which can offset large sums if necessary. Π is the set of plans that satisfy the observations and achieve at least one of the goals $G \in \mathcal{G}$. Note that Π can be quite large since there is potentially a very large space of plans that achieve some $G \in \mathcal{G}$ and satisfy O. Therefore, we will see in the following section how to approximate the probability distribution over these plans, by sampling a set of plans from this space. $C(\pi)$ is the cost of a plan π that solves the transformed planning problem. For example, if there are 4 plans in Π, each with the same cost, then equation 3.2 will result in 0.75 for each of the four plans. Once we normalize the probability distribution $P(\pi|O)$ using β, each plan will be assigned a posterior probability of 0.25, given the observations. On the other hand, if a plan π_1 has a higher cost than another plan π_2, then we get that $P(\pi_1|O) < P(\pi_2|O)$. Thus, the approximation will assign higher posterior probabilities to plans with lower costs. This is desirable since we are comparing plans which satisfy the observation sequence; we would like to assign a higher likelihood to plans with the least cost increase when satisfying O.

Next, we would like to compute the posterior probability of goals given observations $P(G|O)$. Using Bayes rule, we have:

$$P(G|O) = \beta P(O|G)P(G) \tag{3.3}$$

where β is, once again, a normalizing constant that depends on $P(O)$ alone, and $P(G)$ is $\textsc{prob}(G)$. Next, assuming that the observations are independent of the goal G given π, and π ranges over Π, the set of plans that achieve G and satisfy O, $P(O|G)$ can be written as:

$$P(O|G) = \sum_{\pi \in \Pi} P(O|\pi) \cdot P(\pi|G) \tag{3.4}$$

Finally, recalling that $P(\pi|O) = \beta P(O|\pi)P(\pi|G)P(G)$, the probability distribution of goals given observations can then be computed by a summation over all values of $P(\pi|O)$ for the set of plans, Π, that achieve G and satisfy O, and a subsequent normalization of the summation values:

$$P(G|O) = \sum_{\pi \in \Pi} P(\pi|O) \tag{3.5}$$

The assumptions underlying this approach are as follows: i. when an agent is pursuing a goal G, it is more likely to follow cheaper plans than more expensive ones (this also follows from our assumption of agent rationality in section 3.2.1); ii. the observations are independent of the goal G, given π; iii. an agent only pursues one goal $G \in \mathcal{G}$ at a given time.

Next, in order to compute $P(\pi|O)$ and $P(G|O)$, we must compute the set of plans which satisfy O and achieve a goal $G \in (G)$, Π. Note that there is a large space of plans that achieve some $G \in \mathcal{G}$ and satisfy O and computing all of them is not practical; planning paradigms which compute a large set of plans, such as Top-k planning and diverse planning in the next chapter, are used as a means to approximate the probability distribution over these plans, by sampling a set of plans from this space. In this chapter, Π is computed using Top-k planning, which, following [69] and [15], is defined thus:

Definition 12 *[Top-k Planning Problem] The top-k planning problem is a tuple* (P, k)*, where* P *is a classical 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 $k \leq n$, and $m = n$ 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 exist a plan π' for P, $\pi' \notin \Pi$, and a plan $\pi_i \in \Pi$ such that $\mathrm{COST}(\pi') < \mathrm{COST}(\pi)$. 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.

Finally, the set of sampled plans, Π, is found by instructing the Top-k planner to find k plans for the transformed planning problem that satisfy the observation sequence and achieve at least one of the goals $G \in \mathcal{G}$.

3.5.2 Computing $P(\pi_i|O)$

In order to compute the probability distribution of plans given observations from the perspective of agent i, we turn, once again, to the work in [14]. The MEP theoretical framework allows the root agent to project to reason as other agents and we utilize this powerful feature of their approach in order to allow the observing agent, who is implicitly cast as the root agent, to adopt the perspective of the observed agents.

Intuitively, when projecting as another agent, the root agent assumes that agent's perspective of the world and so the beliefs of the projected agent become truths. More formally, the set of RMLs in the RP-MEP problem is repeatedly filtered according to the appropriate agent where the belief modality is stripped from the front of the RMLs. For example, prior to projecting as R_1 in Example 3.1, a possible RML could be B_{R_1}(connected l1 l2), meaning that R_1 believes that location 1 and location 2 are connected and the path between them is traversable. When projecting as R_1, the belief modality is stripped and we are left with (connected l1 l2). In addition to projecting the initial state of the world in this way, the effects of every operator are also projected in this manner. Note that since different agents can have disparate mental states, the root agent might hold a belief which is inconsistent with the beliefs of the projected agent. For instance, whereas R_1 believes that the path between location 1 and location 2 is traversable, the root agent knows that this is not true. Hence, the fluent ($\neg$(connected l1 l2)) will hold in the initial state. Since we want to assume the perspective of another agent, we must assume their beliefs, and allow them to supersede the, possibly contradicting, knowledge or beliefs of the observing agent. As part of future work, it will be important and interesting to explore the role of Model Reconciliation in the context of MAPR. For example, in [70] the authors enable an artificial agent to model its human teammates' beliefs and, if a false belief is detected, attempt to reconcile between the disparate mental models of the environment.

Formally, given a state s, the agent projection of s with respect to agent i, denoted as $Proj(s, i)$, is formally defined in [14] as:

- $Proj(s, i) = \{\phi \mid B_i\phi \in s\}$

Importantly, we are able to use this definition of projection thanks to the perspectival aspect of the MEP-RP framework, where s is seen from the perspective of the observing agent, $\star$, with $B_\star$ implicitly preceding all RMLs. It should be noted that the projection proposed in [14] can be generalized to a *vector of agents* (the *virtual agent* in [14]), which we would like to reason as. For example, given the vector of agents [Sue, Bob], we would like to reason as Sue reasoning as Bob. For the purposes of this thesis, we will only use a vector of size one, assuming the perspective of a single agent. However, for future work it will be interesting to explore even deeper nested reasoning when recognizing agents' goals and plans.

Finally, after modifying the planning problem according to agent i, we apply our approach to compute $P(\pi_i|O)$, which is equivalent to $P(\pi|O)$ in the projected planning problem, using equation 3.1 and the Top-k planning-based approach described in the previous section.

3.6 Experimental evaluation

In this section, we illustrate the power and flexibility of our proposed computational approach; to this end, we first demonstrate how our proposed approach, introduced in the previous section, can successfully address and solve the scenarios illustrated by the motivating example in section 3.1; second, we demonstrate that our approach can successfully address an Urban Search & Rescue scenario, as introduced in [7]; finally, we analyze the complexity of our approach, as well as its scalability.

As input to our recognition system, we format each problem instance as an epistemic MAPR problem, as defined in Definition 9 (a concrete example of how a problem instance is represented can be found below the definition). In order to solve each MAPR problem instance and compute the probability distribution over the agents' plans, we compile the MAPR problem into a classical planning problem, using the compilation process described in section 3.4. To find the sampled set of plans, Π, we used the TK* planner [71], with $k = 10$ and an admissible heuristic. Different values of k (up to 100) did not impact which plan was deemed most likely. As part of future work, with more complex domains and problem instances, the number of sampled plans will likely have a greater impact and experimentation will have to determine k's value. Note, the results were obtained by running TK* once for each problem instance. We ran our experiments on a 3.30GHz GHz Intel(R) Xeon(R) CPU E3-1230 processor with 256 GB RAM.

We created a set of problem instances for scenarios 1 and 2 by varying the parameters of the problem, as will be explained for each scenario below. Next, we evaluated our approach by the percentage of problem instances in which it correctly identified the ground truth plan of each agent $i \in Ag$, i.e. assigned to it the highest posterior probability, $P(\pi_i|O)$. To this end, we created the ground truth plan of each agent by doing the following: i. encoding each problem instance as a multi-agent epistemic planning problem with each problem instance having a shared high-level goal state (e.g., transporting a victim to the drop zone); ii. projecting each scenario's respective MEP problem as each of the agents; iii. encoding each problem instance as a classical planning problem, following the encoding in [14]; and finally iv. solving the classical planning problem using an optimal classical planner [72] with an admissible heuristic. The ground truth plan of each agent is then the output of the classical planner. We wished to test our approach's ability to, in the future, intervene and assist the agents; to this end, the sequence of observations, O, was created by selecting the actions from the ground truth plan that precede the so-called intervention point, as determined by each of the scenarios. For example, in scenario 1 we take the actions, from each of the agents' plans, which are executed before the agents go looking for one another. $\mathcal{G}$ in each problem instance contained a single goal - the aforementioned higher-level common goal.

Scenario #1 (Example 3.1) - Transporting a Victim to the Drop Zone by Seeking Out a Fellow Small Robot

As a reminder to the reader, in order to address the problem which arises in Example 3.1, as illustrated in section 3.1, the recognition system must infer that both R_1 and R_2 will seek out one another's assistance in transporting the victim to the drop zone, given the capabilities of each robot and their respective mental state (including their beliefs regarding other agents). In order to intervene before the agents 'get lost', each agent's plan must be inferred; to this end, we compute $P(\pi_{R_1}|O)$ and $P(\pi_{R_2}|O)$ to determine the most likely plan of each agent, given the observations.

As mentioned above, we vary the parameters of the scenario in order to create a larger set of problem instances. We vary the number of rooms (2 vs. 4 vs. 8 vs. 16); the distances between the different rooms and the drop zone (either R_1's path to the drop zone is shorter, or R_2's path is shorter); the locations of the two victims (for example, in instances with four rooms, there are 4 different options for the locations of the two victims: i. both victims are in the farthermost (relative to the drop zone) rooms; ii. both victims are in the closer rooms; iii. + iv. one victim is in the closer room, while the other victim is in the farthermost, and the converse).

We found that *our approach correctly identified the ground truth plan of each agent in all cases*. For example, the plan that was deemed most likely for R_1 was that it would go and look for R_2, with R_2 staying in place, which was indeed R_1's ground truth plan. Similarly, R_2's ground truth plan was recognized, which included R_2 seeking R_1, with the latter staying put.

Scenario #2 (Example 3.2) - Transporting a Victim to the Drop Zone by Seeking Out a Capable Robot

In Example 3.2, as illustrated in section 3.1, the recognition system must infer that both R_1 and R_2 will seek out R_c's assistance in carrying the located victim to the drop zone, given the capabilities of each robot and their respective mental state. In order to intervene, each agent's plan must be inferred; to this end, we compute $P(\pi_{R_1}|O)$ and $P(\pi_{R_2}|O)$ to determine the most likely plan, given the observations. We vary the parameters of the scenario similarly to what was done above in Example 3.1.

We found that *our approach correctly identified the ground truth plan of each agent in all cases*. For example, the plan that was deemed most likely for R_1 was that R_1 would seek R_c, which was indeed R_1's ground truth plan; similary for R_2.

Scenario #3 - Retrieving a Medical Kit - From Talamadupula et al. [7]

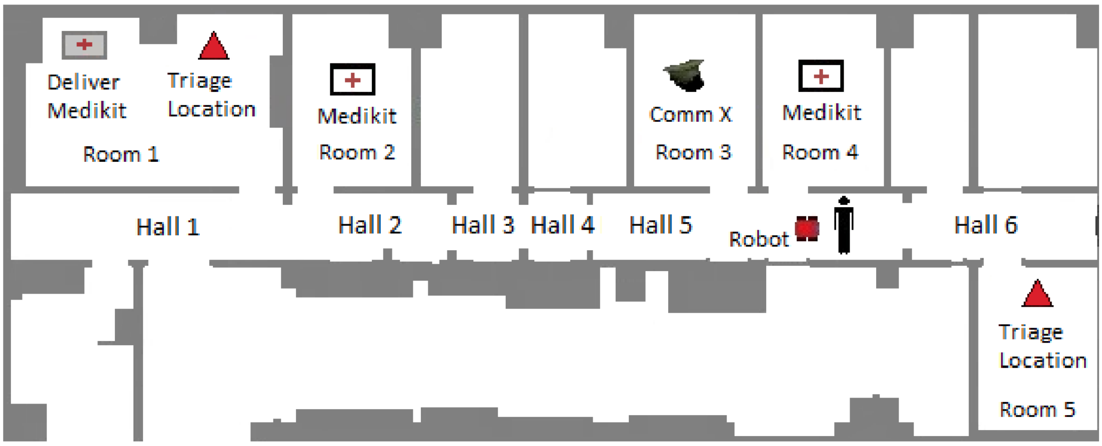

FIGURE 3.3: An illustration of the USAR scenario, as appears in [7]

In [7], the authors demonstrate how plan recognition, belief modeling and automated planning can be integrated in order to achieve coordination among different agents in a human-robot teaming scenario, set in the USAR domain. In their paper, the authors propose and successfully address a USAR scenario, illustrated in Figure 3.3. To demonstrate that our work can straightforwardly address their proposed scenario, we cast it as a MAPR problem instance and evaluate our approach on it, following the method of evaluation which appears in [7]. We will first present the scenario and then proceed to describe how our approach was successful in addressing the scenario.

Example 3.3 *[USAR Scenario, taken from [7]] As illustrated in Figure 3.3, the USAR task occurs in a building with a long hallway. Rooms 1 and 2 are at the extreme end of one side, whereas rooms 3-5 are on the opposite side. In the example, the robot (located between hallway 5 and 6) is notified that Commander X's (located in room 3) goal is to perform triage in room 5. The robot is tasked with retrieving a medical kit and bringing it to a certain room, however, we will focus here on the task of predicting and recognizing Commander X's plans (integration of planning and plan recognition is beyond the scope of this thesis, as discussed in section 5.2). The robot knows that Commander X requires a medical kit in order to perform triage and can therefore reason that the commander will head to the room in which she believes the medical kit is located. The commander's choice of room depends on many factors, including her beliefs about the environment (the location of the medical kits), which could be different than those held by the robot (the robot, in some variations of the scenario, is notified that there is an additional medical kit, a fact that is unknown to the commander).*

The authors evaluate their approach to belief modeling and plan recognition by tasking their system with recognizing the commander's plan (i.e. which medical kit is she most likely to retrieve); the authors run an extensive set of simulations by varying the different parameters associated with the scenario. We follow their evaluation approach and vary the number of medical kits the robot believes the commander knows about (1 vs. 2); we vary the believed (by the commander) location of each medical kit (rooms 1-5); finally, we vary the believed goals of the commander (triage in room 1, room 5, or both). The different configurations yield 90 distinct cases; the authors compare the recognized plan of the commander with what would be expected of a rational agent following an optimal plan to achieve their goal. The ground truth plans of the commander were computed similarly to the ground truth plan generation described above, pertaining to scenarios 1 and 2. We treat the communication received by the agent (e.g., the initial location of the commander, as well as her goals and the location(s) of the medical kit(s)) as observations and compute $P(\pi_{commX} \mid O)$ in order to recognize the commander's most likely plan. Similarly to the results presented in [7], we also achieve 100% accuracy when predicting which medical kit the commander will choose. In section 5.2, we compare our approach to that of [7].

3.6.1 Complexity Analysis

In this section, we present an analysis of the runtime complexity of our approach to solving the formulation of the MAPR problem presented in this chapter. The runtime complexity of our approach can be divided into two parts: the encoding time, which is the time required to complete the compilation process and transform the epistemic MAPR problem into a classical planning problem; and the solving or computation time, which is the time it takes to reach our desired inferences, i.e. the solutions to the MAPR problem, given the classical planning problem.

Compilation runtime complexity: the runtime here is dominated by the first step of the compilation, where we make use of the encoding to classical planning proposed in [14]. Specifically, there is an exponential number of newly introduced fluents, due to the exponential number of entailments given each RML with a belief modality. Further, the max depth of nested belief, d, produces a bottleneck since the number of newly introduced fluents is also exponential in d. To reduce the computational complexity, the authors, in later work, introduced AK (always known) fluents, which are fluents that all agents have common and complete knowledge about. The introduction of AK fluents allows the compilation to focus on just the fluents that are important to the domain wrt belief. Analyzing which fluents are revelant to the domain, however, required a relevance analysis which was extremely expensive, computationally. Thus, we elect, in this work, to experiment with a low bound of nested belief and leave the question of optimizing complexity and deeper nested belief to future work.

Computation runtime complexity: as mentioned, in order to compute the probability distributions $P(\pi|O)$ and $P(G|O)$, we run the Top-k planner, TK*, once on the transformed classical planning problem P'. The runtime complexity of TK* is analyzed in [69] and [71]. When solving the top-k planning problem, a sequence of cost-optimal planning problems are solved, starting with finding a plan of optimal length n for the initial planning problem. A new set of planning problems is created in each round, where the size of the new set is dependent on the number of problems, m, created at a previous iteration. The actions in the planning problems at a certain iteration are modified so that an action is prevented from reappearing, if it had appeared in a previous iteration. This process is repeated until k best plans are found. Finally, the worst case time complexity for this process and for the TK* planner is $O(m + nlogn + kn)$.

Note that the overall runtime of our approach is dominated by the complexity of the compilation process. As part of future work, an improved approach could take advantage of specific domain properties in order to make small adjustments to the compilation, rather than compiling the MAPR problem 'from scratch'. Additionally, as part of future work we would like to perform online plan recognition; in such a setting, the probabilities of plans and goals are updated as the recognition system receives new incoming observations. Online recognition, thus, would require us to compile the MAPR problem from scratch for every new observation; therefore, it would be extremely beneficial to devise a better method of doing so.

3.6.2 Scalability

In this section, we evaluate the scalability of our approach to more complex problems, with a larger number of agents. To do so, we vary the number of agents in scenario 1, as described above and in section 3.1. For example, where $|Ag| = 4$ we have 4 small robots, each headed towards a different room in search of victims. As can be seen in Table 3.1, and as claimed in the previous section, the time required to compile the problem dominates the total runtime. While the solving time (one call to the TK* planner) does not increase dramatically with the number of agents, the compilation time does.

# of Agents	2	4	6	8
Solving time (s)	0.7	12.5	30.3	57.8
Encoding Time (s)	21.7	160	427.8	603.5

TABLE 3.1: The solving and encoding time, in seconds, of scenario 1, as appears in Example 3.1, with a varying number of agents

In the $|Ag| = 2$ case, the unmodified Example 3.1, we see that a solution is reached quickly and that our approach could potentially be applied to a real-world scenario. This is promising, however, it will be important for future work to reduce the computational complexity of the compilation process in order to address problems of a larger scale.

Chapter 4

MAPR with Temporal Actions and Unreliable Observations

In this chapter we continue to address sub-problem #2, *sub-problem #3*, and *sub-problem #4* by characterizing a second novel formulation of the MAPR problem. This chapter is an adaptation of the author's published work ([73] [74]), which began during an internship at IBM Research, mentored by Dr. Shirin Sohrabi. The work served as an important first step in the MAPR as planning research, and has helped the author make his first steps in the academic world.

The structure of this chapter is as follows: first, we present a motivating example; second, we define the main problem addressed in this chapter, namely the MAPR problem with temporal actions and unreliable observations which relaxes two strong assumptions made by previous work; third, we show how to transform the MAPR problem with temporal actions and unreliable observations to a temporal planning problem; fourth, we present our proposed AI planning-based computational approaches to solving this formulation of the MAPR problem; finally, we present the results of our experimental evaluation on a set of novel benchmarks, consisting of various MAPR problem instances.

4.1 Motivation

Example 4.1 *[Running Example] Let us consider an example, illustrated in Figure 4.1, taken from the International Planning Competition (IPC) Depots domain; in this domain, there are two different types of agents: hoist operators and truck drivers. In this example, and in this chapter in general, a common goal is shared by all agents and possibly 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 and their activities are interdependent and complementary. 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 shared goal of the agents is to move a set of crates between different locations.*

Crucially, in the MAPR context we do not know, a priori, the goals the agents are trying to achieve, nor do we know how they plan to achieve those goals. The recognition system is only given partial information, a sequence of observations, about the state of the world (or the actions of the agents in the previous chapter), as affected by the actions of the agents. To illustrate, in Figure 4.1 the white areas indicate the observations given to the recognition system; for example, at 08:22, we know that the red truck was at depot 1, with the green crate loaded onto it. Given a MAPR problem where (1) actions are temporal and may occur concurrently (e.g., two truck driver agents driving at the same time); (2) agents have different skill sets (e.g., truck driver agents drive and hoist operator agents lift); and (3) observations could be unreliable (e.g., a faulty sensor might provide the recognition system with an incorrect location of a truck), the task is to recognize the goals and the plans of the agents given the observations. For example, the dashed and solid red lines represent alternative hypothesized plans which the red truck might be pursuing, given the observations. The common goal of the agents in this case is getting the orange crate to depot 1 and the green crate to depot 3.

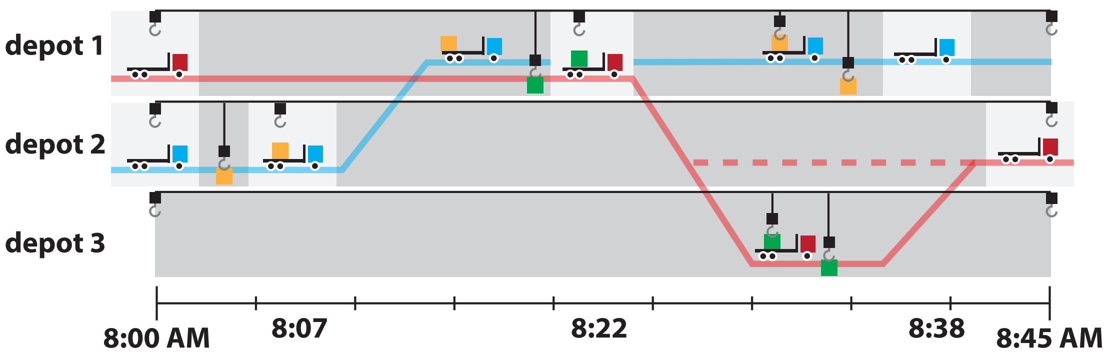

Figure 4.1: A timeline illustrating a 45-minute timeframe for the Depots domain, with 2 trucks drivers and 3 hoist operators. The y-axis is the three depot locations; the x-axis is the time line. The white areas indicate the observations. The lines represent alternative possible sequences of actions given the observations (color image).

Importantly, as previously discussed, the recognition process requires taking into consideration all manner of interaction between agents, in addition to the temporal aspects of the domain and the observations, and cannot be achieved by breaking apart the MAPR problem into many single-agent plan recognition problems. Instead, our approach transforms the MAPR problem into a temporal planning problem whose plans and makespans approximate the probability distributions of goals and plans given the observations. Finally, by addressing this previously unaddressed combination of elements, we relax two strong assumptions made by previous work: i. we do not assume that the actions of the agents are instantaneous, thus allowing for a more realistic modelling of the temporal aspects of the domain, including concurrency and interleaving of actions; ii. we no longer assume that the recognition system is given a perfect sequence of observations, thus making our approach robust to both noisy and missing observations.

4.2 Assumptions

Before introducing the multi-agent plan recognition problem with temporal actions and unreliable observations, we consider the assumptions made in this chapter:

Agents behave rationally: agents choose to perform actions which lead to an expected optimal result, where optimality is defined relative to some objective metric. In this chapter, the objective metric is the makespan of a plan, defined as the time that elaspes between the beginning of the first action and the end of the last action in the plan. Note that this is predicated on the assumption that minimizing the makespan of a plan corresponds to the rationality of the agents. In certain cases we may wish to augment this assumption to reflect the importance of factors other than makespan. For example, an alternative to reducing makespan is minimizing the effort for each agent in the joint plan.

Multiple executing agents: in this work, we assume that the different agents are following one, centrally planned, joint plan, in pursuit of a common goal. That is, we assume a single planning agent and multiple executing agents which carry out the plan.

Agents are not competing and share a common higher-level goal: we assume that agents are not adversarial and do not hinder other agents' plans. We further assume that all agents share a common goal (for example, moving a set of crates between locations). As mentioned, sub-goals might be formed and distributed amongst the different agents.

Heterogeneous group of agents: we offer in this chapter a highly expressive framework in which to represent the heterogeneous nature of a group of agents. Different sets of capabilities can be associated with the different agents in the domain. In our example, truck drivers can drive between locations, while hoist operators can load and unload crates onto and from trucks.

Actions are not instantaneous: in this work, we assume that actions have durations and that agents can act concurrently. In order to address concurrency and actions with durations, we leverage advances in temporal planning, as will be explained in the next section.

Nature of the observations available to the observing agent: in this chapter, we assume that the sequence of observations available to the observing agent are over the state of the world, rather than over the actions of the agents (this will be elaborated upon in the next section).

4.3 Problem Definition

In this section, we define the multi-agent plan recognition problem with temporal actions addressed in this chapter, and its solution. As mentioned in the previous chapter, the multi-agent planning problem, as defined in Definition 3, while sufficient to solve various classes of problems, makes strong assumptions which prevents its applicability to a wider range of problems. Specifically, the actions of the agents are assumed to be instantaneous; as discussed, in many real-world scenarios, the effects of the agents' actions are not instantaneous. For example, when passing a soccer ball to a fellow player, some amount of time will have elapsed before the ball reaches its destination. To model temporal actions with durations, as well as notions of concurrency between the agents' actions, we choose to leverage advances in Temporal Planning (e.g, [75], [76]), which provides a rich framework in which to represent the temporal aspect of a domain, including concurrency and action durations. We first modify the basic planning problem, as defined in Definition 1, to include temporal actions as defined in [75].

Definition 13 *[**A 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 in Definition 1, and A is a set of temporal actions. Each $a \in A$ is associated with a 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 $a \in A$ is *executable* in a state s_{start}, ending in state s_{end} if $\mathrm{pre}_s(a) \subseteq s_{start}$ and $\mathrm{pre}_e(a) \subseteq s_{end}$. The resulting states $s_{start'}$ and $s_{end'}$ are defined as $s_{start'} = ((s_{start} \setminus \mathrm{del}_s(a)) \cup \mathrm{add}_s(a))$ and $s_{end'} = ((s_{end} \setminus \mathrm{del}_e(a)) \cup \mathrm{add}_e(a))$. Note that s_{end} comes after $s_{start'}$. Additionally, the overall precondition, $\mathrm{pre}_o(a)$ 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. The makespan of the solution is the total time that elapses between the beginning of the first action and the end of the final action. To use Example 4.1, the actions *drive*, *load*, and *lift*, each have an associated duration and are temporal. For example, the add_s effect of the *drive* action is that the truck is no longer at the origin location; similarly, the add_e effect is that the truck is at the destination location. Also as is mentioned, it is possible for two actions to occur concurrently. In this work, we assume that the only change to the system is caused by the actions of the agents, specifically by the end effect of a temporal action executed by some agent. We further assume a discrete notion of time such that each action end effect occurs at a particular instant in time and marks a change of state in the system. Finally, between the end effects of any two actions, no change in the system is assumed to occur.

Next, we augment the basic definition of the multi-agent planning problem, as defined in Definition 3, with temporal actions and define the multi-agent planning problem with temporal actions as follows:

Definition 14 *[**Multi-Agent Planning Problem with Temporal Actions**] A Multi-Agent Planning Problem with temporal actions is a tuple $P^m = (F, \{A_i\}_{i=1}^N, I, G)$, where F and I are defined as before, G is the goal of the multi-agent problem, achieved by N agents, each with their own set of temporal action descriptions, A_i, $1 \leq i \leq N$.*

Note that the notion of concurrency amongst actions is modeled via temporal actions. This stands in contrast to much past research (e.g., [77, 78]) which used joint actions and defined concurrency constraints over them. The use of joint actions to model concurrent actions performed by multiple agents is restricting in that a single agent cannot perform two actions concurrently. Use of temporal actions allows concurrency of a single agent's actions as well as actions of different agents. The solution to P^m is a set of action-time pairs, where the action in each pair belongs to some A_i, $1 \leq i \leq N$; note that G, as before, is a conjunction of fluents that must hold in the final state.

Next, we define the plan recognition problem with temporal actions, as well as unexplainable and missing observations, adapting the definitions of Sohrabi et al. [15], where quality as measured by cost is used instead of action durations to approximate the probability values.

Definition 15 *[**PR Problem with Temporal Actions**] A plan recognition problem with temporal actions is a tuple $P^r = (F, A, I, O, \mathcal{G}, \text{PROB})$, where F, I, are defined as before, A is a set of temporal actions as defined earlier, $O = [o_1, ..., o_m]$ is the sequence of observations, where $o_k = (f_k, t_k)$, $1 \leq k \leq m$, $f_k \in F$ is the observed fluent, t_k is the* *time f_k was observed, and $\forall o_i, o_j$, if $i < j$ then $t_i < t_j$. $\mathcal{G}$ is the set of possible goals G, $G \subseteq F$, and* PROB *is a probability distribution over $\mathcal{G}$, specifying the prior probability of a goal, $P(G)$.*

Definition 16 *[**Unexplainable/Missing Observations**] Given an observation sequence O and a plan π for a particular goal G, an observation $o = (f, t)$ in O is said to be unexplainable (aka noisy), if f is a fluent that does not arise as the consequence of any of the actions a_i from the plan π for G. In contrast, an observation $o' = (f', t')$ is said to be missing from O, if o' is not in the sequence O and f' is added by at least one of the executed actions $a_i \in \pi$.*

In this chapter, we consider sequences of observations where each observation $o_i \in O$ is an observable fluent, with a timestamp that indicates when that fluent was observed. We focus on observations that range over fluents rather than over actions, as actions may not be directly observable but rather inferred via the changes they manifest. In this work, as mentioned, we assume that fluents only arise as the consequence of the agents' actions, thus assuming that the environment does not change apart from that. This is a simplifation and relaxing this assumption is left to future work. If two observable fluents are observed at the same time, we increase the timestamp of one by an arbitrarily small duration. Note, we assume that the act of observing an observation is instantaneous and adheres to the order in which the observable fluent appears in O. To illustrate, in Figure 4.1, ((at *redTruck depot1*), 08:00) is a possible observation in O.

Also note that both missing and unexplainable observations belong to the class of unreliable observations. To address the unexplainable observations, Sohrabi et al. 2016 modify the definition of satisfaction of an observation sequence by an action sequence introduced in [68] to allow for observations to be left unexplained.

Definition 17 *[**Satisfaction of a Sequence of Observations (following Sohrabi et al. [15]**] 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 monotonic 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 $o_j \notin s_{f(j)}$.*

Hence, we take into account the order in which the observations were made through the mapping of the non-decreasing function and also allow for observations to be left 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 unexplainable. Note that in the definition above, o_j refers to the fluent, f, in the observation tuple (f, t). Additionally, in line with the assumptions outlined after definition 13, every s_i in σ is the end effect of some temporal action a_i which occurs at a particular instant in time and marks a change of state in the system.

Finally, the solution to the plan recognition problem, P^r, is given as two probability distributions, the probability of plans given observations, $P(\pi|O)$, and the probability of goals given observations, $P(G|O)$.

Next, recall the basic definition of the MAPR problem, as defined in Definition 5. In this chapter, we wish to relax two of the various strong assumptions made by this basic formulation and by previous work in MAPR: i. the actions of the agents have no durations; ii. the observation sequence, O, is perfect and reliable. We do so by augmenting the MAPR formulation with a temporal planning framework, as well as extending the notion of satisfying a sequence of observations, to address unreliable observations.

Finally, we put everything together and define the problem we address in this chapter:

Definition 18 *[**MAPR Problem with Temporal Actions**] The Multi-Agent Plan Recognition (MAPR) problem with temporal actions is described as a tuple $P = (F, \{A_i\}_{i=1}^N, I, O, \mathcal{G}, \textsc{prob})$, where F is a finite set of fluents, A_i is a set of temporal actions for agent i, $1 \leq i \leq N$, $I \subseteq F$ defines the initial state, $O = [o_1, ..., o_m]$ is the sequence of observations, where $o_k = (f_k, t_k)$, $1 \leq k \leq m$, $f_k \in F$ is the observed fluent, t_k is the time f_k was observed, $\mathcal{G}$ is the set of possible goals, $G \in \mathcal{G}$, $G \subseteq F$, and* PROB *is a probability distribution over $\mathcal{G}$, specifying the prior probability of a goal, $P(G)$.*

Definition 19 *[**Solution to the Multi-Agent Plan Recognition Problem with Temporal Actions**] Given a multi-agent plan recognition problem with temporal actions, $P = (F, \{A_i\}_{i=1}^N, I, O, \mathcal{G}, \textsc{prob})$, the solution is given as two probability distributions. The first is the probability of plans given the observations, $P(\pi|O)$, where each π is a plan that achieves a goal $G \in \mathcal{G}$ and satisfies the observation sequence, O. The second distribution is the probability of goals given the observations, $P(G|O)$, where each G assigned a non-zero probability is a goal achieved by a plan in the first distribution.*

The set of possible goals, $\mathcal{G}$, as discussed in the previous chapter, can be seen as a set of alternative hypotheses for the common high-level goal shared by all agents, which are each assigned a posterior probability, given the observations. Note that each goal $G \in \mathcal{G}$ is a conjunction of fluents from F which must hold in the final state. In the most general case, and in the absence of prior knowledge, it is the set of all possible goals the agents might be pursuing. That is, $\mathcal{G}$ could contain all possible combinations of all possible grounded instances of the fluents in F. In some cases, however, we have at our

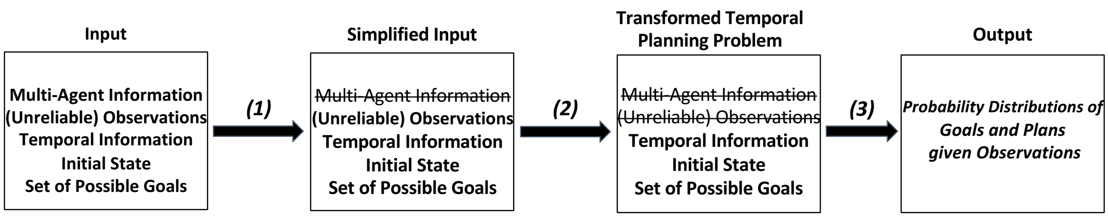

FIGURE 4.2: A pipeline showing our proposed compilation approach: transforming the original MAPR problem with temporal actions and unreliable observations into a plan recognition problem (1), a transformation step that compiles away the observations (2), allowing the use of temporal planning to compute a solution to the MAPR problem (3).

disposal some domain knowledge. Using this knowledge, $\mathcal{G}$ can be populated only with alternative possible goals, given the domain and problem specifications. For instance, in the Depots domain, $\mathcal{G}$ might contain the following possible shared and common goals: 1. (at orangeCrate depot1) $\wedge$ (at greenCrate depot3) and 2. (at orangeCrate depot3) $\wedge$ (at greenCrate depot3). The solution to the respective MAPR problem would then be assigning probabilities to the two goals, given the observations, thus deciding which one is more likely being pursued by the agents. In Figure 4.1, (at orangeCrate depot1) $\wedge$ (at greenCrate depot3) is the actual goal pursued by the agents; if successful, our approach would assign a higher likelihood to this goal, deeming it the most likely given the observations.

4.4 Transformation

In this section, we describe a two-step compilation technique, formalized in Algorithm 2, that allows the use of temporal planning on the MAPR problem. That is, we first transform the given MAPR problem as defined in Definition 18 into a plan recognition problem with temporal actions; second, we transform the plan recognition problem into a temporal planning problem; finally, we use temporal planning to compute the solution to the MAPR problem, namely the probability distributions of plans and goals given observations, in keeping with the previous plan-recognition-as-planning approaches. The compilation pipeline is shown in Figure 4.2.

Algorithm 2: Transforming the MAPR Problem with Temporal Actions to a Temporal Planning Problem

Input: A MAPR problem with temporal actions, $P = (F, \{A_i\}_{i=1}^{N}, I, O, \mathcal{G}, \textsc{prob})$, as defined in Definition 18

Output: A temporal planning problem, $P' = (F, A, I, G)$, as defined in Definition 13

$P^r = \textsc{TransformToSingleAgentPRProblemWithTemporalActions}(P)$

$P' = \textsc{TransformToTemporalPlanningProblem}(P^r)$

RETURN P'

4.4.1 Step 1 of Algorithm 2 - Transformation to the Plan Recognition Problem with Temporal Actions

To transform the original MAPR problem with temporal actions to a single-agent plan recognition problem with temporal actions, as defined in Definition 15, we make use of the technique proposed by Muise et al. 2015, which maps multi-agent planning problems to single-agent planning problems, thus enabling the use of single-agent classical planners. We modify this technique in a straightforward way so that it creates a temporal planning domain theory instead of a classical planning domain theory, given our input. The technique compiles away the multi-agent information by 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. To address the temporal aspect, the introduced action precondition is adjusted to meet the specifications of a temporal action; action durations are left unchanged.

4.4.2 Step 2 of Algorithm 2 - Transformation to a Temporal Planning Problem

Next, we compile away the observations, so that the plan recognition problem can be solved using the plan-recognition-as-planning approaches (see discussion in section 3.4.2). These approaches view the plan recognition problem as an inverse planning problem, in that the goals and plans of the agents are not known to the system, and the goal of the transformed planning problem becomes explaining the given observations. 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 [13], as we do in the previous chapter. Observations can also be compiled away following Haslum and Grastien [80] using the "advance" action that ensures the observation order is preserved; another paper that addresses the compilation of observations is [81], where a goal recognition design problem is compiled into a classical planning problem and observations, which are over the agent's actions, are compiled into the transformed planning problem. In this chapter, however, observations are defined over the fluents, so we will follow the technique proposed in [15], which extends Ramírez and Geffner's approach by addressing unexplainable and missing observations. To incorporate a temporal aspect into the compilation process, our work replaces the notion of cost with that of duration, and compiles the observations into temporal actions that are part of the transformed temporal planning domain.

The transformation compiles away observations, using special predicates for each fluent in the observation sequence O, while ensuring that their order is preserved. Specifically, we ensure that observation o_1 with timestamp t_1 will be considered (explained or discarded) before observation o_2 with timestamp t_2, where $t_1 < t_2$; o_1, as explained previously, will appear before o_2 in the observation sequence O. 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 serves to encourage the planner to explain as many observations as possible. We also update the duration of the original actions, by adding a constant duration to each action; this is the penalty for the possible missing observations, which encourages the planner to use as few unobserved actions as possible. While these penalties artificially inflate the makespan of the plans, we are able to post-process these plans by removing the extra actions and updating the durations of the actions. 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, we add a special predicate *done* to the goal of the transformed planning problem. To ensure that all observations are considered, we also add a predicate, p_{last}, corresponding to the final fluent in the observation sequence, to the goal state of the problem. In addition, we add an action a_g for each goal $G \in \mathcal{G}$ with precondition g (the fluents correspond to goal G), and with the add effect *done* to the set of actions. Hence, the goal of the transformed planning problem is defined such that O is satisfied (as defined in Definition 17) and at least one of the goals $G \in \mathcal{G}$ is achieved.

Note, after solving the transformed single-agent temporal planning problem, we are able to straightforwardly rewrite the solution, a single-agent temporal plan, as a multi-agent temporal plan, such that we can attribute the different actions in the plan to the corresponding agents. By so doing, we are able to hypothesize about the plan of each of the agents, given the observations. For example, in Figure 4.1, a possible solution to the transformed temporal planning problem that successfully explains the observations (in the white areas), is a joint plan, comprised of the actions of all five agents. This plan can be decomposed into the individual plans of the different agents, using the tokens associated with each agent (e.g. *blueTruck* in the action (drive *blueTruck* depot2 depot1)); specifically, the blue and red trucks' hypothesized plans, which successfully satisfy the observations, are represented by the blue and red lines, respectively.

In the next section, we propose three different AI planning-based computational approaches to compute the solution to the MAPR problem. In order to apply the Delta approach, as well as the Hybrid approach, we modify the transformation discussed above to not include the *done* predicate, as a new planning problem will be generated for each goal separately (this will be elaborated upon in the next section). In addition, the "discard" actions are removed for our Delta approach that is based on [68] as this approach does not address the unexplainable observations by discarding them.

Theorem 3 *[Soundness and Completeness] Given a MAPR problem with temporal actions, $P = (F, \{A_i\}_{i=1}^N, I, O, \mathcal{G}, \text{PROB})$, as defined in Definition 18, and the corresponding transformed temporal planning problem $P' = (F', A', I', G')$, which is the result of applying Algorithm 2 to P, we have the following:*

(1) For all $G \in \mathcal{G}$, if π is a plan for the planning domain $(F, \{A_i\}_{i=1}^N, I)$ and goal G, then there exists a plan π' for the corresponding planning problem, P', such that the plan π can be constructed straightforwardly from π', by removing the extra actions (i.e., discard, explain, and goal actions) from π' and updating the duration of the actions in the planning domain such that $d(\pi') = d(\pi) + M + (b_2 \cdot D)$, where $d(\pi)$ is the makespan of the plan π, M is the cumulative incurred penalty for missing observations, $b_1 \leq M \leq b_1 \cdot |\pi'|$ ($M = 0$ if $|\pi'| = 0$), $|\pi'|$ is the number of actions in π', D is the number of discard actions in π', and b_1 and b_2 are positive coefficients that express weights to the importance of missing and unexplainable observations, respectively.

(2) If π' is a plan that achieves G' in P', then there exists a plan π that achieves some goal $G \in \mathcal{G}$ and satisfies O in P, where π can be constructed straightforwardly from π' by removing the extra actions (i.e., discard, explain, and goal actions) from π' and updating the durations of the actions such that $d(\pi') = d(\pi) + M + (b_2 \cdot D)$, as described in (1).

Note that the lower bound for M, the cumulative incurred penalty for missing observations, is b_1 since that will be the penalty if $|(\pi)| = 1$ or if there exists one temporal action in the plan, a_i, such that all other temporal actions in the plan, $a_j \neq a_i$, occur during a_i's execution; the upper bound for M is $b_1 \cdot |(\pi)|$ when all actions are executed sequentially.

Proof:

(1)

Let G be a goal in $\mathcal{G}$. Assume that π is a plan that achieves the goal condition G and satisfies O in P. We prove that there exists a plan π' that achieves G' in P', such that the plan π can be constructed straightforwardly from π'. We do so by showing a correspondence between π and π', which only differ in the inclusion of the extra actions (i.e., explain, discard, and goal) and the modified durations of the actions in their respective planning domains. First, we note that the extra actions only preserve the ordering amongst the observations and do not change the state of the world. Thus, after removing these actions from π', we are left with π which achieves G and satisfies the observations. Specifically, we know that the goal G is achievable since π achieves it (if G is not achievable then (1) holds trivially). Thus, one possible solution for the planning problem P', is a plan, π', that achieves G and satisfies the observation sequence, O. Since *done* holds in π' and G is achieved, we know that the action $a_G \in A_{goal}$, corresponding to G, is included in π'. Further, following definition 17, we know that there exists a non-decreasing function f that maps the observation indices into the state indices, such that every observation is either included or not included in a state. We know, thus, that the corresponding discard or explain actions, for each of the observations in O, will appear in π' since it achieves G'. Note that π' achieves G' in P'. In order to construct π from π', we remove the goal action a_G from π'. Finally, the total makespan of the transformed planning problem, P', incorporates the objective function that includes the original duration of the actions, as well as the penalty incurred for the missing and unexplainable observations. Thus, after removing the extra actions from π' we have that π' new makespan is equal to that of π. In this way, we create a correspondence between $d(\pi')$ and $d(\pi)$, as is described in Theorem 3.

(2)

Assume that π' is a plan that achieves that achieves G' in P'. We prove that there exists a plan π that achieves some goal condition $G \in \mathcal{G}$ and satisfies O in P. We do so by showing a correspondence between π and π', which only differ in the inclusion of the extra actions (i.e., explain, discard, and goal). First, since π' achieves G', we know that *done* holds. Therefore, following the definitions in section 4.4.2, it follows that at least one of the goals $G \in \mathcal{G}$ is satisfied and that some action $a_G \in A_{goal}$ is included in π'. Hence, we construct π from π' by removing the extra goal action $a_G \in \pi'$. After removing a_G, we know that the goal G is still satisfied in the final state, after the execution of π (since a_G has no effect on the state of the world). Next, we remove the discard and explain actions from π', since these only preserve the order of the observations and do not affect the state of the world. Hence, following the definitions in section 4.4.2, it follows that O is still satisfied in π. Finally, as done in the proof for (1), we create a correspondence between $d(\pi')$ and $d(\pi)$ as is described in Theorem 3. □

Theorem 4 *[Algorithm 2 is solution preserving]*

Proof: This follows directly from Theorem 3, where we have shown that no solution to the input, P, is lost after applying Algorithm 2 to it. □

We have shown, then, that our compilation of the multi-agent plan recognition problem with temporal actions to a temporal planning problem is solution preserving, and that plans that solve the transformed planning problem correspond to plans that solve the various planning problems within the original MAPR problem. Using this correspondence, we will see in the next section how $V(\pi)$, which maps to the makespans of plans which solve the transformed planning problem, is used to approximate $P(O|\pi)P(\pi|G)$. This approximation, in turn, will allow us to compute the probability distributions, $P(G|O)$ and $P(\pi|O)$, which are the solutions to the original MAPR problem, as defined in Definition 18.

4.5 Computation

In this section, we lay out our approaches to computing a solution to the MAPR problem, as described in Definition 18, namely the probability distributions of plans and goals, given observations. We present three planning-based approaches to computing the probability distribution of goals given the transformed planning problem, as described in the previous section. Note that while we focus on recognizing the goals of agents in our experimentation, both the Diverse and the Hybrid approaches first compute the probability distribution of plans in order to compute the probability distribution of goals; the first approach, Delta, can theoretically be extended to compute the probability distribution of plans, and this is left to future work.

4.5.1 Computing $P(G|O), G \in \mathcal{G}$

The first approach (Delta) 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 goal recognition approach proposed in [68]. The second approach (Diverse) computes the probability distribution of goals by finding a set of diverse plans, that serves as a representative approximation of the distribution of plans that satisfy the observations and achieve one of the possible goals ($P(\pi|O)$); it is a modification of the proposed approach in [15]. The third approach (Hybrid) is a combination of the two previous approaches, in that it computes a set of plans for each of the goals. Note that the Diverse and Hybrid approaches both compute the probability distribution of plans given observations in order to compute $P(G|O)$, while the Delta approach is not capable of doing so without further modifications.

4.5.1.1 Approach 1 : Delta

Given the transformed temporal planning problem, this approach computes the probability distribution of goals given 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' Rule as:

$$P(G|O) = \alpha P(O|G)P(G) \tag{4.1}$$

where α is a normalization constant and $P(G)$ is PROB or the goal priors. The cost (or makespan) difference, or Δ, is defined as the difference in the makespan of the optimal plan that achieves G and O, and the makespan of the optimal plan that achieves G but not O. Assuming a Boltzmann distribution, as is assumed in [68], $P(O|G)$ is defined as:

$$P(O|G) \approx \frac{e^{-\beta\Delta}}{1 + e^{-\beta\Delta}} \tag{4.2}$$

where β is a positive constant. This approach assumes that the agent pursing goal G is more likely to follow cheaper plans and 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 for each setting of the problem.

4.5.1.2 Approach 2 : Diverse

Given the transformed temporal planning problem, this approach computes an approximation to the probability distribution of plans as well as goals, given the observation, by running a diverse temporal planner on the transformed temporal planning problem. In particular, it first computes $P(\pi|O)$ as follows:

$$\begin{aligned} P(\pi|O) &= \beta P(O|\pi)P(\pi) \\ &= \beta P(O|\pi) \sum_G P(\pi|G)P(G) \\ &= \beta P(O|\pi)P(\pi|G)P(G) \end{aligned} \tag{4.3}$$

where β is a normalizing constant that depends on $P(O)$ only, and $P(G)$ is PROB(G). Note, we assume that only one goal is being pursued and $P(\pi|G)$ is 0 for the action sequences π that are not plans for G. $P(O|\pi)P(\pi|G)$ is approximated as follows:

$$P(O|\pi) \cdot P(\pi|G) \approx 1 - \frac{\beta' V(\pi)}{\sum\limits_{\pi'' \in \Pi} V(\pi'')} \tag{4.4}$$

where β' is a positive constant, Π is a sampled set of plans that satisfy the observations and achieve at least one of the goals $G \in \mathcal{G}$; $V(\pi)$, which respects the objective function as mentioned in section 4.4, is the makespan of the plan that is the solution to the transformed planning problem and is equal to the sum of the original duration of the actions plus M plus b_2 times D. Coefficients b_1 (incorporated in M) and b_2 are used to give weights to the importance of the original actions together with the potential of having missing observations and unexplainable observations, respectively. D is the number of "discard" actions in π and $b_1 \leq M \leq b_1 \cdot |(\pi)|$ (see Theorem 3).

Using Bayes rule, the probability distribution of goals given observations is then computed by a summation over all values of $P(\pi|O)$ for the sampled set of plans, Π, that achieve G and satisfy O, and a subsequent normalization of the summation values (more details in [15]).

$$P(G|O) = \sum_{\pi \in \Pi} P(\pi|O) \tag{4.5}$$

The set of plans Π is computed using diverse planning, which is defined as follows:

Definition 20 *[Diverse Planning Problem] A diverse planning problem is a tuple (m, d), 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.*

Several techniques exist for computing the set of diverse plans (e.g., [82–85]); in this chapter, we use LPG-d [86], the diverse extension of a local search-based planner LPG [87]. Note that there is a large space of plans that achieve G and satisfy O and computing all of them is not practical; diverse planning is used as a means to approximate the probability distribution over these plans, by sampling a set of plans from this space.

The set of sampled plans is found by instructing the diverse planner to find m plans for the transformed temporal planning problem that satisfy the observation sequence and achieve at least one of the goals $G \in \mathcal{G}$.

The Diverse approach is identical to the computational approach presented in the previous chapter. The two implementations only differ in the objective metric which measures a plan's optimality, as well as their treatment of unreliable observations. In this chapter, we address potentially unreliable observations; to this end, $V(\pi)$ respects the objective function which assigns weights to both missing and unexplainable observations. Hence, following Theorem 3 and the compilation process, we use the V value of a plan in Equation 4.4; in contrast, the simple cost of a plan is used in Equation 3.2 in the previous chapter, where unreliable observations are not addressed. Further, whereas in the previous chapter we use Top-k planning to find the set of plans, Π, in this chapter we elect to use diverse planning as we address a temporal component in our work which is not yet supported by existing Top-k techniques.

4.5.1.3 Approach 3: Hybrid

In order to take advantage of both previous approaches, we propose a hybrid approach in which we use a temporal planner to compute a smaller set of plans for each of the different goals. After merging the sets of plans, we are able to compute the probability distribution of goals, just as we did in the second approach. However, the Hybrid approach forces the 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. Thus, each of the possible goals is assigned at least one representative plan when computing the probability distribution over the different goals.

4.6 Experimental Evaluation

In this section, we present the results of our experimental evaluation, where we evaluate the goal recognition capabilities of our three proposed computational approaches. We begin by describing the experimental setup and benchmark generation, and proceed to present the results of our experimentation on these benchmarks.

To evaluate our MAPR approach, we used a temporal planner, LPG-TD [88], for the delta approach and the hybrid approach, and a diverse planner, LPG-d [86], for the diverse 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 [89], OPTIC [16]), either timed out on most problem instances, or did not accept the transformed planning problem as input. Note, the results for the diverse approach were obtained by running LPG-d once for each problem. LPG-TD was run once for each goal, i.e. $|\mathcal{G}|$ times, for the hybrid approach and $2 \times |\mathcal{G}|$ times for the delta approach. We used a timeout of 30 minutes and ran 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 a $(10, 0.2)$ setting of (m, d), since this setting performed best; thus, the planner computed 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 b_2 to be ten times b_1; thus, discarded observations are penalized more heavily than missing observations. The ratio between b_1 and b_2 was set based on experimentation, and as part of future work, should be tuned based on the domain in which the system is operating.

In this chapter, 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 collection of rovers navigate a planet surface, finding samples and communicating them back to a lander), Depots (trucks transport crates between depots and then the crates must be stacked onto pallets at their destinations by hoist operators), Satellites (a collection of observation tasks carried out by multiple satellites, each equipped differently), and ZenoTravel (transportation of people between cities in planes, using different modes of movement). The original domains each have separate temporal and multi-agent versions and are not plan recognition problems. We modify the domains to create benchmark problems for the MAPR problem with temporal actions. In combining the multi-agent and temporal aspects 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 used a modified version of the domains, where these issues are addressed.

To construct the MAPR problems, we compute a plan that is a solution to the original planning problem. From this plan, 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., the ground truth goal). The effects of these actions then serve as the fluents in O. Additional goals were created manually for each problem instance to populate the set of possible goals; the goals were created with approximately equal prior likelihoods. Automatic generation of goals was considered, and as part of future work this approach could be applied so as to create a more controlled set of possible goals. Overall, the set of possible goals for each problem in the benchmarks consisted of 4 goals. As the sets of goals were created manually, the number of goals was chosen mainly based on the amount of time consumed by the manual generation of goals. In order to introduce

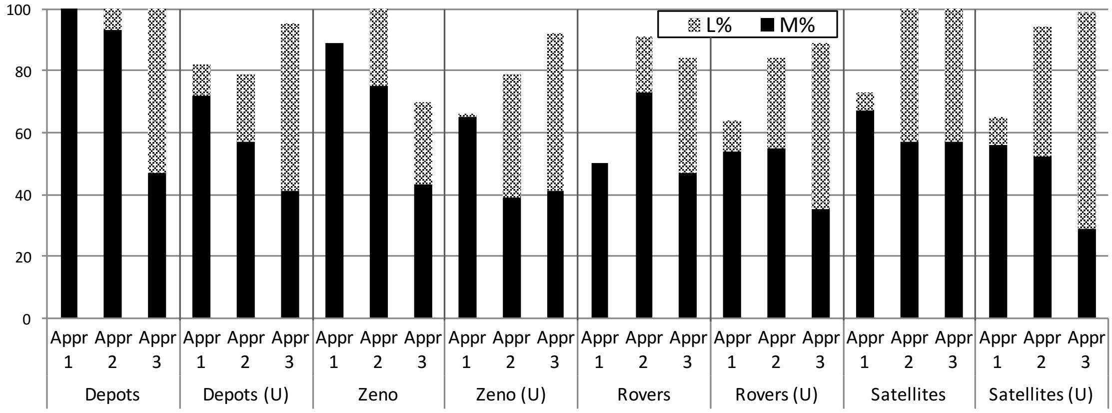

FIGURE 4.3: Comparison of our proposed approaches for recognizing a goal: (1) Delta, (2) Diverse, and (3) Hybrid

Appr	Depots		Zeno		Rovers		Satellites	
	R	U	R	U	R	U	R	U
Delta	15/20	81/220	9/18	83/198	2/19	52/209	18/19	198/209
Diverse	15/20	146/220	8/18	90/198	11/19	154/209	14/19	173/209
Hybrid	15/20	160/220	14/18	152/198	19/19	207/209	19/19	204/209

TABLE 4.1: Comparison of the number of problems solved by our three proposed approaches

missing observations, we created several variations of each problem that did not include the full observation sequence, by randomly selecting 10%, 40%, 70% and 100% of the observations in O.

Figure 4.3 shows the summary of the results when evaluating our three proposed MAPR approaches on goal recognition. Approach 1 is the delta approach, approach 2 is the diverse approach and approach 3 is the hybrid approach. Each domain consists of 16-20 problems; 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., introduced noise; 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 noise, only this time relative to the size of the ground truth plan. 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. We average over instances which were successfully solved before the timeout.

To evaluate the coverage and accuracy of the different approaches, we compute 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 posterior probability given the observations. These values, M and L, are shown respectively in the lower (solid black) and upper 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. 0.03 was chosen arbitrarily as a small number; as part of future work, the sensitivity of this threshold should be tuned based on the domain in which the system is operating.

The results in Figure 4.3 show that approach 1, on average, does best (i.e., highest M value) across all domains when observations are reliable and no noise is introduced. The results also show that, on average, approach 3 achieves the best coverage, i.e., the total height of the bars, across all domains, and also manages to successfully solve more problems than the other approaches. The total number of problems solved by each approach is shown in Table 4.1; U, as before, signifies cases where unreliable observations were introduced, whereas R signifies cases where they were not.

Note that 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 (i.e. less missing observations and hence larger observation sequences). 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 plans. This, combined with the the ground truth plans being 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 also account for the different results.

Further, in related work [15], Top-k planning has been used, successfully, in order to compute the set of plans required for the probability computation. While we wanted to compare different methods for creating the set of plans, we address a temporal component in our work and so could not use existing Top-k techniques. The diverse planner used, LPG-D, generated relatively low-quality plans, which might have affected the computed probabilities.

Finally, by addressesing different elements, namely temporal actions, a multi-agent setting and unreliable observations, our work offers greater depth to our inferences regarding the goals and plans of agents. However, this also makes the problem addressed in this chapter quite complex, rendering its computational solution expensive. Recent work [90] suggests an approach that propagates cost and interaction information in a plan graph, which are then used to estimate probabilities of goals. This proposed method might be enhanced and exploited here to reduce computation time. Further, we have attempted to apply the techniques in [91] so as to compile away the temporal aspect of the problem, thus transforming it to a single-agent classical planning problem. However, this approach did not scale well, causing many of our experiments to time out.

4.6.1 Complexity

As mentioned, we use several different temporal planners in order to obtain our results. Rintanen et al. [92] showed that the complexity of temporal planning without self-overlapping actions is equivalent to the complexity of classical planning, namely, PSPACE-complete (the complexity result for classical planning was obtained in [93]). The complexity of the transformation process is done with negligible memory and so the space complexity remains PSPACE-complete. The runtime complexity is dependent on the computational approach and the chosen off-the-shelf planner. For the diverse approach we ran LPG-d once, hence the complexity is the cost of a LPG-d run. For the Delta approach, LPG-TD was run $2 \times |\mathcal{G}|$ times and so the complexity is $O(2 * |\mathcal{G}| * C_{LPG-TD}) = O(|\mathcal{G}| * C_{LPG-TD})$, where C_{LPG-TD} is the cost of a LPG-TD run. Finally, For the Hybrid approach, LPG-TD was run $|\mathcal{G}|$ times and so the complexity is $O(|\mathcal{G}| * C_{LPG-TD})$.

Chapter 5

Conclusion and Future Work

In this section, we will discuss how the thesis addressed the research problems outlined in section 1.2; we will conclude by discussing important avenues for future research.

5.1 Addressing the Research Problems

In this thesis, we set out to answer a main research problem, namely, to characterize and solve novel multi-agent plan recognition formulations which are applicable to a wide range of real-world instantiations of the MAPR problem.

Chapter 2 addressed *sub-problem #1* by first enumerating a set of attributes which delineates the general case of the MAPR problem; next, previous work in the field was surveyed; finally, the surveyed previous work was characterized as different configurations of the enumerated attributes, and various strong assumptions made by this body of work were made explicit.

Next, chapters 3 and 4 turned to address *sub-problem #2*, *sub-problem #3* and *sub-problem #4* by characterizing two novel formulations of the MAPR problem, each relaxing different assumptions made by previous work. The chapters proceeded to propose different computational approaches which solve the novel formulations of the MAPR problem, thereby addressing new classes of previously unsolvable problems. The first formulation, proposed in Chapter 3 addresses the Epistemic MAPR problem, no longer assumes that all agents must share a common mental state, thus enabling the observing agent to consider the unique perspective of each agent when recognizing its plans and goals. The second formulation, proposed in Chapter 4, addresses the MAPR problem with temporal actions and unreliable observations, which relaxes the assumptions that (a) the agents' actions are instantaneous and (b) the observations are perfect and reliable.

Importantly, all of the proposed computational approaches conceive the computational core of the MAPR problem as an AI planning task, thus eliminating the need for rigid plan libraries, used by most previous work in MAPR. Finally, the power and flexibility of the proposed computational approaches were illustrated by demonstrating their applicability to various instantiations of the MAPR problem; to do so, domains and scenarios which specifically require the relaxation of the aforementioned assumptions were used.

5.2 Future Work

There are many exciting and important avenues left to explore as part of future work. For instance, as mentioned, plan recognition is key if we are to achieve successful human-machine and machine-machine coordination, cooperation and communication. One way of achieving these important objectives, is the integration of planning and plan recognition. Recall that in the scenarios that appear in section 3.1, in order to successfully intervene and assist, the recognition system must act upon its inferences regarding the goals and plans of the agents. While in this work we focus on the recognition component and leave the integration of planning and plan recognition for future work, previous research has successfully integrated the two in the single-agent plan recognition case. In [27], the authors integrated planning and plan recognition by allowing the observing agent to act on the recognized goals and plans of the observed agent.

Further, the approach presented in [7] also closes the planning-recognition loop and integrates the two, enabling an artificial agent to assist a fellow agent in life-or-death scenarios. The results presented in section 3.6 are promising since they demonstrate that our approach is able to perform the necessary reasoning in order to intervene and address Example 3.3, thus laying the foundations for future integration with planning and online assistance. Importantly, our approach, by utilizing a powerful theoretical framework, can address a large new class of problems, which cannot be addressed by previous work, including [7]. This claim is backed by a number of reasons. First, by building on the MEP framework, we provide a natural and expressive multi-agent representation, which allows us to model a heterogeneous group of agents, possibly holding disparate mental states; further, the approach in [7] cannot straightforwardly address Examples 3.1 and 3.2, as described in section 3.1. This is because, in order to successfully address the scenarios, the plans of multiple agents must be recognized, which themselves must reason about other agents, thus requiring a richer representational framework. Finally, we are able to show that our approach is solution preserving, a result which is missing from the work in [7]. We note that such theoretical contributions were not the aim of their work.

Next, in Chapter 3 we limit our experimentation to one domain, namely Urban Search & Rescue. Future work should test the applicability of our approach to a larger array of domains. Further, as part of future work, it will be interesting to extend the evaluation conducted in Chapter 4 to domains which explicitly introduce required cooperation [94], and that are not straightforwardly compilable to single-agent domains.

Finally, it has been shown in Chapter 3 that while our approach can successfully address previously unsolvable classes of the MAPR problem, it does not scale well due to the complexity of the compilation process. Hence, it will be important for future work to reduce the computational complexity of the compilation process in order to address problems of a larger scale. In Chapter 4, the computational solution to the problem is also quite expensive. By leveraging recently proposed methods for faster recognition, computation time could perhaps be reduced.

5.3 Concluding Remarks

To conclude, the work in this thesis enables the application of a MAPR approach to previously unaddressed classes of problems by modelling them in planning domains with underlying rich and expressive frameworks, and by relaxing strong assumptions made by previous work. The approaches presented in this work will hopefully enable researchers to, relatively straightforwardly, cast their research problem as a MAPR problem, thus allowing them to choose the planner that works best for their domain and compute an effective solution.

Bibliography

[1] Gita Sukthankar, Christopher Geib, Hung Hai Bui, David Pynadath, and Robert P Goldman. *Plan, activity, and intent recognition: theory and practice*. Newnes, 2014.

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

[3] Shirin Sohrabi, Octavian Udrea, and Anton Riabov. Hypothesis exploration for malware detection using planning. In *AAAI*, pages 883–889, 2013.

[4] Shirin Sohrabi, A Riabov, and Octavian Udrea. Planning-based scenario generation for enterprise risk management. In *Proceedings of the Scheduling and Planning Applications woRKshop (SPARK)*, 2017.

[5] Shirin Sohrabi, Anton Riabov, Michael Katz, and Octavian Udrea. An ai planning solution to scenario generation for enterprise risk management. In *Proceedings of the 32nd Conference on Artificial Intelligence (AAAI-18)*, 2018.

[6] E Yolanda, David E Smith, et al. Goal recognition with noisy observations. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[7] Kartik Talamadupula, Gordon Briggs, Tathagata Chakraborti, Matthias Scheutz, and Subbarao Kambhampati. Coordination in human-robot teams using mental modeling and plan recognition. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2957–2962. IEEE, 2014.

[8] Suchi Saria and Sridhar Mahadevan. Probabilistic plan recognition in multiagent systems. In *ICAPS*, pages 287–296, 2004.

[9] Bikramjit Banerjee, Jeremy Lyle, and Landon Kraemer. Multi-agent plan recognition: Formalization and algorithms. In *AAAI*, 2010.

[10] Hankz Hankui Zhuo, Qiang Yang, and Subbarao Kambhampati. Action-model based multi-agent plan recognition. In *NIPS*, pages 377–385, 2012.

[11] Gabriel Synnaeve, Pierre Bessiere, et al. A bayesian model for plan recognition in rts games applied to starcraft. In *AIIDE*, 2011.

[12] Mohammad Taghi Saffar, Mircea Nicolescu, Monica Nicolescu, Daniel Bigelow, Christopher Ballinger, and Sushil Louis. Intent recognition in a simulated maritime multi-agent domain. In *International Workshop on Machine Learning, Optimization and Big Data*, pages 158–170. Springer, 2015.

[13] Miquel Ramírez and Hector Geffner. Plan recognition as planning. In *IJCAI*, 2009.

[14] Christian J. Muise, Vaishak Belle, Paolo Felli, Sheila A. McIlraith, Tim Miller, Adrian R. Pearce, and Liz Sonenberg. Planning over multi-agent epistemic states: A classical planning approach. In *AAAI*, pages 3327–3334, 2015.

[15] Shirin Sohrabi, Anton Riabov, and Octavian Udrea. Plan recognition as planning revisited. In *IJCAI*, 2016.

[16] J. Benton, Amanda Coles, and Andrew Coles. Temporal planning with preferences and time-dependent continuous costs. In *ICAPS*, 2012.

[17] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: theory and practice*. Elsevier, 2004.

[18] James A Hendler, Austin Tate, and Mark Drummond. Ai planning: Systems and techniques. *AI magazine*, 11(2):61, 1990.

[19] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. Pddl-the planning domain definition language. 1998.

[20] Jörg Hoffmann. Everything you always wanted to know about planning. In *Annual Conference on Artificial Intelligence*, pages 1–13. Springer, 2011.

[21] Edmund H Durfee and Victor R Lesser. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on systems, Man, and Cybernetics*, 21(5):1167–1183, 1991.

[22] Anders Jonsson and Michael Rovatsos. Scaling up multiagent planning: A best-response approach. In *ICAPS*, 2011.

[23] Raz Nissim and Ronen I Brafman. Multi-agent a* for parallel and distributed systems. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1265–1266. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[24] Mathijs De Weerdt, Adriaan Ter Mors, and Cees Witteveen. Multi-agent planning: An introduction to planning and coordination. In *In: Handouts of the European Agent Summer*. Citeseer, 2005.

[25] Stefano V Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *arXiv preprint arXiv:1709.08071*, 2017.

[26] Philip R Cohen, C Raymond Perrault, and James F Allen. Beyond question answering. *Strategies for natural language processing*, pages 245–274, 1981.

[27] Richard G Freedman and Shlomo Zilberstein. Integration of planning with recognition for responsive interaction using classical planners. In *AAAI*, pages 4581–4588, 2017.

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

[29] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. Robocup: The robot world cup initiative. In *Proceedings of the first international conference on Autonomous agents*, pages 340–347. ACM, 1997.

[30] Christopher F Argenta. *Recognizing Teams and Their Plans: General Plan Recognition in Multi-Agent Domains.* North Carolina State University, 2017.

[31] Robert Demolombe and Ana Mara Otermin Fernandez. Intention recognition in the situation calculus and probability theory frameworks. In *International Workshop on Computational Logic in Multi-Agent Systems*, pages 358–372. Springer, 2005.

[32] Stephen S Intille and Aaron F Bobick. A framework for recognizing multi-agent action from visual evidence. *AAAI/IAAI*, 99(518–525), 1999.

[33] Bikramjit Banerjee and Landon Kraemer. Branch and price for multi-agent plan recognition. In *AAAI*, 2011.

[34] Jerome Azarewicz, Glenn Fala, and Christof Heithecker. Template-based multi-agent plan recognition for tactical situation assessment. In *Artificial Intelligence Applications, 1989. Proceedings., Fifth Conference on*, pages 247–254. IEEE, 1989.

[35] Adam Sadilek and Henry A Kautz. Recognizing multi-agent activities from gps data. In *AAAI*, volume 39, page 109, 2010.

[36] Gita Sukthankar and Katia Sycara. Efficient plan recognition for dynamic multi-agent teams. In *Grace Hopper Women in Computer Science Conference*, pages 204–208, 2007.

[37] Dorit Avrahami-Zilberbrand and G Kaminka. Towards dynamic tracking of multi-agents teams: An initial report. In *Proceedings of the AAAI Workshop on Plan, Activity, and Intent Recognition (PAIR)*, pages 17–22, 2007.

[38] Gal A Kaminka, David V Pynadath, and Milind Tambe. Monitoring teams by overhearing: A multi-agent plan-recognition approach. *Journal of Artificial Intelligence Research*, 17(1):83–135, 2002.

[39] Hankz Hankui Zhuo and Lei Li. Multi-agent plan recognition with partial team traces and plan libraries. In *IJCAI*, volume 22, page 484, 2011.

[40] Chris Argenta and Jon Doyle. Probabilistic multi-agent plan recognition as planning (p-maprap): Recognizing teams, goals, and plans from action sequences. In *International Conference on Agents and Artificial Intelligence*, pages 575–582, 2017.

[41] Joseph A Barbera and Anthony Macintyre. Urban search and rescue. *Emergency Medicine Clinics*, 14(2):399–412, 1996.

[42] Jennifer Casper and Robin R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(3):367–385, 2003.

[43] Geoff Nagy and John Anderson. Active team management strategies for multi-robot teams in dangerous environments. In *Canadian Conference on Artificial Intelligence*, pages 385–396. Springer, 2017.

[44] Cade Earl Bartlett. *Communication between Teammates in Urban Search and Rescue*. Arizona State University, 2015.

[45] Kartik Talamadupula, Paul Schermerhorn, J Benton, Subbarao Kambhampati, and Matthias Scheutz. Planning for agents with changing goals. In *Twenty-First International Conference on Automated Planning and Scheduling: Proceedings of the System Demonstrations*, pages 71–74, 2011.

[46] David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526, 1978.

[47] Chris Frith and Uta Frith. Theory of mind. *Current Biology*, 15(17):R644–R645, 2005.

[48] Georg Henrik Von Wright. An essay in modal logic. 1951.

[49] Jaakko Hintikka. Knowledge and belief. an introduction to the logic of the two notions. 1965.

[50] Joseph Y Halpern. Using reasoning about knowledge to analyze distributed systems. *Annual Review of Computer Science*, 2(1):37–68, 1987.

[51] Andrés Perea. *Epistemic game theory: reasoning and choice.* Cambridge University Press, 2012.

[52] Peter Gärdenfors. *Knowledge in flux: Modeling the dynamics of epistemic states.* The MIT press, 1988.

[53] Jan van Eijck. Dynamic epistemic logics. In *Johan van Benthem on logic and information dynamics*, pages 175–202. Springer, 2014.

[54] Ronald Fagin, Joseph Y Halpern, Yoram Moses, and Moshe Vardi. *Reasoning about knowledge.* MIT press, 2004.

[55] Raymond Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems.* MIT Press, 2001.

[56] Chitta Baral, Thomas Bolander, Hans van Ditmarsch, and Sheila McIlrath. Epistemic planning (dagstuhl seminar 17231). In *Dagstuhl Reports*, volume 7. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[57] Xiao Huang, Biqing Fang, Hai Wan, and Yongmei Liu. A general multi-agent epistemic planner based on higher-order belief change. In *IJCAI 2017, Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, Sydney*, 2017.

[58] Son Thanh To, Tran Cao Son, and Enrico Pontelli. Contingent planning as and/or forward search with disjunctive representation. In *ICAPS*, 2011.

[59] Tiep Le, Francesco Fabiano, Tran Cao Son, and Enrico Pontelli. Efp and pg-efp: Epistemic forward search planners in multi-agent domains. In *Proceedings of the Twenty-Eigth International Conference on Automated Planning and Scheduling (ICAPS 2018).* AAAI Press, 2018. To appear.

[60] Filippos Kominis and Hector Geffner. Beliefs in multiagent planning: From one agent to many. In *ICAPS*, pages 147–155, 2015.

[61] Thorsten Engesser, Thomas Bolander, Robert Mattmüller, and Bernhard Nebel. Cooperative epistemic multi-agent planning for implicit coordination. *arXiv preprint arXiv:1703.02196*, 2017.

[62] Christian Muise, Frank Dignum, Paolo Felli, Tim Miller, Adrian R Pearce, and Liz Sonenberg. Towards team formation via automated planning. In *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*, pages 282–299. Springer, 2015.

[63] Guillaume Aucher and Thomas Bolander. Undecidability in epistemic planning. In *IJCAI-International Joint Conference in Artificial Intelligence-2013*, 2013.

[64] Thomas Bolander, Martin Holm Jensen, Francois Schwarzentruber, and ENS Rennes. Complexity results in epistemic planning. In *IJCAI*, pages 2791–2797, 2015.

[65] Gerhard Lakemeyer and Yves Lespérance. Efficient reasoning in multiagent epistemic logics. In *ECAI*, pages 498–503, 2012.

[66] Miquel Ramírez and Hector Geffner. Goal recognition over pomdps: Inferring the intention of a POMDP agent. In *IJCAI*, pages 2009–2014, 2011.

[67] Ramon Fraga Pereira and Felipe Meneguzzi. Heuristic approaches for goal recognition in incomplete domain models. *arXiv preprint arXiv:1804.05917*, 2018.

[68] Miquel Ramírez and Hector Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *AAAI*, 2010.

[69] 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 Int'l Conference on Automated Planning and Scheduling (ICAPS)*, pages 10–16, 2014.

[70] Sarath Sreedharan, Tathagata Chakraborti, and Subbarao Kambhampati. Explanations as model reconciliation-a multi-agent perspective. 2017.

[71] Michael Katz, Shirin Sohrabi, Octavian Udrea, and Dominik Winterer. A novel iterative approach to top-k planning. In *Proceedings of the Twenty-Eigth International Conference on Automated Planning and Scheduling (ICAPS 2018)*. AAAI Press, 2018. To appear.

[72] Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

[73] Maayan Shvo, Shirin Sohrabi, and Sheila A. McIlraith. An ai planning-based approach to the multi-agent plan recognition problem (preliminary report). In *Proceedings of the The AAAI 2017 Workshop on Plan, Activity, and Intent Recognition (PAIR)*, 2017.

[74] Maayan Shvo, Shirin Sohrabi, and Sheila A McIlraith. An ai planning-based approach to the multi-agent plan recognition problem. In *Advances in Artificial Intelligence: 31st Canadian Conference on Artificial Intelligence, Canadian AI 2018, Toronto, ON, Canada, May 8–11, 2018, Proceedings 31*, pages 253–258. Springer, 2018.

[75] Maria Fox and Derek Long. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *JAIR*, 20:61–124, 2003.

[76] William Cushing, Subbarao Kambhampati, Daniel S Weld, et al. When is temporal planning really temporal? In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1852–1859. Morgan Kaufmann Publishers Inc., 2007.

[77] Ronen I. Brafman and Carmel Domshlak. From one to many: Planning for loosely coupled multi-agent systems. In *ICAPS*, pages 28–35, 2008.

[78] Matthew Crosby, Anders Jonsson, and Michael Rovatsos. A single-agent approach to multiagent planning. In *ECAI*, pages 237–242, 2014.

[79] Christian Muise, Nir Lipovetzky, and Miquel Ramirez. MAP-LAPKT: Omnipotent multi-agent planning via compilation to classical planning. In *Competition of Distributed and Multiagent Planners*, 2015.

[80] Patrik Haslum and Alban Grastien. Diagnosis as planning: Two case studies. In *SPARK*, pages 27–44, 2011.

[81] Sarah Keren, Avigdor Gal, and Erez Karpas. Privacy preserving plans in partially observable environments. In *IJCAI*, 2016.

[82] Daniel Bryce. Landmark-based plan distance measures for diverse planning. In *ICAPS*, pages 56–64, 2014.

[83] Mark Roberts, Adele E. Howe, and Indrajit Ray. Evaluating diversity in classical planning. In *ICAPS*, pages 253–261, 2014.

[84] Biplav Srivastava, Tuan Anh Nguyen, Alfonso Gerevini, Subbarao Kambhampati, Minh Binh Do, and Ivan Serina. Domain independent approaches for finding diverse plans. In *IJCAI*, pages 2016–2022, 2007.

[85] Alexandra Coman and Hector Muñoz-Avila. Generating diverse plans using quantitative and qualitative plan distance metrics. In *AAAI*, pages 946–951, 2011.

[86] 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. *AIJ*, 190:1–31, 2012.

[87] Alfonso Gerevini, Alessandro Saetti, and Ivan Serina. Planning through stochastic local search and temporal action graphs in LPG. *JAIR*, 20:239–290, 2003.

[88] Alfonso Gerevini, Alessandro Saetti, and Ivan Serina. LPG-TD: a fully automated planner for PDDL 2.2 domains. In *14th Int'l Conference on Automated Planning and Scheduling, system demo session (ICAPS-04)*, 2004.

[89] Amanda Coles, Andrew Coles, Maria Fox, and Derek Long. Forward-chaining partial-order planning. In *ICAPS*, pages 42–49, 2010.

[90] Yolanda E-Martín, María D. R-Moreno, and David E. Smith. A fast goal recognition technique based on interaction estimates. In *IJCAI*, pages 761–768, 2015.

[91] Sergio Jiménez Celorrio, Anders Jonsson, and Héctor Palacios. Temporal planning with required concurrency using classical planning. In *ICAPS*, pages 129–137, 2015.

[92] Jussi Rintanen et al. Complexity of concurrent temporal planning. In *ICAPS*, pages 280–287, 2007.

[93] Tom Bylander. The computational complexity of propositional strips planning. *Artificial Intelligence*, 69(1-2):165–204, 1994.

[94] Yu Zhang, Sarath Sreedharan, and Subbarao Kambhampati. A formal analysis of required cooperation in multi-agent planning. In *ICAPS*, pages 335–344, 2016.