Planning and Goal Recognition in Humans and Machines

by

Chenyuan Zhang

ORCID: 0009-0007-4860-0696

A thesis submitted in total fulfillment for the
degree of Doctor of Philosophy

in the
School of Computing and Information Systems
Faculty of Engineering and Information Technology
THE UNIVERSITY OF MELBOURNE

May 2024

THE UNIVERSITY OF MELBOURNE

Abstract

School of Computing and Information Systems
Faculty of Engineering and Information Technology

Doctor of Philosophy

by Chenyuan Zhang
ORCID: 0009-0007-4860-0696

The rapid advancement of artificial intelligence, exemplified by systems such as AlphaGo and large language models, has great potential to contribute to the development of human-like intelligence. However, fundamental differences exist between the underlying mechanisms of these systems and those of biological organisms. For instance, humans can achieve impressive performance with limited data and computing resources, while existing algorithms often require significant amounts of data and computing power for real-time operations. One of the reasons for this disparity is the human ability to plan in a model-based sense, making computational models that can capture human planning behavior valuable to bridge the gap between existing AI systems and human-like intelligence.

This thesis explores the effectiveness of planning algorithms in modeling human behavior. Existing literature often overlooks timing information, and I develop a novel tree-based model that aims to capture both human action selection and human reaction times. The thesis also introduces a timing-sensitive goal recognition framework that incorporates timing information, and uses this framework to model human goal inference. My findings indicate that a Bayesian framework that incorporates a prior based on goal difficulty and a likelihood derived from an online planner accurately predicts human goal inference. This thesis underscores the promise of planning algorithms in mimicking human behavior and their utility in human-robot collaborations. More generally, it suggests that planning algorithms have an important role to play in advancing human-like intelligence.

Declaration of Authorship

I, Chenyuan Zhang, declare that this thesis titled, *Planning and Goal Recognition in Human and Machine* and the work presented in it are my own. I confirm that:

- The thesis comprises only my original work towards the degree of Doctor of Philosophy except where indicated in the preface;
- due acknowledgement has been made in the text to all other material used; and
- the thesis is fewer than the 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Chenyuan Zhang

Dec 2023

Preface

This thesis is submitted in total fulfilment of the requirements for the degree of Doctor of Philosophy at the University of Melbourne. The research presented here was primarily conducted at the School of Computing and Information Systems, The University of Melbourne, under the supervision of Dr. Nir Lipovetzky and Prof. Charles Kemp.

Below is the list of publications and manuscripts arising from this thesis. I was the principal author of all papers and contributed more than 50% on each paper. I was responsible for designing the algorithm's architectures, collecting data-sets, implementation, running experiments and analysing the experimental results. My co-authors provided feedback on the proposed algorithms and models and contributed to the revisions of the manuscripts. Ethics approval to conduct the studies comprising this thesis was provided by The University of Melbourne's human ethics committee (Chapters 3 - 5, ID: 2057080.1).

- Part of the contents of Chapter 3 has been published in the following paper: Zhang, C., Lipovetzky, N., & Kemp, C. (2023). Comparing AI Planning Algorithms with Humans on the Tower of London Task. In Proceedings of the Annual Meeting of the Cognitive Science Society (Vol. 45, No. 45).
- Part of the contents of Chapter 4 has been published in the following paper: Zhang, C., Kemp, C., & Lipovetzky, N. (2023). Goal Recognition with Timing Information. Proceedings of the International Conference on Automated Planning and Scheduling, 33(1), 443-451.
- Part of the contents of Chapter 5 has been accepted by the 23rd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2024).

I gratefully acknowledge Melbourne Research Scholarship for funding.

Acknowledgements

This journey has been both challenging and fulfilling, and it could not have been achieved without the support of numerous individuals.

First and foremost, my heartfelt thanks go to my supervisors, Professor Nir Lipovetzky and Professor Charles Kemp. I am deeply appreciative of your support, guidance, and positivity over these years. I cannot imagine better supervisors than you. You have always provided unwavering support, both emotionally and academically, without ever pressuring me. You've guided me in understanding the intricacies of high-quality research and in becoming a proficient researcher, all while maintaining a healthy work-life balance. Nir, you have been an exceptional supervisor throughout my master's degree, consistently providing reliable assistance with coding and writing whenever I encountered challenges. Your teaching in the subject of automated planning has opened the door to the world of planning for me, and it will always remain my favorite subject. Charles, as I've mentioned before, you are the most intelligent individual I have encountered at the University of Melbourne. Your enthusiasm for academia, curiosity about complex problems, and your ability to identify, organize, and frame research questions have all provided me with an exemplary template of what an outstanding researcher embodies. I feel incredibly fortunate to have you as my supervisor.

I have been blessed to discover friendship and camaraderie among many outstanding PhD scholars. My heartfelt thanks to each of you for the laughter, engaging discussions, and wonderful companionship. I would also like to extend my sincere thanks to Lianglu, Guang, Ruihan and Yujing. The time spent with you has always been both enjoyable and relaxing. I'm also grateful for your invaluable assistance with various personal matters, such as helping me move homes, car servicing, and more.

I would like to express my heartfelt gratitude to my friends outside the university. Special thanks go to Karen and the members of our boardgame groups, who have brought a delightful spice to my PhD life. Additionally, I am grateful to my friend Pingping; our conversations about music and past lives have profoundly connected me to my roots and heritage. Lastly, I want to acknowledge the friends from the Wuyong Book Club. Although my participation was limited, the moments spent with you have always been relaxing, and I always look forward to them.

I am deeply thankful for the support of my partner Qiu. Your cooking always brightens my life, and your adeptness at installing household appliances superbly balances my own shortcomings. I also want to express my thanks to my dog, Ginger, and my cat, Misty. Although you both cause your fair share of mischief, the joy of your companionship

makes it all worthwhile. I also owe a tremendous thank you to my parents, Jizhong and Zhuo, for providing both financial and emotional support throughout this challenging journey. You are the ones who have given me the opportunity to truly be myself.

Thank you!

Chenyuan Zhang

Contents

List of Figures

List of Tables

Abbreviations

AI	**A**rtificial **I**ntelligence
A-LH	**A**daptive **L**ook**a**head Planner
BIC	**B**ayesian **I**nformation **C**riterion
PDDL	**P**lanning **D**omain **D**escription **L**anguage
PRP	**P**lan **R**ecognition as **P**lanning
MDP	**M**arkov **D**ecision **P**rocess

Chapter 1

Introduction

Imagine a scenario where a robot collaborates with a human worker in a warehouse environment. Together, their objective is to pick items from shelves and carefully place them into designated containers for shipping. Now, consider a situation where the robot must decide which item to pick next, taking into account specific order requirements and packaging constraints. To execute this task effectively, the robot must grasp the human worker's intentions, preferences, and reasoning, enabling it to provide valuable assistance.

If the robot does not understand the human worker and just follows a predetermined set of rules, it may make incorrect assumptions that do not align with the human's goals, which might lead to inefficiencies, errors, and frustration for both the robot and the human [1]. For example, the robot might choose an item that the human worker had intentionally skipped because it was damaged, resulting in wasted time and resources. However, if the robot can comprehend the human worker's decision-making process by taking into account factors such as the human's actions, body language, verbal instructions, thinking time or past behaviors, it can adapt its actions to support and complement the human's intentions. In this case, the robot might recognize that the human worker skipped a particular item due to damage and choose an alternative, saving time and ensuring accurate order fulfillment. As this example suggests, by having a better understanding of humans in complex environments, robots can effectively collaborate with humans, enhance their capabilities, and provide valuable assistance, ultimately improving the overall efficiency and productivity of human-robot collaboration [1, 2].

FIGURE 1.1: David Vogt. A human working together with a robotic arm. Accessed 11 November 2023, https://research.engineering.asu.edu/exploring-new-frontiers-in-human-robot-collaborations

This particular example can be seen as an instance of goal recognition, where the observer infers the goal of the actor based on observed behavior. The goal recognition problem can be considered as one application of *Theory of Mind* in cognitive science (CogSci), which refers to the ability to attribute mental states——such as beliefs, desires, intentions, and emotions-to others in order to understand and predict behavior [3, 4]. Meanwhile, certain researchers argue that goal inference may not inherently employ Theory of Mind [5–7], suggesting that inferences could be drawn without a direct mental representation of the actor. While this perspective introduces an interesting alternative for further exploration, this thesis does not discuss the controversy extensively. Instead, my research will be based on the assumption that Theory of Mind plays a role in the process of goal recognition, following prior work on Plan recognition as Planning [8] and Bayesian Theory of Mind [9] streams within the literature. This assumption is especially likely to be valid when goal inference is conducted on complex tasks. Therefore, the thesis will focus on goal recognition within sequential decision-making tasks that likely necessitate explicit reasoning. I will discuss the opposite perspective (i.e. reflexive reasoning) with more details in Chapter 6.

FIGURE 1.2: TA self-driving Cruise robotaxi inconveniencing pedestrians at a crossroads in San Francisco in 2019. Andrej Sokolow. Accessed 11 November 2023, https://spectrum.ieee.org/self-driving-cars-2662494269

In the field of AI, current research on human-agent interaction focuses on improving human understanding of algorithms and models through explanations, known as Explainable AI [10]. However, the reciprocal aspect, wherein algorithms comprehend the internal mechanisms of human behavior and decision-making processes to enhance the effectiveness of the AI system, often receives less attention within the community [11]. Model reconciliation [12, 13] and transparent planning [14] represent two efforts in this direction, although they proceed with assumed human models and without the validation of human experiments. Nonetheless, considering precise human mental models is crucial in complex environments, not only empowering agents to autonomously fine-tune their behavior and enhance support, but also inspiring the development of more efficient algorithms since humans continue to outperform agents in numerous scenarios [15]. This approach holds particular significance in two kinds of goal recognition scenarios. Firstly, when the actor is human and the observer is an AI agent, the agent should factor in human considerations to enhance its inference of the human actor's goals. For instance, in Figure 1.1, the robotic arm can offer more efficient and targeted assistance by understanding human intentions. Secondly, in situations involving a human observer and an

AI actor, the AI actor should account for how humans conduct goal inference, adapting its behavior to effectively convey its intentions to humans in collaborative settings. As shown in Figure 1.2, pedestrians may experience frustration if the self-driving car does not communicate its intentions in a manner that is easily understood by humans.

Integrating human factors into AI systems within the context of human-agent interaction involves a three-step approach [16]. First, the design and execution of human experiments are essential to gather behavioral data within a defined domain and task. Second, this collected data serves as the foundation for developing an algorithm that emulates human decision-making in complex environments—a so-called 'human-like model.' This algorithm operates at an abstract level, encapsulating the core principles of human decision processes. Finally, the human-like algorithm is harnessed to enhance the efficiency and performance of the AI systems.

While I employ planning techniques to develop domain-general algorithms, this does not imply that a unified single planner can encompass all scenarios. As discussed in Chapter 6, different scenarios may require adjustments to parameters or the use of alternative components within planners. In the first two chapters, I highlight our aim to avoid relying on specific knowledge about solving particular types of problems. Instead, we focus on modeling legal transitions that represent broader principles or strategies of human problem-solving. This is a key reason for employing planning techniques in our study. Meanwhile, I also acknowledge the importance of incorporating individual or condition-specific factors during the modelling, as presented in Chapter 3, 4 and 5.

1.1 Research Contribution

This thesis will consider two primary topics: planning and goal recognition, with the actor or observer being either a human or an agent, as illustrated in 1.1. The ultimate goal of this line of work is to support human-agent interactions by furnishing agents with user-friendly, human-like algorithms. Additionally, I will introduce a framework for integrating these algorithms into goal recognition tasks in scenarios involving human participants.

The field of artificial intelligence has seen limited efforts in generating human-like responses rather than optimal responses. This task presents additional challenges due to

		Actor	
		Human	**Agent**
Domain: Planning		• Chapters 3&5	• Chapters 3&4
Domain: Goal Recognition	**Human**	—	• Chapters 4&5
	Observer **Agent**	• Chapters 4&5	• Chapter 4

TABLE 1.1: The topics addressed in Chapters 3, 4, and 5. In Chapter 3, I explore human planning behavior and assess the capacity of planning models to simulate it. Chapter 4 centers on goal recognition scenarios, encompassing interactions between human and agent actors. Chapter 5 focuses on the computational mechanisms of human goal recognition, while also examining the performance of planning models when confronted with unsolvable goals.

the inherent variability and instability in human responses across multiple levels. Firstly, individual differences contribute to distinct responses among different individuals. Secondly, the same person may exhibit different responses based on changes in their internal state, such as increased experience or cognitive resources. Furthermore, even when an individual retains a consistent internal state, variations persist, emphasizing the dynamic nature of experiences and decisions. As a result, predicting individual responses with certainty becomes challenging, if not impossible. However, it is feasible to describe human responses through population-level probabilistic distributions, indicating typical behavior for a given scenario [17]. When addressing sequential decision-making tasks, generating human-like responses becomes even more challenging. The complexity of the problem's state space and transition function impedes learning from the perspective of a human decision-maker. Moreover, the intricate causal relationships among sequences of behaviors further complicate the task [18]. In this thesis, I will concentrate on domains involving sequential decision-making, also known as problem solving in the field of CogSci.

In addition, existing computational models of human sequential decision making, either in the AI or CogSci communities, have primarily focused on the actions themselves,

disregarding the significance of reaction time. Nonetheless, the accessibility of timing information in the real world makes it an invaluable asset for building computational models, as response times can be easily captured and measured. Moreover, when considering goal recognition algorithms, timing information can significantly enhance their accuracy (as discussed in Chapter 4), providing additional contextual cues that sharpen the inferences of an actor's goals. Last but not least, Incorporating timing information allows goal recognition algorithms to better align with human intuition and behavior, resulting in more relatable and human-like inferences [19].

In light of this, my thesis aims to develop a human-like sequential decision making model by capturing both the actions and the reaction times. Through considering timing as a key variable, the model aspires to mirror the cognitive mechanism of how people make decisions, offering a more comprehensive and authentic representation of human behavior. This innovative step stands to refine our approach to intelligent system design, ensuring that such systems are better equipped to interact with humans in a manner that feels more natural and effective.

I begin by evaluating planning algorithms to determine their effectiveness in predicting human responses within sequential decision-making contexts, covering both action selection and reaction time. Following this assessment, I use the best-performing algorithm to develop models of goal recognition, with the aim of enhancing AI observer performance. This assessment involves conducting synthetic experiments using an AI actor on goal recognition benchmarks, as well as human experiments involving human actors. Finally, my investigation extends to scenarios involving unsolvable goals, which pose unique challenges for AI observers. Within this context, I explore the factors that influence human goal inference and develop a Bayesian model of human goal recognition behavior.

1.1.1 Research Questions

This thesis addresses the following research questions, which serve as the foundation for my work.

RQ1: Which algorithm is most suitable for emulating human responses (both action selection and response times) in sequential decision-making tasks?

RQ2: How can a human-like planning algorithm be leveraged to enhance the performance of methods for goal recognition?

RQ3: How do humans carry out goal inference, and can this inference be captured within a Bayesian framework?

1.2 Applying a Human-like Problem Solving Model to the Tower of London Task

To address **RQ1**, I adopt the automated planning approach [20, 21], specifically using automated planning algorithms to model human behavior in the Tower of London task. Several reasons support my choice of the automated planning approach over learning approaches or computational models in CogSci. Firstly, the automated planning approach shares similarities with the General Problem Solver (GPS) proposed by Newell and Simon [22]. It offers greater explainability compared to learning approaches and contributes to a better understanding of consciously planning processes in the human mind [15]. Secondly, researchers have endeavored to create models that generalize across various domains, such as cognitive architectures in cognitive science and approaches like transfer learning in AI. Despite these efforts, such models often still rely on some degree of domain-specific knowledge or rules [23–25]. In contrast, the automated planning approach is designed to be domain-independent. This aligns with a key characteristic of human problem-solving, where individuals can typically perform well without explicit training or precise knowledge about the task at hand [26]. Lastly, automated planning algorithms offer flexibility in integrating various findings from Cognitive Science. For example, learning effects can be modeled as the development of more accurate heuristic functions [27], while the trade-off between accuracy and speed can be modeled as the depth of the search tree [28]. Overall, the adoption of the automated planning approach provides a robust framework for exploring human behavior in sequential decision-making tasks and enables the incorporation of relevant insights from both CogSci and AI.

1.2.1 Tower of London Task

I selected the Tower of London task (see Figure 1.3) as the focus domain for studying human problem-solving due to two primary reasons. Firstly, while numerous researchers are

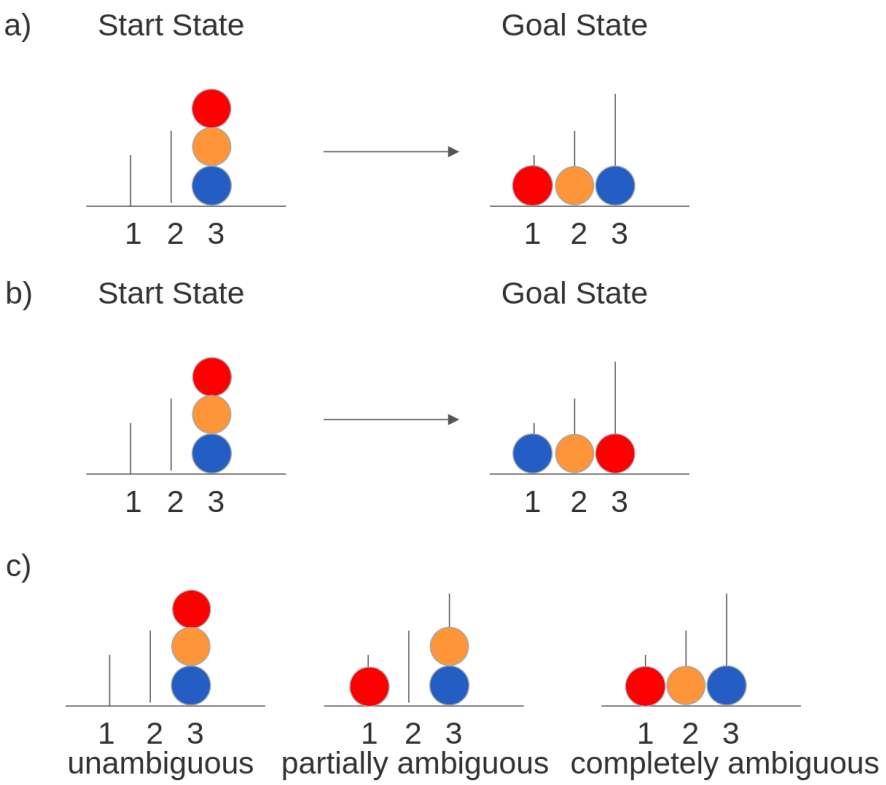


FIGURE 1.3: Tower of London task. (a) A problem instance that requires two moves to transition from the start state to the goal state. (b) A problem instance requiring five moves. (c) *Start Hierarchy* is a structural parameter that classifies each instance as unambiguous (all balls on one peg), partially ambiguous or completely ambiguous (all balls on different pegs). The "ambiguity" refers to the initial action: unambiguous actions allow only one action, but completely ambiguous instances allow 4 possible actions.

increasingly directing their attention towards non-deterministic or partially observable environment tasks, my work aligns with the Newell and Simon tradition, which investigates insights from human performance on deterministic, fully observable tasks, such as Tower of London. The task shares similarities with the well-known Tower of Hanoi task, and both tasks have been extensively studied by psychologists [29–33]. Moreover, the Tower of London task bears resemblance to the classic planning domain *Blockwords* [34], making it an ideal domain for leveraging knowledge from both the psychological and planning communities.

In addition, the Tower of London task specifically requires participants to engage in explicit reasoning rather than relying on reflexive behavior [15] and individuals often demonstrate the ability to solve this task successfully, even when they are new to it [35]. This characteristic highlights the task's suitability for examining human problem-solving capabilities and the cognitive processes involved in tackling complex reasoning

challenges.

Figure 1.3a shows an instance of the TOL task. The board shown has pegs that can hold 1, 2 and 3 balls respectively from left to right. Participants are given the board in some initial state, then asked to move balls from peg to peg until the board matches some specified goal state. The participant can move the topmost ball from each peg to another peg, and needs to ensure that the move does not violate the maximum capacity of any peg. The instance in Figure 1.3a can be solved in just two moves, but the shortest solution of the instance in Figure 1.3b involves five moves. The previous literature has identified several key features within the TOL task that influence human performance [29, 36]. For example, figure 1.3c illustrates the concept of *Start Hierarchy*, which is one of the structural parameters analyzed by Berg et al. [36].

1.3 Using a Human-Like Planning Algorithm for Goal Recognition

RQ2 builds upon the human-like planning algorithm developed in the previous question. Existing approaches in goal recognition often assume actor rationality [37, 38], which does not hold true for human behavior. Additionally, most goal recognition algorithms consider only actions as observations, overlooking the important aspect of timing. I overcome both limitations by building on the human-like model developed in addressing **RQ1**. I propose an extended framework that incorporates both action and timing, and I construct a timing goal recognition dataset based on an existing goal recognition benchmark [39].

Furthermore, I develop a goal recognition algorithm using the human-like planning algorithm. To evaluate its effectiveness, I conduct experiments on both synthetic datasets spanning 10 different domains and human experiments using the Sokoban domain. The results demonstrate that my algorithm outperforms existing approaches in capturing human behaviors, while also exhibiting similarities to human goal inference. This provides evidence of the algorithm's superiority in modeling and understanding human decision-making processes.

Regarding **RQ3**, my research centers on the challenge of recognizing goals when faced with unsolvable scenarios, a problem that existing goal recognition algorithms struggle

to address. To explore how humans address this challenge, I develop a set of goal-recognition experiments involving unsolvable goals. In parallel, I evaluate a Bayesian framework as a model of human goal inference, leading to the development of a goal recognition algorithm designed to mimic human inference and handle unsolvable goals. My research goes beyond just actions, adding new dimensions to the Bayesian Theory of Mind and establishing the correlation between human problem solving and goal recognition performance.

1.3.1 Sokoban Task

As part of my investigation in cognitive problem-solving and human-agent interactions for **RQ2** and **RQ3**, we turn our attention to the Sokoban task as shown in Figure 1.4, a classic puzzle game originating from Japan. This game has been widely studied and serves as a benchmark domain in the field of artificial intelligence [40]. In Sokoban, the player assumes the role of a warehouse worker who must strategically maneuver crates to their designated storage locations within a confined warehouse. The game presents a complex set of challenges that require logical reasoning, planning, and spatial awareness. The objective of Sokoban is to successfully push all the crates onto the target locations while avoiding obstacles and creating blockades that may render the puzzle unsolvable. The game mechanics impose strict constraints on the movement of both the player and the crates, adding an extra layer of intricacy to the puzzle-solving process.

Sokoban puzzles are characterized by their increasing levels of difficulty, ranging from simple introductory stages to highly complex configurations that demand advanced problem-solving skills [40]. The game's popularity stems from its ability to engage players in critical thinking, strategic planning, and pattern recognition, making it an ideal domain for studying various aspects of cognitive processes and algorithmic approaches [41, 42].

In the academic community, Sokoban has been extensively used to evaluate the performance of intelligent systems, including automated planning algorithms [37, 39], heuristic search techniques [40, 43], and machine learning models [44, 45]. Researchers have leveraged Sokoban's well-defined problem space and clear success criteria to assess the effectiveness of novel algorithms and methodologies in the domain of puzzle solving.

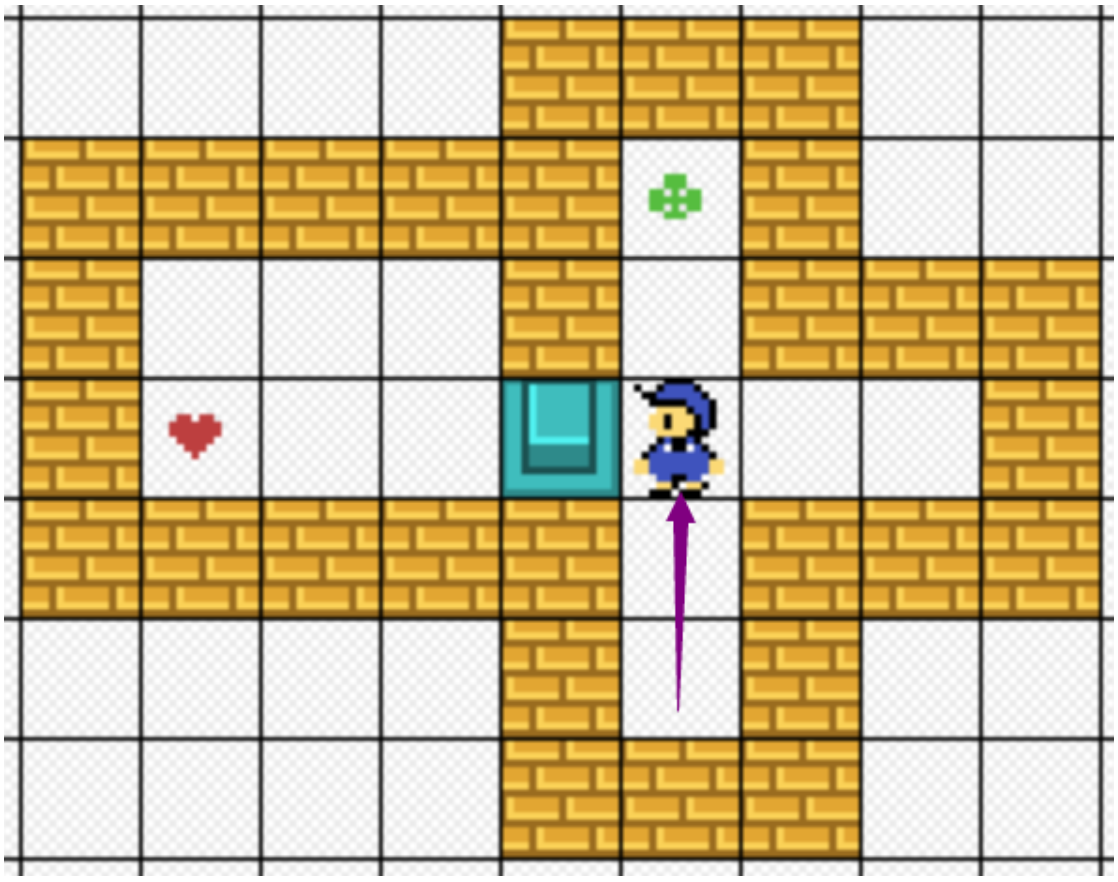


FIGURE 1.4: A Sokoban instance. In the problem solving task, the worker needs to the crates to the target location (marked in red). In the goal recognition task, multiple potential locations (in red and green) are shown, requiring the observer to infer which is the real target location the worker is aiming for.

In the context of human goal recognition experiments, Sokoban task offer a convenient framework for manipulating and presenting different configurations of goal positions. This flexibility enables us to systematically investigate and analyze human goal recognition abilities in a controlled experimental setting. In order to streamline my analysis and maintain an appropriate level of difficulty for human observers, I deliberately limit my investigation to Sokoban tasks featuring a single crate. This approach ensures that the task remains accessible and comprehensible while enabling us to extract meaningful insights into human goal inference processes.

1.4 Thesis Outline

The subsequent sections of this thesis are structured as follows. Chapter 2 provides an extensive review of relevant literature, encompassing previous work on Theory of

Mind, Goal Recognition, Human Problem Solving Models etc. This chapter culminates with a brief discussion on how these collective findings contribute to the construction of the human-like problem-solving model and goal recognition algorithm. By critically examining and synthesizing existing research, I establish a comprehensive foundation for the subsequent chapters of this thesis.

Chapter 3 presents novel human-like planning algorithm, which directly addresses the first research question (**RQ1**). The chapter explores the model's design choices, and explains its underlying architecture and mechanisms. Furthermore, an empirical evaluation is conducted to assess the performance of the proposed algorithm on the Tower of London (TOL) task, comparing it against existing algorithms. The results obtained highlight the effectiveness and superiority of the introduced algorithm in achieving more accurate and human-like problem-solving capabilities. This work previously appeared at the 44th Annual Meeting of the Cognitive Science Society (CogSci23).

Chapter 4 presents a framework for goal recognition with timing information, which leverages the proposed human-like planning algorithm (**RQ2**). This chapter outlines the integration of the algorithm into the goal recognition task, showcasing its effectiveness and advantages. To validate its performance, extensive experiments are conducted using synthetic datasets as well as human behavioral data on the Sokoban task. The results demonstrate the superior performance of the algorithm in accurately recognizing goals, both in synthetic scenarios and when applied to human behavior. This chapter illustrate how the human-like model can be used to enhance the goal recognition capabilities of AI agents. This work has been published in the 33rd International Conference on Automated Planning and Scheduling (ICAPS23).

Chapter 5 continues the exploration from Chapter 4, where I look closely at human goal inference with unsolvable goals in the Sokoban task. In this chapter, I aim to understand the challenges posed by these situations and how humans handle unsolvable goals. I also explore a Bayesian goal recognition framework in depth and create a goal recognition algorithm that mimics how humans infer goals in dynamic and complex environments(**RQ3**). This work has been accepted by the 23rd International Conference on Autonomous Agents and Multi-agent Systems (AAMAS24).

Chapter 6 discusses potential future directions in the field. This chapter explore the possibilities of integrating deep learning approaches into the domain, highlighting other

important challenges that remained unsolved within this thesis, such as working memory mechanism and human goal recognition mechanism. In Chapter 7, the thesis concludes with a summary of the contributions made in the realm of human-like AI and human-agent collaboration. This chapter provides a comprehensive overview of the key findings and advancements achieved throughout the thesis, arguing for the significance of the work in advancing our understanding of human behavior and enhancing interactions between humans and intelligent agents.

Chapter 2

Background

Problem solving and goal/intent recognition are both important activities in our daily life and there is a large literature spanning multiple disciplines on both topics. This chapter aims to provide an overview of how the AI community and the cognitive science community approach these topics from different perspectives.

The chapter is divided into two main sections: problem solving and goal recognition. In the problem-solving part, I begin by providing an overview of human problem-solving behavior and associated computational models. I then examine how the AI community represents and models problems. Next, I introduce relevant planning algorithms used in automated planning to solve these problems. Finally, I briefly introduce the primary domains typically studied in problem-solving research. In the goal recognition section, I will first discuss work on *Theory of Mind* and other investigations related to how humans understand others. I then discuss AI research in this area, including categorizing various goal recognition approaches. The discussion end with a detailed exploration of the model-based approach.

2.1 Problem Solving

Problem solving is a term with various definitions. It can refer to cognitive processing directed at a goal when the solution method is not initially known [46]. It can also mean engaging in a task for which the solution method is not known in advance [47].

In many scenarios, particularly in the realm of artificial intelligence and operations research, problem solving extends beyond finding a single response. Instead, it requires identifying a sequence of actions or observations to achieve a desired goal. Classic puzzles like the 8-queen problem exemplify this form of sequential decision-making. In this puzzle, one needs to place eight queens on an 8x8 chessboard such that no two queens threaten each other. Deciding where to place each queen is a sequence of decisions, and each decision influences the subsequent choices. It is worth noting that there are various approaches to solve this type of puzzle, and not all of them rely strictly on sequential decision-making. Some algorithmic solutions might employ backtracking, genetic algorithms, or other combinatorial techniques. However, in this thesis, I will focus on problem-solving as finding the solution to a well-defined problem as a sequential decision making task.

In academia, problem solving or sequential decision making is considered as a hallmark of intelligent behavior, and efforts to develop problem solving agents have been pursued by both AI researchers [48] and cognitive scientists [49] since the development of the Logic Theorist in 1956, a theorem prover sometimes described as the first AI program [50]. These two communities, however, focus on different themes: psychologists are interested in the cognitive processes behind human problem solving, and AI researchers focus on developing efficient algorithms that generate optimal or approximately optimal solutions.

2.1.1 Computational Models for Human Problem Solving

Researchers have looked at how humans tackle a variety of challenges, from puzzles like the Tower of Hanoi [51] to more complex problems like the Travelling Salesman Problem [52]. While many models have been created to mimic how humans behave in these tasks, these models are often specific to one task and don't work as well for others. Interestingly, humans seem to quickly understand and make good decisions in new situations or games after just learning the rules. This suggests that humans might use general strategies that work across many tasks, instead of needing specific strategies for each one [53, 54]. The General Problem Solver (GPS) was one of the first attempts to model human problem-solving in a domain-general manner, laying the foundation for numerous subsequent cognitive models. [22]. At its core, the GPS operates via means-ends analysis. It starts by comparing the current state with the desired goal state and

then identifies differences between the two. The system then selects an operator or action that can reduce this difference. If the selected action cannot be directly applied, the system sets a sub-goal to create conditions that enable the action. This process of setting sub-goals continues recursively until the system can apply an action directly. Throughout this iterative process, the GPS continually makes decisions about which differences to focus on and which operators to apply, navigating its way from the initial state to the goal state.

Later, several general cognitive architectures like BDI [55], SOAR [56] and ACT-R [57] were developed. These architectures are intended to capture various human cognitive mechanism, including problem solving, attention, memory etc. However, for a specific task, they usually include some domain-specific production rules.

SOAR is a cognitive architecture designed to mimic general intelligence and human cognition across a wide range of tasks. In the context of problem-solving, SOAR employs a decision-making process called the recognize-decide-act cycle. Initially, the system perceives its environment and forms internal symbolic representations. When faced with a problem, SOAR first attempts to retrieve a relevant solution from its memory. If a past solution isn't available or doesn't fit the current problem, the system enters a problem-solving mode. In this mode, SOAR generates and tests various hypotheses or strategies using chunking, a mechanism to store new knowledge. This iterative process of hypothesis generation and testing continues until a viable solution is found. Once the solution is identified, SOAR acts upon it and stores this new knowledge for future reference. The system's ability to learn from past experiences and adapt its strategies is a core feature that allows it to tackle a diverse array of problems, though it does often rely on domain-specific knowledge to enhance its performance, especially in hypothesis or strategies generation stage.

Over the past two decades, significant advancements have been made in modeling human domain-general problem solving behaviors, drawing from diverse computational approaches. One of the most notable advancements in modeling human problem-solving behaviors comes from the realm of artificial intelligence (AI) and machine learning. A key method in this area uses tree search algorithms to simulate human decision-making [58, 59]. Tree search involves expanding a tree of possibilities, where each node represents a state, and the edges between nodes depict actions or transitions from one

state to another. Starting from an initial state (root of the tree), the algorithm explores potential actions and their outcomes, branching out to form a tree structure. As the tree expands, decision-making relies on evaluating the desirability of states, often guided by heuristics or domain knowledge. A more sophisticated approach, Monte Carlo Tree Search (MCTS), employs random sampling combined with traditional tree search [60, 61]. The ultimate goal of tree search is to find a path through the tree, from the initial state to a goal state, that represents a solution to the problem at hand. These methods mirror the way humans might mentally map out the consequences of their actions when faced with complex challenges. Their efficacy becomes especially evident in constrained environments, such as board games, where they have managed not only to match but surpass human expertise in some instances [62].

Another paradigm known as resource-rational analysis has been gathering attention in the cognitive science community [63–65]. This framework presents a fresh perspective on human decision making and problem solving by assuming that human cognition is adept at deploying its limited computational resources in an optimal manner. At its heart, resource-rational analysis suggests that humans make decisions by strategically allocating their cognitive resources, weighing the computational cost against the potential benefit of a decision. This often results in heuristics or "shortcuts" that might not always yield the perfect solution, but provide a satisfactory outcome with less cognitive effort. The idea is not to always find the best solution, but to find the most efficient one given the cognitive constraints. By adopting this framework, researchers aim to uncover the underlying mechanisms by which humans navigate complex tasks. This framework offers explanations for why certain seemingly suboptimal behaviors might actually be the result of a finely tuned balance between effort and reward, and underscores the adaptability and efficiency of human cognition in a wide range of scenarios.

Probabilistic inference is also one well-known approach to model human problem solving in cognitive science. The application of probabilistic inference methods has also opened avenues to model how humans naturally handle uncertainty, predict future events, and generalize from sparse data [33, 66, 67]. Probabilistic models propose that human cognition functions in a Bayesian manner, where the brain integrates prior knowledge with incoming evidence to update beliefs and make informed decisions. Essentially, these models consider the brain as a probability calculator, continuously updating its beliefs about the world based on the evidence it encounters. When faced with uncertainty,

humans weigh the likelihood of various outcomes based on prior experiences and current observations. This probabilistic approach to cognition can be observed in various cognitive tasks, from simple perceptual judgments to complex problem-solving scenarios. It provides a mathematical framework for understanding how humans predict future events, make sense of ambiguous situations, and even learn from minimal data. Moreover, it helps explain the flexibility and adaptability of human thinking, allowing us to adjust our beliefs and strategies in the face of changing evidence.

2.1.2 Model-based Problem Solving Agents

Like domain-general and domain-specific problem solving models in cognitive research, in the field of AI, a critical distinction also exists between model-free and model-based problem solving agents. Model-free methods, as the name suggests, operate without a concrete model of the environment. These methods learn purely from trial and error, leveraging experiences and interactions to determine the best actions, often without a comprehensive understanding of the environment's dynamics. In contrast, model-based approaches come equipped with a structured model of the environment. Such models act as blueprints, detailing how actions impact the world, which enables more reasoned, systematic exploration and planning.

Recent progress in AI has seen a strong preference for the model-free approach. Well-recognized systems, such as AlphaGo [68], mainly rely on reinforcement learning techniques [69, 70]. However, these methodologies are not without limitations. Notably, there is a clear difference between how model-free systems work and how humans naturally solve problems. Firstly, these AI systems need large amounts of data and a lot of computing power to either come close to or surpass human skills [70]. In addition, the 'black box' aspect of deep learning also raises questions, especially when we think about using these systems in vital areas like healthcare or self-driving cars [25, 71]. Furthermore, these algorithms struggle to deal with novel scenarios [72].

Nowadays, as the AI community leans more towards enhancing human-computer interaction and ensuring system interpretability [73, 74], adopting model-based methods to simulate human behavior might be a step in the right direction [70, 75]. The model-based approach, especially automated planning and search algorithms, stand out due to their

domain-independent nature as human problem solving. To be specific, automated planning uses general search algorithms and possibly domain independent heuristic functions, much like how humans often approach new challenges using broad strategies instead of narrow, specific knowledge. In addition, unlike deep learning systems, which can be complex and need a lot of resources, model-based methods aim for clarity and wider applicability.

2.1.3 Markov Decision Process

Before I move to the agents used to solve the problem, the environment in which agents run should be considered first, especially for the model-based approach.

In the realm of problem-solving tasks, there exist various potential representations, such as Constraint Satisfaction Problems and Finite Automata. However, the Markov Decision Process (MDP) emerges as a central framework within the model-based approach for sequential decision-making problems [76]. The MDP offers a rigorous mathematical structure for scenarios where decision sequences intertwine with both stochastic outcomes and a decision-maker's control. Within this framework, it becomes feasible to identify an optimal policy, which dictates the best action or decision for each state to maximize a cumulative reward over a given timeframe.

While MDPs are a popular and widely used representation, several other formalisms and extensions can be used based on the specific characteristics of the environment and problem [77]:

- **Partially Observable Markov Decision Process (POMDP)** [78]: When the agent cannot fully observe the state of the environment, a POMDP can be used. It incorporates a belief state which represents a probability distribution over possible states the system might be in.
- **Semi-Markov Decision Process (SMDP)** [77]: This is an extension of MDP where the time between transitions is not necessarily one time step, but a random variable with a certain distribution. It is useful when actions might take varying amounts of time to complete.

- **Decentralized Markov Decision Process (Dec-POMDP)** [79]: Used when there are multiple agents making decisions, and the environment is partially observable.
- **Hierarchical Markov Decision Process (HMDP)** [80]: These are used when the decision-making problem can be decomposed hierarchically, with high-level decisions guiding lower-level ones.

For simplicity, my work considers only environments that are both fully observable and deterministic. Within this context, an MDP can be equivalently referred to as a *classical model*. Formally, a classical model is denoted as $\langle S, s_0, S_G, A, a(s), f, c\rangle$, where:

- State Space S is a finite set of discrete states.
- s_0 is the designated initial state, and is a member of S.
- S_G represents the set of goal states, which is a subset of S.
- The Action Space A is a finite set that encompasses all possible actions. The applicability function is defined as $f_a : s \to 2^A$, where 2^A denotes the power set of A, and $a(s) \subseteq A$. This function specifies which actions can be validly applied to a given state s.
- $f : S \times A \to S$ is the state transition function. For a given state s and an action a, it determines the subsequent state s'.
- $c : S \times A \to \mathbb{R}$ defines the cost function. When an agent executes action $a \in A$ in state $s \in S$, it incurs a cost $c(s, a)$.

Here I show how I model the Sokoban task with one box shown in Figure 1.4 using this formulation:

- **State Space (S):**

$$S = \{< box_{loc}, agent_{loc} > \mid box_{loc} \in \text{Grid}, agent_{loc} \in \text{Grid}\}$$

where box_{loc} represents the location of the box on the grid and $agent_{loc}$ represents the player's position. The set Grid contains all possible grid locations (e.g. $loc_1, loc_2, ...$) as propositions.

- **Initial State s_0:**

$$s_0 =< box_{loc}^{initial}, agent_{loc}^{initial} >$$

This state corresponds to the initial configuration of the Sokoban puzzle.

- **Goal Set S_G:**

$$S_G = \{< box_{loc}, agent_{loc} > \mid box_{loc} = target, agent_{loc} \in \text{Grid}\}$$

A state s is a goal state if the box_{loc} is located on the predefined location $target$ in the grid.

- **Action Space (A):**

$$A = \{\text{move\_up}, \text{move\_down}, \cdots \text{push\_up}, \text{push\_down}, \cdots\}$$

.

- **Applicability function (f_a):** A move action is applicable if the player's adjacent cell in the direction of the move is empty (i.e., not occupied by a wall or a box).

A push action is applicable if there is a box in the adjacent cell in the direction of the push, and the cell beyond the box is empty.

- **Transition Function (f):**

$$f : S \times A \to S$$

For instance, if the current state is $s =< box_{loc}, agent_{loc} >$ and the action is move_up, the new state s' would be $< box_{loc}, agent_{loc}^{up} >$ where $agent_{loc}^{up}$ is the location one unit above $agent_{loc}$ on the grid.

- **Cost Function (c):**

$$c : S \times A \to \mathbb{R}$$

$c(s, a) = 1$ for any combination of state s and available action a of that state.

2.1.4 Planning Domain Definition Language (PDDL)

Up to this point, we have presented the model using mathematical formulations. However, when it comes to implementing and running search algorithms, a more descriptive

and structured representation is beneficial. For this purpose, we use the Planning Domain Definition Language (PDDL) [81, 82]. PDDL serves as the standard language for expressing planning problems within the automated planning community. Introduced during the International Planning Competition in the late 1990s, PDDL was developed to provide a common and unambiguous format for specifying planning domains and problems. The language is primarily declarative, focusing on describing states, actions, and their effects. Each problem in PDDL is defined in terms of an initial state, a set of goals, and a domain that lists possible actions along with their preconditions and effects. This structured format facilitates comparisons of different planning algorithms, as they can all operate on the same problem descriptions. Additionally, PDDL has evolved over the years, with extensions supporting temporal planning, preferences, and other advanced features, making it a good tool for a wide range of planning scenarios.

In PDDL formulation, the basic element is the predicate. States in the planning problem are described as a set of predicates, which convey the conditions or properties that are true in that particular state. Actions, on the other hand, are defined in terms of their preconditions and effects. Preconditions specify the necessary predicates that must hold true for the action to be applicable. The effects of an action outline how the set of predicates (i.e., the state) changes upon the execution of that action. Essentially, the effects represent the transition function by indicating which predicates become true and which become false post-action. This framework provides a declarative way to specify the structure of the problem domain, allowing the planner to generate a sequence of actions (a plan) that transitions the system from an initial state to a desired goal state, with the goal itself being defined as another set of predicates that need to be satisfied.

In the Sokoban example, beyond just using predicates to signify the agent's and the box's locations, it's essential to incorporate predicates that capture adjacency between locations. In our grounded version of Sokoban, each grid location is given a distinct identifier—like `loc1`, `loc2`, `loc3`, and so on. The relationship of adjacency between grid locations can be expressed using the predicate `(adjacent ?x ?y)`, where `?x` and `?y` are variables representing location propositions. The possible values for `?x` and `?y` are limited to identifiers within the set of defined locations, ensuring that adjacency is only established between valid neighboring points on the grid. This predicate holds true if `?x` is directly adjacent to `?y`.

In our initial setup, predicates like `(at-agent loc1)` and `(at-box loc2)` can be used to suggest that the agent begins at `loc1` and the box at `loc2`.

When modeling actions, instead of using directions like "up" or "down", actions are defined in a more general manner by enumerating all feasible movements or pushes from one location to another. Furthermore, to accurately model the impact of actions, it would be beneficial to differentiate between movements that do not involve pushing the box and those do push. For instance, an action might be denoted as `(move-push ?a ?b ?c)`, where the agent moves from location `?a` to `?b`, pushes the box from `?b` to `?c`, with the conditions that `?a` is adjacent to `?b`, `?b` is adjacent to `?c`, and all three locations, `?a`, `?b`, and `?c`, lie on a straight line. Preconditions for this action would require the agent to be at `?a`, the box at `?b`, and `?c` to be vacant. Once executed, the agent's new position is `?b`, and the box is shifted to `?c`.

The goal can specify a certain location for the box, such as `(at-box loc3)`. This means that the puzzle is solved once the box reaches `loc3`.

After expressing the Sokoban tasks in this PDDL format, planning algorithms can craft a sequence of moves for the agent to successfully position the box at the target location, while following Sokoban 's rules and constraints. The complete PDDL files for the problem shown in Figure 1.4 (red target) are provided in the Appendix.

2.1.5 Automated Planning

Automated planning represents a core area in artificial intelligence, focusing on the generation of sequences of actions that transition a system from an initial state to a desired goal state. The General Problem Solver (GPS) can be viewed as an early conceptual precursor to automated planning algorithms, laying foundational ideas for computational problem-solving that have since evolved into more sophisticated and diverse planning techniques. As model-based approaches, these algorithms (also known as search algorithms), equipped with techniques to efficiently navigate vast solution spaces, bridge the gap between high-level objectives and low-level actions. A prominent characteristic of modern planning algorithms is their ability to leverage domain-general heuristics, which are often automatically derived from problem representations (e.g. classical model presented in section 2.1.3) using relaxations [20]. Such an approach not only enhances

the planner's efficiency but also holds the potential to inspire novel models of human problem solving, adaptable across a diverse range of challenges without problem-specific strategies.

Classical Planners

Classical planners are designed for deterministic and fully observable tasks. Two classic search algorithms, breadth-first search and depth-first search, constitute the most basic blind classical planners. Blind classical planners have no information about which state is closer to the goal, and hence the order in which they explore the space of solutions is independent of the goal. By contrast, AI researchers have focused on planning as heuristic search. A heuristic planner usually consists of two components, a search algorithm and a heuristic function derived automatically from the symbolic description of the problem [83]. Since the late 90s, heuristic planning has been the dominant approach to solving deterministic tasks [21, 84, 85]. The latest state-of-the-art planners also include pruning techniques such as helpful actions [84] and novelty pruning [86] in order to reduce the size of the search.

Online Planners

When dealing with complex tasks with limited reasoning time, it is sometimes impractical to find a full plan. Online planners are proposed to deal with this scenario. These planners excel in dynamic situations by iteratively determining the most suitable immediate action rather than attempting to devise a complete plan from the initial state. Situated planning falls within this category, emphasizing an agent's adaptive responses to real-time environmental feedback and its own impact within that environment. This approach is especially relevant when agents must not only respond to the environment but also maintain a continuous interaction with it, aligning the planning process with the immediate and evolving context of the task at hand [87]. A prominent approach to online planning is Monte Carlo Tree Search (MCTS), which has achieved striking success at playing Go [60]. These planners have also been explored as tree search based model of human problem solving in cognitive research [59, 61].

Temporal Planners

Temporal planning extends the scope of classical planning by introducing timing considerations for actions [88, 89]. In this paradigm, actions have durations, can be executed concurrently, and are subjected to temporal constraints regarding their start and end times. This allows for richer, parallelized sequences of actions that adhere to complex timing requirements. Challenges in temporal planning include optimizing for various metrics such as makespan (i.e. total amount of time taken to complete a sequence of tasks), resource usage, or energy consumption. Temporal planners and algorithms, like Simple Temporal Networks (STNs) [88], Forward-chaining partial-order planning (POPF) [90], and Temporal Fast Downward [91], have been developed to handle such complexities.

2.1.6 Domains for problem solving

In the context of problem solving, a domain refers to the set of conditions, rules, entities, and actions that define a particular problem space. For instance, in chess, the domain encompasses the board, the specific pieces, their legal moves, and the objective of checkmating the opponent's king. In automated planning and problem-solving, specifying the domain is crucial because it provides the framework and boundaries within which solutions are sought. By cleanly defining a domain, one can isolate the intrinsic challenges and complexities of the problem, allowing both humans and machines to apply general or domain-specific strategies to find solutions. Furthermore, clear domain definitions enable the reusability of planning algorithms across multiple problems that share similar domain characteristics, thereby emphasizing the strength of domain-independent planning approaches. In this section, I provide an overview of some well-known domains in the field, highlighting their advantages and limitations.

Navigation Tasks

Navigation tasks serve as a prototypical example of problem-solving domains [40, 41, 92–95]. In these tasks, an agent is typically placed in an environment and must find the most efficient or safest path to a specified destination. Environments can range from simple 2D grids, where the agent moves between adjacent cells, to more complex 3D

terrains with varying elevations and obstacles. The challenges in navigation tasks arise from factors such as dynamic obstacles, limited visibility, or even adversaries that the agent might encounter. For example, a robot in a warehouse might have to find the quickest route to retrieve an item while avoiding moving equipment and other robots. Or, in a video game setting, a player character might need to navigate a maze while evading enemies.

Navigation tasks come in various forms, each introducing unique complexities and challenges. One classic variant is the Sokoban puzzle we modelled previously, which originated in Japan. In Sokoban, the player must push boxes onto designated target locations within a confined space. The key is that boxes can only be pushed and not pulled, demanding careful planning to ensure that none become trapped against walls or corners. The task combines spatial reasoning with intricate problem-solving, making it a popular benchmark in both AI research and cognitive studies [40, 41, 92].

Another widely studied navigation-related problem is the Travelling Salesman Problem (TSP). Unlike the grid-based environment of Sokoban, TSP involves finding the shortest possible route that visits a set of cities and returns to the origin city. While it sounds straightforward, the TSP is a combinatorial problem that grows exponentially with the number of cities, making it computationally challenging. Its significance stems from its applicability in logistics, transportation, and even DNA sequencing [93–95].

Navigation tasks are valued for their intuitive appeal and their direct relevance to practical situations, notably in fields like autonomous driving and robotic path planning [96]. In addition, the extensive understanding of these tasks provides a solid foundation for new research, allowing for quicker advancements and more refined methodologies. Yet, they come with certain constraints when modelling human behavior on these domains with automated planning algorithms. A primary one is the tendency of individuals to lean heavily on visual indicators, often bypassing the need for forward search or reasoning. Consider a navigation task within a video game where the player must find their way out of a maze. The game is designed with distinct visual markers, such as brightly colored paths or arrows pointing in certain directions. Players may start relying solely on these visual cues to navigate through the maze, choosing paths that appear visually appealing or following the arrows without considering other potential strategies.

Additionally, the cognitive mapping people employ for tasks like Sokoban, especially in grid environments, might diverge from standard representations in automated planning [97]. Instead of merely considering moves to adjacent cells, people might adopt a more hierarchical approach to understand and solve the problem [98]. Consequently, algorithms designed to tackle such tasks might incorporate methods like Hierarchical Task Network (HTN) planning [99].

2.1.6.1 Disc Moving Tasks

Disc-moving tasks, or tasks that involve the strategic movement and placement of discs or blocks, are also fundamental in studying problem-solving behaviors and strategies [100, 101]. Such tasks usually revolve around rearranging items in specific configurations, demanding a blend of spatial reasoning, planning, and sequencing.

The Blocksworld is one of the most iconic domains in the annals of artificial intelligence and cognitive science [34]. In this environment, the player manipulates a set of blocks on a table to achieve a target configuration, guided by specific rules such as only one block can be moved at a time, and a block can only be moved if there's no other block on top of it. The challenge isn't merely to reach the goal configuration but to do so in an optimal or minimal number of moves. The Blocksworld has been used to study various aspects of reasoning, from causal reasoning to the interplay between perception and action in problem solving [34].

Another classic disc moving task is the Tower of Hanoi [29, 30, 32, 33, 51, 102]. In this task, participants are presented with three or more pegs and multiple discs of different sizes. The objective is to transfer all discs from one peg to another, subject to two constraints: only one disc can be moved at a time, and a larger disc cannot be placed on top of a smaller one. The Tower of Hanoi serves as a benchmark for recursive reasoning since the optimal solution often requires breaking the problem down into sub-problems, solving each in turn, and then combining the solutions One noteworthy variant of the Tower of Hanoi is the Tower of London (TOL) task. While both tasks revolve around disc movement, TOL introduces additional complexities with its flexibility in start and goal configurations. Instead of a fixed initial setup, TOL often involves varied and randomized ball placements on pegs. Moreover, the pegs in the TOL have different height

limitations, further complicating the solution strategy. This flexibility not only intensifies the planning required but also makes the Tower of London task a good playground in cognitive research to assess problem solving strategies [24, 33, 35, 103].

Disc moving tasks like the Blocksworld and the Tower of London are valued for the structured challenge they present, reflecting fundamental aspects of human problem-solving [104]. Disc moving tasks present several distinct advantages for problem-solving research. Firstly, they demand a more concentrated form of explicit reasoning than navigation tasks [32, 33]. Adjusting the complexity of these tasks is also straightforward, facilitating varied levels of challenge [32]. Notably, humans' mental models of disc moving tasks often mirror the way the AI community represents the problem [33]. Additionally, just like navigation tasks, both the AI and cognitive science communities have accumulated a lot of knowledge and outcomes in these domains, making future research and applications more streamlined.

2.1.6.2 Other Tasks

Beyond navigation, and disc moving tasks, the realm of problem-solving stretches into various other challenges [105]. Shape manipulation tasks stand as another cornerstone in problem-solving domains. These tasks necessitate the transformation or rearrangement of geometric or abstract patterns to achieve a specific end state or solve a particular puzzle. An example is the Tangram puzzle, where players must arrange a set of shapes to form a specific silhouette or image. These tasks offer a unique vantage point into visual-spatial reasoning and how individuals deal with the constraints of geometry and spatial relationships. Another popular shape manipulation problem is the jigsaw puzzle, where the goal is to interlock varying pieces to construct a complete picture. Such tasks push individuals to match patterns, colors, and shapes while also keeping the broader image in mind. This interplay between local pattern recognition and global planning offers a rich avenue for understanding cognitive strategies and the balance between detail-oriented and big-picture thinking [106]. Shape manipulation tasks underscore the depth of human spatial intelligence. Unlike tasks that are predominantly logic-based, these tasks emphasize intuitive visual processing and the ability to visualize end goals.

Some researchers also use other domains to study problem solving. Logic puzzles, for instance, often presented in the form of riddles or deductive reasoning games, demand

abstract thinking and the ability to consider multiple constraints simultaneously. An example is the Einstein's Riddle, also known as Zebra Puzzle, which is a form of constraint satisfaction problem [107]. Additionally, real-world tasks, like scheduling, resource allocation, or even everyday chores, can be also structured as problem-solving tasks. For instance, the act of organizing a cluttered room can be viewed as a spatial problem-solving task, where items must be efficiently placed while considering the available space and desired organization [108].

These varied tasks provide a snapshot into the diversity of challenges that can be framed within the problem-solving paradigm. They reinforce the idea that problem-solving is a widespread human skill, applicable across a multitude of scenarios, both abstract and concrete.

2.2 Goal Recognition

In the domain of artificial intelligence, goal recognition refers to the process of inferring the goals or intentions of an agent based on its observed actions or behavior. This task is often referred as intention recognition, and is closely related to plan recognition, which involves inferring the agent's plan by observing its behaviors. Although there can be subtle differences between the two, such as the same goal leading to different plans, or the same plan aiming at different goals, these distinctions are minor and not the focus of my thesis. Therefore, in this thesis I will use goal recognition and plan recognition interchangeably [109, 110].

Goal recognition is worth studying in AI field as it contributes significantly to the development of intelligent systems capable of understanding and predicting human actions. Its applications are vast and include fields like video surveillance, video games, assistive care for the elderly, personal assistant agents, and human-robot interaction. Meanwhile, from the perspective of cognitive science, goal recognition is instrumental in understanding the cognitive mechanisms that underpin our ability to infer intentions, an aspect fundamental to social interactions. Moreover, the principles derived from human goal recognition are crucial for developing sophisticated goal recognition algorithms. Studies on human cognition, especially those related to the Theory of Mind, indicate that individuals predict behaviors by integrating observations with their knowledge of someone's

goals and plans. Applying these human-centric principles to AI systems can enhance their efficiency and accuracy in predicting goals. The understanding gleaned from the human cognitive process is also invaluable in overcoming the challenges inherent in creating such systems, such as navigating uncertainties and adapting to ever-changing environments

The approaches to goal recognition vary and are often categorized into logic-based approaches, classical machine learning approaches, deep learning approaches. In the following subsections, I will discuss the mechanisms of human goal recognition and the logic-based goal recognition algorithms that form the cornerstone of this thesis. Before that, I will provide a concise overview of learning-based goal recognition algorithms in the remaining of this section.

Classical machine learning approaches for goal recognition involve training a model on a labeled dataset of actions and goals. The model can then be used to recognize the goal of a new action based on its features [111]. These approaches are based on statistical models that learn from data and can be used to recognize a wide range of goals. They are relatively easy to implement and can handle large datasets of labeled data. However, they may struggle with complex relationships between entities and may not generalize well to new situations. They also require labeled data, which can be time-consuming and expensive to obtain.

Deep learning approaches are based on artificial neural networks that attempt to mimic the structure and function of the human brain and both of them can be applied to goal recognition tasks [112, 113]. In deep learning, artificial neural networks are trained on large datasets of labeled data to recognize the goal of a new action based on its features. These networks can handle raw sensor data and learn complex patterns from it, making them particularly useful for recognizing goals in video or sensor data. For example, a deep learning approach could be used to recognize the goal of a person walking into a room based on the features of their gait and the objects they interact with. This approach is able to adapt to new situations and generalize well to new tasks. However, they are computationally expensive and lack interpretability.

2.2.1 Theory of Mind and Human Goal Recognition

One crucial skill humans have is recognizing others' goals from their actions and the context in which their actions occur. For example, when observing someone reaching for a cup, we can infer that their goal is to drink from it based on our knowledge of the typical sequence of actions involved in drinking from a cup. This skill falls under the *Theory of Mind*, enabling us to appreciate the diversity in others' thoughts and consider the range of mental states – emotions, desires, intentions, beliefs, and knowledge – that influence how others and ourselves behave [114]. This understanding is vital for successful communication, cooperative endeavors, and social engagement. Children start developing Theory of Mind at an early age [5, 115], indicating that very young children may be able to recognize that other people can have beliefs or knowledge distinct from their own. These abilities keep evolving throughout adolescence and well into adulthood. [116].

Within cognitive science, the study of Theory of Mind is encompassed by social cognition research, which explores the mechanisms of how we perceive and interact with the social world. Two primary theories attempt to explain the functioning of Theory of Mind: the theory-theory and the simulation theory [110]. The theory-theory suggests that people use a simplified abstract model of minds that is different from their own decision-making mechanism to make predictions about the behavior of others. In contrast, the simulation theory proposes that humans use their own mental states to simulate the mental states of others. According to this theory, humans use their own experiences and emotions to understand the experiences and emotions of others.

Imagine a child watching their friend reach for an apple. According to the theory-theory, the child applies a simple theory they have about behavior and motivation – in this case, the understanding that people usually take an apple because they are hungry or because they like apples. Thus they infer that their friend is hungry or like apples. This inference is made independently of child's own hunger; it is based on their generalized understanding of behavior, not their immediate physical state. In contrast, according to simulation theory, the child engages in an empathetic process. They imagine themselves in their friend's situation, drawing on personal experiences of hunger or the enjoyment of eating an apple. However, this doesn't require the child to currently feel hungry. Instead, they recall past experiences of hunger to understand their friend's possible intentions.

By simulating, not replicating, these experiences, the child deduces that their friend is likely reaching for the apple due to hunger or the anticipation of enjoyment.

Bayesian Theory of Mind

While Theory of Mind is central to research in social cognition, there have been relatively few efforts to create quantitative models that simulate human Theory of Mind, and even fewer that accurately reflect the mechanism of human goal recognition processes. The most prominent approach among their lines is Bayesian Theory of Mind (BTOM) [9, 117]. BToM is based on the principles of Bayesian inference, which is a statistical method used to update the probability for a hypothesis as more evidence or information becomes available, and it allows for the incorporation of prior knowledge and the handling of uncertainty. BToM has been shown to be effective in a variety of domains, including spatial navigation [98, 117] and social interaction [118].

BToM suggests that people use Bayesian reasoning to infer the intentions of actors. Observing an actor's actions, a person holds prior knowledge about the initial probability of different potential objectives (prior) and considers how likely various intentions are to produce those actions (likelihood). Then prior and likelihood are integrated, leading to an updated set of beliefs (posterior) about the actor's probable objectives.

Applying the BToM framework, let's analyze the example that a child observes their friend reaching for an apple and aims to infer the underlying intention. The prior in this context would be the child's pre-existing beliefs about why someone might reach for an apple (objectives). These beliefs could include eating the apple, wanting to make a pie, etc. The child may assign probabilities to these goals based on their observations and experiences. For instance, if the child has often seen their friend be hungry, the prior probability for hunger would be high. The prior probabilities might be $P(g_1 = "eat") = 0.7$ and $P(g_2 = "makeapie") = 0.3$.

The likelihood involves assessing how probable the observed action (reaching for an apple) is, given different potential goals. For example, the child sees the friend reach for an apple (o) and knows that if the friend want to eat an apple, the probability of them reaching for an apple is high, let's say $P(o|g_1) = 0.9$. However, if they want to bake

a pie, the probability of reaching for an apple is lower, perhaps $P(o|g_2) = 0.5$, because they also need other ingredients.

Using Bayes' theorem, the child updates their beliefs as follows:

$$P(g_1|o) = \frac{P(o|g_1) \cdot P(g_1)}{P(o)} = \frac{0.9 \cdot 0.7}{0.78} = 81\%$$

$$P(g_2|o) = \frac{P(o|g_2) \cdot P(g_2)}{P(o)} = \frac{0.5 \cdot 0.3}{0.78} = 19\%$$

where $P(o)$ is the total probability of reaching for an apple, calculated by summing the likelihood of reaching for an apple under all hypotheses weighted by their prior probabilities:

$$P(o) = P(o|g_1) \cdot P(g_1) + P(o|g_2) \cdot P(g_2) = 0.9 \cdot 0.7 + 0.5 \cdot 0.3 = 0.78$$

Therefore, the child will favor the hypothesis with the higher posterior probability. In this case, it points to the likelihood that the friend is reaching for the apple due to a desire to eat the apple.

2.2.2 Logic-based Goal Recognition Algorithms

Having explored the Bayesian Theory of Mind, which offers a probabilistic view on understanding intentions, we now turn our attention to the field of AI and its logic-based goal recognition algorithms. These algorithms provide a different approach, employing formal logic to deduce goals from observed actions and contrasting with the Bayesian model's statistical inferences. These algorithms use logical frameworks to establish diverse relationships among entities, like preconditions, mutual exclusions, or decompositions. Additionally, these logical relationships permit the systems to produce new potential plans on the fly, which are then contrasted with real observations.

Two commonly used logic-based approaches are plan library-based algorithms and domain-theory based algorithms [110]. Algorithms based on plan libraries, also known as plan recognition as parsing, present plans as a hierarchy of simpler actions. The main task

becomes aligning the observed actions with these structured plans. Hierarchical task networks (HTN) and grammars are typical methods for representing knowledge in plan libraries [119]. HTN outlines tasks using a set of subtasks and their constraints, either separately or in relation to each other. Meanwhile, grammars describe the structure of plans through a set of production rules. These algorithms are useful in domains where the set of possible plans is known in advance, such as in video game AI and robotics.

On the other hand, algorithms based on domain theory, often referred to as plan recognition as planning (PRP), use standard planning algorithms to create potential plans for the observed agent [37, 39, 120] . These planning algorithms typically depend on planning languages like STRIPS or PDDL (refer to section 2.1.4), enabling them to outline the state of the environment and the impacts of applicable actions. They also use this information to formulate potential plans that could achieve specified goals. The plan recognition system then assigns weights to these candidate plans based on gathered observations, and the most likely plan or goal is chosen based on these weights.

In the framework of PRP, a goal recognition problem is defined as follows:

Definition 2.1. A planning domain $D = \langle S, s_0, A, f, c \rangle$ consists of a finite set of discrete states S, an initial state $s_0 \in S$, a finite set of actions A, a state transition function $f : S \times A \to S$ that maps a state-action pair (s, a) into another state s' and a cost function $c : S \times A \to \mathbb{R}$ which specifies the cost $c(s, a)$ incurred when applying action $a \in A$ on state $s \in S$.

Definition 2.2. A goal recognition problem $G = \langle D, O, S_G, prior \rangle$ is defined within a planning domain $D = \langle S, s_0, A, f, c \rangle$, where:

- $O = \{o_1, o_2, ..., o_n\}$ is a sequence of observations.
- $S_G \subseteq S$ is a set of possible goal states.

The goal recognition task is to infer the most likely goal state $g \in S_G$ that the agent intends to reach, given the sequence of observations O and the planning domain D.

For the PDDL files that define the corresponding goal recognition problem shown in Figure 1.4, please see Appendix.

Interestingly, given the formulation, PRP also adapt the bayesian framework to solve the goal recognition problem like BToM. It transforming the inference problem to the problem that find the goal with highest posterior:

$$P(g_i|O) = \frac{P(O|g_i) \cdot P(g_i)}{P(O)}$$

where:

- $P(g_i|O)$ is the posterior probability of goal g_i given the observations O.
- $P(O|g_i)$ is the likelihood of observing O if g_i were the true goal.
- $P(g_i)$ is the prior probability of g_i, representing our initial belief about the likelihood of g_i.
- $P(O)$ is the probability of the observations under all possible goals, computed as:

$$P(O) = \sum_{j=1}^{m} P(O|g_j) \cdot P(g_j)$$

The goal with the highest posterior probability $P(g_i|O)$ is considered the most likely goal the actor is trying to achieve.

In PRP, most researchers simply assume the uniform prior [37, 38, 120]. The main contribution of PRP in contrast to BToM is that PRP provides a logic-based method (i.e. planning algorithms) to generate and evaluate possible plans an observed agent might be executing for likelihood estimation. Bayesian reasoning can then be applied to weigh these candidate plans against observed behaviors, updating the probabilities of each goal.

In Chapters 4 and 5, I extend the Bayesian Theory of Mind (BToM) and Plan Recognition as Planning (PRP) methodologies to investigate how factors other than observed actions affect human goal recognition. I will demonstrate the adaptability of the Bayesian framework across human and AI contexts and highlight how additional information can be used to enhance the performance of goal recognition algorithms.

2.3 Summary

As we conclude this chapter, I have shown that AI and cognitive science share common themes in their literature. This parallel is not merely coincidental but rather indictive of the complementary nature of the two disciplines. With the rise of human-robot interaction as a prominent field of study, the need to integrate these disciplines becomes increasingly crucial and presents a promising frontier for research and application. By integrating the computational power and algorithmic precision of AI with the deep, experiential insights of human cognition from cognitive science, it might lead to advancements for both disciplines.

Within the specific contexts of planning and goal recognition, the interdisciplinary union offers a robust foundation for developing systems in human-agent interaction scenario. Such systems can be used to not only mimic human-like problem-solving and goal recognition capabilities, but also understand the underlying cognitive processes, which equips them to interact with humans more effectively in real world applications. In the subsequent chapters, I will present three distinct projects that contribute to this endeavor, encompassing computational models for problem solving and goal recognition. These models incorporate and synthesize insights from both disciplines to offer a more comprehensive understanding of these cognitive processes.

Chapter 3

Automated Planning Algorithms and Human Performance[1]

3.1 Introduction

In the warehouse setting, human workers need to coordinate a series of actions, such as deciding which item to pick next and determining the best path to reach the item, in order to prepare a package for shipment. Modeling human behavior in this context is essential for agents to comprehend and even predict human actions through simulation. Cognitive scientists often refer to these types of activities as problem-solving tasks, while AI researchers commonly classify them as planning tasks. Problem solving or planning is a hallmark of intelligent behavior, and has been extensively studied by both AI researchers and cognitive scientists since the development of the Logic Theorist in 1956, a theorem prover sometimes described as the first AI program [50, 121]. In subsequent decades, psychologists have studied human performance on a wide range of problem solving tasks, including water jug problems [122] and the tower of Hanoi [30].

Planned behavior can be distinguished from reflex behavior, similar to the distinctions regarding goal-directed versus habitual behavior, model-based versus model-free decision making, and type II versus type I reasoning [15]. Reflex-based approaches do not consider the outcome of each action or evaluate the utility of these outcomes, which

[1]This chapter is adapted from the published article: "Comparing AI Planning Algorithms with Humans on the Tower of London Task." Proceedings of the Annual Meeting of the Cognitive Science Society. Vol. 45. No. 45. 2023.

limits their ability to perform well in dynamic situations. Both approaches are used by humans and animals, and in order to study human planning behavior, it is essential to choose tasks that cannot be solved through reflexive behavior, forcing participants to rely on planned behavior to solve the problem.

In this study, I build on an approach to planning that was initially developed by researchers including Newell et al. [104]. I have selected a simple task that is suitable for laboratory study and is unlikely to be solved through reflexive behavior [15]. For us the task is the Tower of London (TOL) problem, a variant of the well-known Tower of Hanoi problem. My goal is to identify a planning algorithm that matches human performance on the TOL task, and towards that end I evaluate a set of planning algorithms including several inspired by state-of-the art approaches in AI. My approach therefore falls squarely in the tradition established by researchers like Newell and Simon who used computational models such as the General Problem Solver (GPS) to account for human performance on tasks like the Tower of Hanoi [30, 104].

The Newell-Simon approach to problem solving arguably reached its pinnacle in the 1970s, and has been pursued less actively from the mid 1990s onwards [123]. There are at least two reasons, however, why this approach may be worth revisiting. First, AI researchers have developed new approaches to planning that may help to capture aspects of human problem solving. For example, from the mid 1990s modern planning algorithms have relied on domain-general heuristics that can be derived automatically from a problem representation via *relaxations* [20]. This approach to deriving heuristics could potentially lead to new models of human problem solving that can be applied to broad families of problems without requiring problem-specific strategies.

Second, psychologists have continued to construct new computational models to account for several aspects of human decision making [58, 124]. A key issue explored in recent computational modeling work is the tradeoff between time cost and decision quality. Models exploring this idea build on the idea of bounded rationality [125] and the framework of rational analysis [126]. An agent that makes optimal use of bounded cognitive resources must decide when to stop the search process and act, and recent work on metareasoning has explored this *stopping problem* [127, 128]. Solway and Botvinick [58] use an evidence accumulation mechanism to model performance in a two-step decision

problem, but applying a similar approach to more complex sequential decision making problems (e.g. TOL) is a challenge that has not yet been addressed.

The next section provides a brief overview of previous computational research on problem-solving, laying the foundation for the work discussed in this chapter. For more in-depth coverage of relevant studies, please refer to Chapter 2. I then describe the Tower of London task and the behavioral experiment. The following sections introduce the specific planners that I evaluate and discuss the extent to which they account for the behavioral data. As a preview of my results, I found that people tend to use different strategies under different conditions and that the adaptive lookahead planner provides the best overall account of human performance.

3.2 Models of Human Problem Solving

Perhaps the most influential cognitive model of problem solving is the General Problem Solver [104] and this model can be regarded as a variant of breadth first search. Subsequent work in this tradition used production systems such as ACT-R [129], 4CAPS [24] and SOAR [130] to develop models of problem solving on tasks including the Tower of Hanoi [29] and the Tower of London [24].

In recent years researchers have departed from the earlier emphasis on production systems by considering a range of alternative approaches. Kuperwajs et al. [59] used a tree search model with a domain-specific heuristic to predict human performance on a two-player game. Working within the framework of bounded rationality, Callaway et al. [63] derived a meta-level Markov decision process model to simulate human behavior on a navigation task known as Mouselab. Donnarumma et al. [33] developed an approach that combines probabilistic inference with subgoaling to account for human performance on the Tower of Hanoi task.

Across the recent literature there is evidence that the extent to which people look ahead while planning varies across individuals and across tasks [64, 131]. Meder et al. [132] found that an approach that looks ahead only one step provided the best account of human performance in the 20-questions game, while Krusche et al. [61] found that people have a planning horizon of at least 3 steps in the farming game that they considered.

Several studies demonstrate that time pressure can lead to a shallower search tree [133, 134].

Most recent studies use non-deterministic or partially observable environments so that humans cannot easily derive optimal solutions [61, 131], and there has been relatively little work on fully observable deterministic environments (e.g. TOL) in recent years. My work, however, belongs to the Newell and Simon tradition that explores what can be learned from human performance on deterministic, fully observable-tasks. A small amount of work in cognitive science explored how to use computational model to reproduce human behavior in the TOL task[24, 33], but none of them considered response time (RT) as far as I know.

3.3 Tower of London Task

Previous work on the TOL has focused on identifying structural parameters that appear to influence the difficulty of a problem instance[36, 135, 136]. Berg et al. [36] carried out an experiment in which participants solved a set of TOL problems with optimal solutions of length between 4 and 7, and used their data to evaluate how 5 structural parameters relate to measures of human performance.

Inspired by their work, our experiment consider two different conditions (described later) and use two most influential factors on initial planning time (i.e. optimal cost and start hierarchy [2]) as baselines in planning time prediction.

A small amount of work has attempted to model the actions people choose when solving TOL problems [24, 33], but to my knowledge no previous work on the TOL task attempts to model both action selection and planning time as I do here.

3.4 AI Planning and Planners

Planning is the model-based approach to reasoning about the action(s) needed to achieve a goal given an initial scenario. In contrast to approaches based on constraint programming [63, 64, 131] that do not naturally capture human problem solving mechanisms,

[2] *Start hierarchy* is based on the initial configuration which is either unambiguous, partially ambiguous, or completely ambiguous (three balls on different pegs), see figure 1.3(c)

AI planning algorithms directly simulate the forward reasoning process, echoing the methodology used by the General Problem Solver. For detailed description of how to apply planning algorithm to problem solving tasks, please refer to Chapter 2.1.

In order to apply AI planners to the TOL problem, I translate the task into the propositional subset of the Planning Domain Definition Language (PDDL), which is a standard language for modelling planning problems that extends the expressivity of the well known STRIPS language [82]. To encode the height constraints in the task, I simply enumerate all possible ball locations. In my setting, since there are just three pegs with heights of 1, 2, and 3 respectively, I have 6 different locations in total. In each state, there is a fluent (proposition) for each ball recording its current location. In addition, I also mark whether each ball is free to move and whether each location is available. For example, in the start state of Figure 1a, the red ball is in LOC3-3 (the third position on peg 3). There is no other ball on the red ball, so it is free to move to other locations. LOC1-1 is available, so I can execute the action that moves the red ball from LOC3-3 to LOC1-1 and the successor state is the middle state in Figure 1.3c.[3].

All of the AI planners evaluated here use the representation just described, but there is another way to model the problem within the PDDL framework. Namely, we can decompose each move action into two steps: first pick up a ball from one peg and then put it down on a peg. The major advantage of this approach is that it allows a player to pick up a ball then return it to the same peg, which occurs occasionally in our behavioral data. I evaluated the planners on both representations, and the differences are relatively subtle. Since most previous research on the TOL treats each move as a single action, I adopt the same approach for consistency.

My model evaluation aimed to consider a set of planning algorithms (i.e. planners) that is broadly representative of prior work on planning in the fields of AI and psychology. The following sections describe the 6 different planners that I settled on.

[3]The PDDL representation of the start state of Figure 1a is {(in RED LOC3-3), (in ORANGE LOC3-2), (in BLUE LOC3-1), (free LOC1-1), (free LOC2-1), (free LOC2-2), (clear RED)}

3.4.1 Cognitive Architecture

4CAPS

The 4CAPS (Cortical Capacity-Constrained Concurrent Activation-based Production System) cognitive architecture integrates both symbolic and connectionist models, while accounting for human cognitive constraints. This architecture proposes that functions are distributed and dynamically balanced across independent processors, which are designed to mimic different brain regions. The component centers correspond to particular brain regions that activate for a particular task. Each center is implemented as a production system like other cognitive architectures, but potentially with different computing approach (e.g. propositional, geometric etc). Communication between the centers is achieved through memory. This involves two types of memory: declarative memory, which stores knowledge (akin to predicates in automated planning), and procedural memory, which specifies the effects and conditions of actions, as well as the action selection criteria (similar to actions and search algorithms in automated planning). The amount of processing activity in each center reflects the activity in the corresponding brain area.

I chose 4CAPS to represent the broader family of cognitive architectures because an existing 4CAPS model of the TOL task is publicly available, and has previously been used to account for both behavioral and brain imaging data [24, 102]. This model includes some productions that are specific to the TOL task, and therefore does not qualify as a fully general model of problem solving.

3.4.2 Classical Planners

Classical planners search until a complete path to the goal has been found. I considered three such planners: Breadth First Search (BrFS), A* and Greedy Best First Search (GBFS).

BrFS

The three-peg TOL problem is sufficiently small that Breadth First Search (BrFS) is a viable algorithm. BrFS first tries all possible actions from the start state, and adds all

states reached in this way to a queue. It then repeatedly takes a state from the front of the queue, tries all actions from that state, and adds all resulting states to the end of the queue, effectively always expanding the state closest to the initial state that has not been expanded yet. Proceeding in this way guarantees that BrFS will find an optimal solution, but the algorithm is blind because it does not consider the goal when choosing the state to expand next.

A*

The A* search algorithm [137] is commonly used as a baseline heuristic search planner in AI planning research. A heuristic is a function that takes a state as input and returns an estimate of the distance between the state and the goal. A heuristic-based algorithm can therefore potentially capture the idea that people are most likely to focus on intermediate states that promise to bring them closer to their ultimate goal. If equipped with an admissible heuristic, then A* is guaranteed to find an optimal solution.[4] When choosing which state to expand next, A* picks the state that minimizes the cost to reach that state plus the heuristic estimate of the distance to the goal. Here I use the *goal-counting* heuristic, a domain-independent heuristic that can be automatically derived from the PDDL description of the problem, which evaluates a state based on how many goals are yet to be achieved (in our case, how many balls are not yet in their final positions).[5] This heuristic is equivalent to the "perceptual distance" heuristic in the psychological literature [33], and has been explored by researchers including Simon [138].

GBFS

The heuristic search algorithm used in most state-of-the-art satisficing planners is greedy best-first search (GBFS) [139]. In contrast to A*, GBFS expands states using only the heuristic function, and chooses the state that lies closest to the goal according to this function. GBFS is not guaranteed to find an optimal solution and hence produces satisficing planners that trade off solution quality and solution speed. When combined

[4] A heuristic function is *admissible* if it never overestimates the real distance between a state and the goal state.

[5] The goal-counting heuristic is admissible in the TOL task. The A* planner in this paper is therefore guaranteed to find an optimal solution.

with the goal-counting heuristic, GBFS yields a search strategy that captures some of the core ideas of means-ends analysis [104].

3.4.3 Online Planners

Online planners are able to choose an action before a complete path has been found, and have been previously explored as models of human problem solving [59, 61]. One prominent approach is Monte-Carlo Tree Search, but I did not consider this approach because it is best-suited for stochastic environments and the TOL is a deterministic task. Instead, I evaluate two lookahead planners that both rely on the goal-counting heuristic.

Lookahead

The basic lookahead planners I consider have a fixed horizon that was set to values from 1 to 7 (maximum solution length). The planner evaluates the value of a state recursively using the minimal state value of its successors, and the state values of all leaf nodes are based on the heuristic function (goal-counting in this work). After computing these state values, the planner chooses the path with minimal estimated cost. If multiple paths have the same minimal value, the planner randomly chooses one of these paths.

Adaptive Lookahead (A-LH)

Although many online planners (e.g. Monte-Carlo Tree search) use a fixed planning horizon or a pre-defined timing budget, a small amount of work in AI has explored methods for optimizing lookahead depth [140]. For example, Kryven et al. [131] develop a model with an adaptive planning horizon for a task that involves navigating through a maze.

Here I propose and evaluate an adaptive lookahead planner (see Algorithm 1) that draws on prior work on evidence integration and human meta-reasoning [58, 127]. To achieve a balance between exploration and exploitation, this planner uses the upper confidence bound (UCB) algorithm as an action selection strategy [141], and keeps searching (evidence integration) until enough nodes have been expanded to suggest that

Algorithm 1 Adaptive Lookahead 1

Parameters: Decision Threshold θ, Exploration constant C
Input: Search space P, Search goal g, Current state s_0
Output: Action selected a, Number of expanded nodes n (used as a proxy for the planning time for this step)

1: Let $tree = Tree(s_0)$, $n = 0$ {Construct a tree rooted on state s_0}
2: Let $v' = -\infty, v'' = -\infty$ {v' and v'' denote the best child node value and second best child node value of the root node respectively}
3:
4: **while** $|v' - v''| \leq \theta$ **do**
5: $node \leftarrow tree.root$ {Use UCB policy to select node for expansion from root node}
6: **while** $node$ is expanded **do**
7: $node = UCBSelect(node, C)$
8: **end while**
9:
10: **if** $node.state$ is g **then**
11: **return** $a \leftarrow softmaxSelect(tree.root), n$
12: **end if**
13:
14: $n \leftarrow n + 1$
15: **for** $succ$ **in** $P.successors(node.state)$ **do**
16: $node.children.add(Node(succ, gc(succ)))$ {Initialize the new generated node using goal counting heuristic}
17: $tree.update(gc(succ))$ {Backpropogate the new evidence so values of all ancestor nodes are updated by selecting the best child node}
18: **end for**
19:
20: $v', v'' \leftarrow tree.getBest()$
21: **end while**
22:
23: **return** $a \leftarrow softmaxSelect(tree.root), n$

the difference in value between the best action and the second best action exceeds some decision threshold. My implementation in this work sets the threshold θ to 1 because the goal-counting heuristic is integer-valued. The exploration constant in UCB algorithm is set to 1.

Given a problem with a goal hypothesis g and initial state s_0, we carry out the procedure laid out in Algorithm 1 and described below until the goal state is achieved or the predefined threshold is exceeded. The input includes tree T containing only the current state. the algorithm traverses the tree using the UCB policy [141] (lines 6-8): for each node choose the action a that maximizes $-c_a + \sqrt{\frac{\log n}{n_a}}$ recursively, where c_a is the current estimation of expected cost-to-go if action a is chosen, n_a is the number of times action a was chosen at the node and n is the number of times the node has been visited.

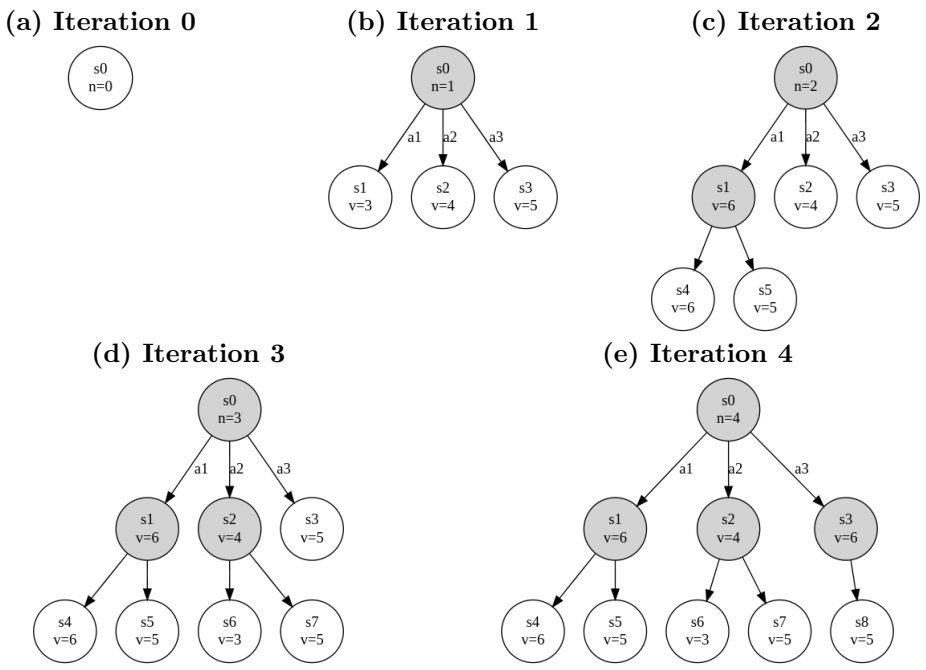


FIGURE 3.1: The Adaptive Lookahead planner run on a goal-directed example. The dark circles represent expanded nodes and the light circles represent nodes generated but not expanded yet. The value v of the node is initialized as some heuristic function (i.e. estimated cost-to-go) of the corresponding state and then updated to reflect the best (i.e. smallest) child node value plus action cost 1.

When the search process encounters an unexpanded node, it checks whether this node represents the goal state g. If it does, the search halts, and the action is selected based on a softmax probability distribution (lines 10-12). If the node is not the goal, it is expanded, generating all potential successor states while excluding any previously visited states to prevent duplication. Each new successor state is assigned an initial value reflecting the anticipated remaining cost (lines 14-18). This value is then propagated backward to the root node (line 20) to inform subsequent decisions.

Once the adaptive lookahead planner reaches the decision threshold, it employs the softmax function to calculate the probabilities for each action choice (line 23), guiding the agent's next move.

We use a goal-directed example to show how the Adaptive Lookahead planner generates human-like planning times and actions using a value-based example. Here we simply assume the cost of action is 1 and the value denotes the estimated cost-to-go. In this configuration the best successor state would be the minimal value.

In the example shown in Figure 3.1, I set the decision threshold at 1 and initiate the process from the current state, denoted as s_0 (Figure 3.1a). In the first iteration, we expand

the start node by generating all applicable actions: a_1, a_2, a_3. The initial heuristic values for these actions are 3, 4, and 5, respectively (see Figure 3.1b). The difference between the best child node, s_1, and the second-best child node, s_2, is 1, exactly matching the decision threshold, so we continue. In the second iteration, we select to expand s_1 and generate two successor states, s_4 and s_5. The value of s_1 is updated to reflect the best child node plus action cost, resulting in $min(6, 5) + 1 = 6$ (refer to Figure 3.1c). Now, the difference between the top two child nodes is still 1, prompting us to continue. In the subsequent iteration (Figure 3.1d), we choose to expand s_2 and generate two successor states, s_6 and s_7. The value of s_2 remains at 4, calculated as $min(3, 5) + 1 = 4$. As the difference is still 1, we must continue. By the fourth iteration (Figure 3.1e), we expand s_3 and introduce a new node, s_8, with a generated value of 5. Consequently, the value of s_3 is updated to 6. At this point, the difference between the top two child nodes becomes $6 - 4 = 2$, exceeding the decision threshold, allowing us to terminate the process. The planning time required is 4. If the decision threshold is larger than 2, then the search needs to continue until the difference between the two best successors is larger than 2.

3.4.4 Implementation

All classical planners, as well as the heuristics were implemented using the LAPKT framework [142]. BrFS, and the online planners were implemented in Python. For 4CAPS, I used v1.2 of the TOL model.

3.5 Behavioral Experiment

To allow us to compare the planners just described, I ran a behavioral experiment to collect fine-grained behavioral data (including response times) as participants solve instances of the TOL. Berg et al. [36] previously ran a comprehensive experiment on the TOL, but their data are not publicly available. I therefore ran my own experiment using the same problem instances that they considered.

My experiment included two between-participant conditions: a *full* condition and a *no-constraint* condition. In the full condition participants were asked to form a full plan to the target configuration before making their first move, and given feedback after each instance indicating whether they had found an optimal solution. In the

no-constraint condition participants were simply asked to solve the task without any further instruction. The *full* condition matches the procedure used by Berg et al. [36], and explicitly instructs participants to act as a classical offline planner. In the absence of this instruction, I anticipated that participants would behave more like an online planner.

Overall, we want to

1. Evaluate a set of planners and ask which provides the best account of human initial planning time and action selection.
2. Test whether these two measurements (and best planner) will be different when I explicitly ask participants to form a complete plan.

I pre-registered the behavioral experiment on AsPredicted (see `https://aspredicted.org/STK_41D`). The experiment was programmed in javascript using the jspsych toolbox [143].

Instances. Following Berg et al. [36], I considered all 117 problem instances with optimal solutions between 4 and 7 in length. For each instance, I generate a corresponding PDDL file automatically using the Python package Tarski [144].

Participants. 239 participants from standard sample [6] completed the experiment on Prolific. Participants were randomly assigned to one of the two conditions, and completed 39 TOL instances randomly picked from 117 instances. My final data set included 130 participants in the full condition and 109 in the no-constraint condition.

Outliers. Observations with abnormal response times were excluded according to a preregistered criterion. For each instance, responses more than 3 standard deviations away from the mean initial planning time for that instance were considered abnormal. As a result, 239 out of 9321 (2.5%) responses are classified as outliers and excluded from my analysis.

[6] Standard sample refers to a sample that matches the demographic distribution of the general population of all Prolific users with English reading proficiency.

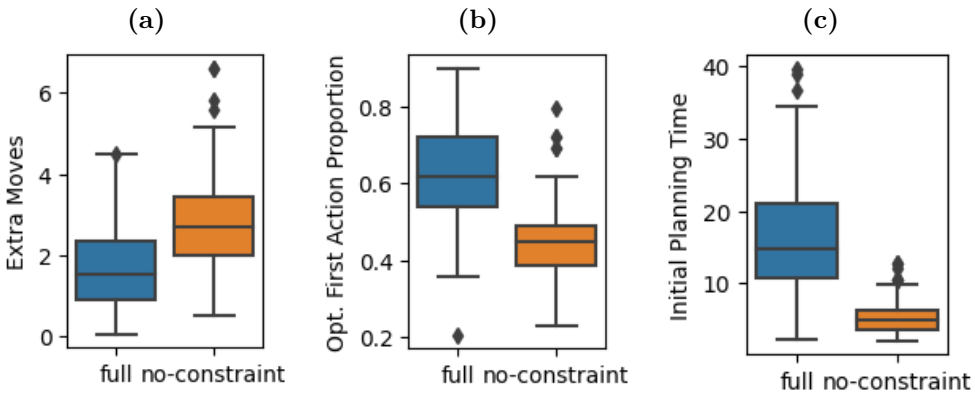


FIGURE 3.2: Comparison between the full and no-constraint conditions at participant level. (a) Extra moves (b) Optimal first action proportions (c) Initial planning times. Each data point shows the average performance across trials for each participant.

3.6 Results

I consider two behavioral measures: the initial action selected for an instance and the initial planning time, or the time taken to select the initial action. Focusing on the first action only simplifies my analyses and facilitates comparisons across a relatively large set of planners.

3.6.1 Human Performance in Two Conditions

I first compare human performance across the two conditions (full vs no-constraint) as shown in Figure 3.2. I focus on three performance measures. *Extra moves* (Figure 3.2a) is defined as the difference between the length of the plan provided by a participant and the length of the optimal plan. I also computed the proportion of participants who select an optimal first move (Figure 3.2b), and considered the time required to select this move (Figure 3.2c).

Figure 3.2 shows that participants in the full condition tend to generate plans that are 1.16 steps shorter than plans in the no-constraint condition, and that the first move in the full condition is more likely to be optimal (61% vs 44%). On average, however, participants in the full condition take an extra 11.23 seconds to produce this first move. Student's t-tests suggest that all three differences are statistically significant: extra moves ($t(238) = -8.12, p < 0.0001$), optimal first action proportion ($t(238) = 10.85, p < 0.0001$) and initial planning time ($t(238) = 15.07, p < 0.0001$).

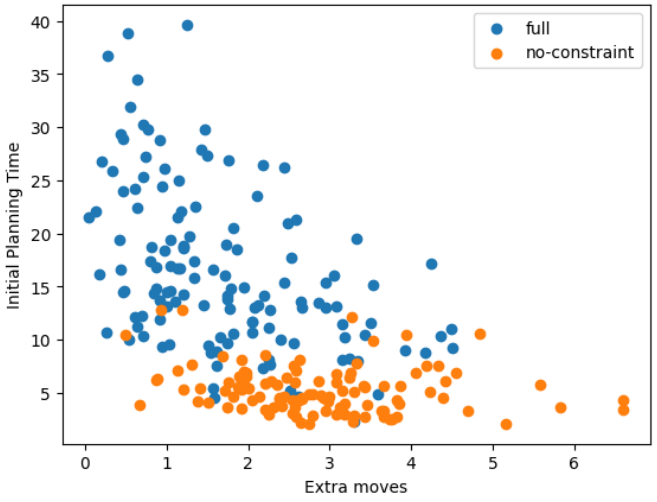


FIGURE 3.3: Extra moves vs initial planning times at participant level

Each data point in Figure 3.2 shows a participant rather than an instance, but an analysis at the level of problem instances produced converging results. For a given instance, plans generated in the full condition tend to have fewer steps ($t(116) = -7.99, p < 0.0001$), are more likely to include an optimal first move ($t(116) = 12.51, p < 0.0001$), and have a longer planning time for the the first move ($t(116) = 20.29, p < 0.0001$).

I then explored the relationship between extra moves and initial planning time at the participant level. I found participants in full condition with longer initial planning time generally have better solution quality ($r(129) = -0.47, p < 0.0001$) but I did not find the same pattern for no-constraint condition ($r(108) = -0.21, p = 0.03$).

As shown in Figure 3.3, there is no correlation between initial planning time and extra moves in the no-constraint condition ($r(108) = -0.01, p = 0.94$) while in the full condition they have a positive correlation ($r(129) = 0.49, p < 0.001$), indicating that in the no-constraint condition, the initial planning time does not appear to significantly influence the overall solution quality, whereas it does in the full condition.

All of these results suggest that my condition manipulation had the expected effect, and that participants rely on different problem-solving strategies across the two conditions. I can now ask which planners provide the best account of responses in the two conditions.

3.6.2 Predicting Action Selection

I first evaluate the extent to which the models can accurately predict the first action selected by participants. For each instance, I use the behavioral data to estimate a distribution over initial actions chosen for that instance. I compare these distributions with distributions derived from the models using *cross-entropy*, which is commonly used as a measure of how well the model can approximate human responses. Since most of the models are non-stochastic and assign a probability of 1 to one action and 0 to all others, inspired by Jarušek and Pelánek [42], I incorporated a noise parameter set to 0.05, distributed uniformly across all applicable actions, to mitigate the issue of zero probabilities. This is to say, to maintain a valid probability distribution, I renormalized the sum of the action probabilities to 0.95, with the remaining 0.05 allocated to a random action selection policy.

The results are summarized in Figure 3.4. Across both conditions, the online planners outperform the classical planners, and A-LH achieves the best overall performance (smallest cross-entropy). The paired t-tests showed that A-LH had a significant advantage over the second best planners in both conditions ($t(116) = -5.04, p < 0.0001$ for the full condition with LH4 and $t(116) = -4.98, p < 0.0001$ for the no-constraint condition with LH3). Although the poor performance of classical planners was anticipated in the no-constraint condition, the fact that they performed worse than the random baseline, despite participants being instructed to behave like classical planners in the full condition, is noteworthy. This finding indicates that classical planners may have limited psychological validity even under conditions that are most favorable to them. Nevertheless, the observation that LH4 is the second best planner in the full condition and LH3 is the second best planner in the no-constraint condition suggests that individuals might engage in deeper thinking in the full condition.

3.6.3 Predicting Initial Planning Time

We now turn to initial planning times, and use Linear Mixed Effects Models to evaluate our family of planners. I consider the following models:

- Base model M0: $IPT = 1 + (1|instance) + (1|participant)$, fixed intercept plus random intercepts for participants and instances.

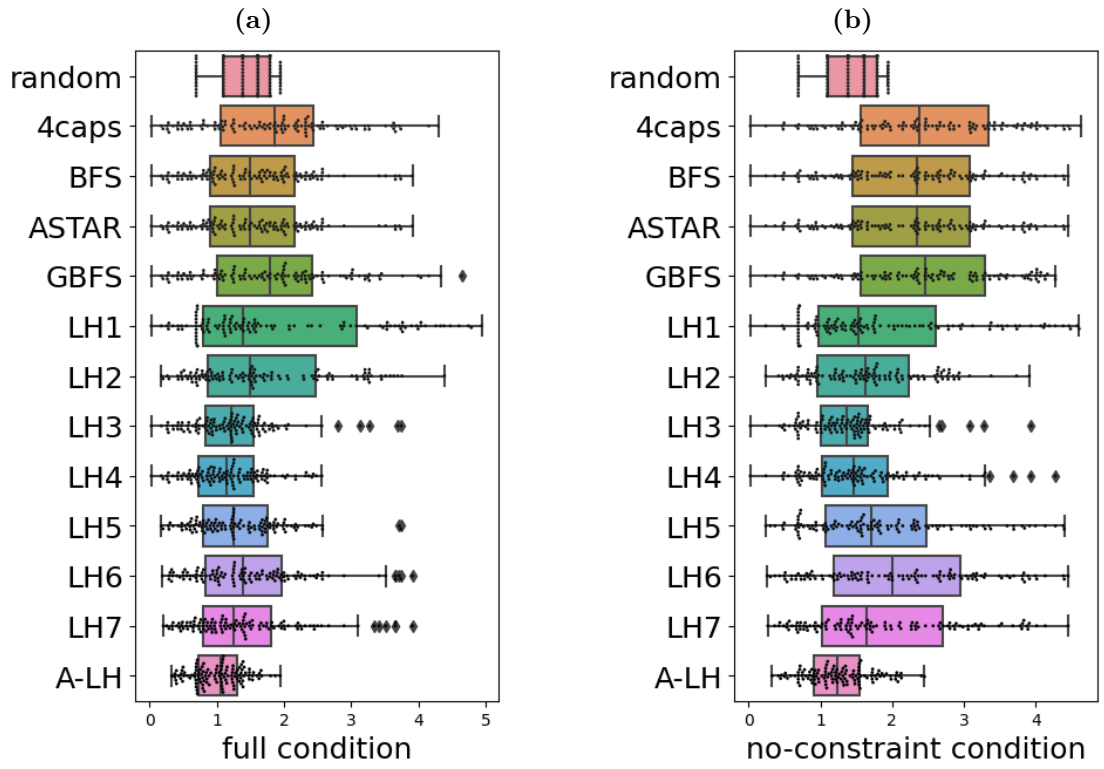


FIGURE 3.4: Evaluation of planner predictions about initial action selection. (a) Cross-entropy of human distribution with respect to model distribution for the full condition. (b) Cross-entropy for the no-constraint condition. Each data point shows the cross-entropy for one instance, and smaller values of cross-entropy indicate better fits.

- Order model M1: $IPT = 1 + order + (1|instance) + (1|participant)$, fixed intercept and order effect plus random intercepts for participants and instances.
- Condition model M2: $IPT = 1 + condition + (1|instance) + (1|participant)$, fixed intercept and condition effect plus random intercepts for participants and instances.
- Full model M3: $IPT = 1 + order + condition + (1|instance) + (1|participant)$, fixed intercept, condition effect and order effect plus random intercepts for participants and instances.

The models take initial planning time (IPT, measured in milliseconds) as the dependent variable, and include fixed effects for condition (full or no-constraint) and order (an integer from 1 to 39 that indicates the order in which a participant encountered a given instance). The models also include random effects for instance and participant that assume normally distributed variability for both factors, which are denoted as 1|instance and 1|participant respectively. I obtained similar results regardless of whether

instance is treated as a fixed or a random effect. We use Bayesian Information Criterion (BIC) to quantitatively compare the models. BIC is a statistical measure used for model selection that balances model fit and complexity by penalizing models with more parameters. It is commonly employed in situations where multiple competing models are under consideration, helping to identify the most suitable model favoring simpler models with comparable explanatory power.

As expected, the full model performed better than the three simpler alternatives that omit either or both of the fixed effects. The BIC value was smaller for the full model than for the three alternatives by a factor of at least 81.

For the full model, the estimate for *condition* is 11192.72 (95%CI [9734.34, 12651.33]), which suggests that responses were around 11 seconds slower in the full condition compared to the no-constraint condition. The estimate for *order* was -102 (95%CI [-123.36, -81.19]), suggesting that participants became around 0.1 second faster with each additional instance that they solved. This order effect is consistent with the work of Berg et al. [36], who report that solution times decrease with experience.

For each planner, I then asked whether the full model could be improved by replacing the random effect of instance with a fixed effect for planner response time, which is operationalized as the number of states expanded by a planner. For example, if the adaptive lookahead model predicted human planning times perfectly, then including response times for this model as a predictor in the full model should allow the resulting regression model to perfectly account for the human data (see Equation 3.1). BIC values for each of these regression models are shown in Table 3.1. Among the fixed lookahead models, LH4 and LH6 achieved the best performance in the no-constraint and full conditions respectively.

$$\text{IPT} \sim 1 + \text{condition} + \text{order} + \text{model prediction} + (1|\text{participant}) \tag{3.1}$$

Table 3.1 also includes baselines that result from replacing the random effect in Equation 3.1 with fixed effects for optimal cost (OC, or the length of the shortest solution) and start hierarchy (SH, see Figure 1.3c). I consider both optimal cost and start hierarchy because these structural parameters predicted human performance best among the full set considered by Berg et al. [36].

TABLE 3.1: BIC scores for regression models that take initial planning time as the dependent variable and incorporate planner predictions or structural parameters (OC and SH). For readability, scores are shown as offsets relative to 110066 (full condition) and 82053 (no-constraint condition).

Category	Planner	full	no-constraint
Baseline	random	625	141
	OC	47	114
	SH	584	**0**
Cognitive Architecture	4CAPS	120	89
Classical Planner	BFS	38	101
	A*	4	83
	GBFS	162	104
Online Planner	LH1	597	78
	LH2	598	80
	LH3	597	79
	LH4	342	70
	LH5	110	88
	LH6	52	95
	LH7	87	110
	A-LH	**0**	87

As expected, the online planners perform better than the classical planners in the no-constraint condition. In the full condition, one of the classical planners (A*) performs relatively well but the best planner for this condition is the adaptive lookahead model. My results for planning time are therefore broadly compatible with the finding in Figure 3 that the adaptive lookahead planner performs well across both conditions.

Table 3.1 reveals, however, that the single best predictor for the no-constraint condition is not a planner but rather the Start Hierarchy parameter shown in Figure 1.3c. It makes sense that participants should respond quickly when there is only one possible initial action (i.e. the instance is completely unambiguous), but common sense and previous work [36] suggest that people's responses are influenced by factors that go beyond Start Hierarchy alone. The strong performance of Start Hierarchy for the no-constraint condition therefore suggests that all of the planners that I evaluated are relatively far from a comprehensive account of human responses to no-constraint condition.

3.6.4 Individual Differences

The analysis summarized by Table 3.1 used individual-level data but did not focus on individual differences. A similar regression approach, however, can be applied to the

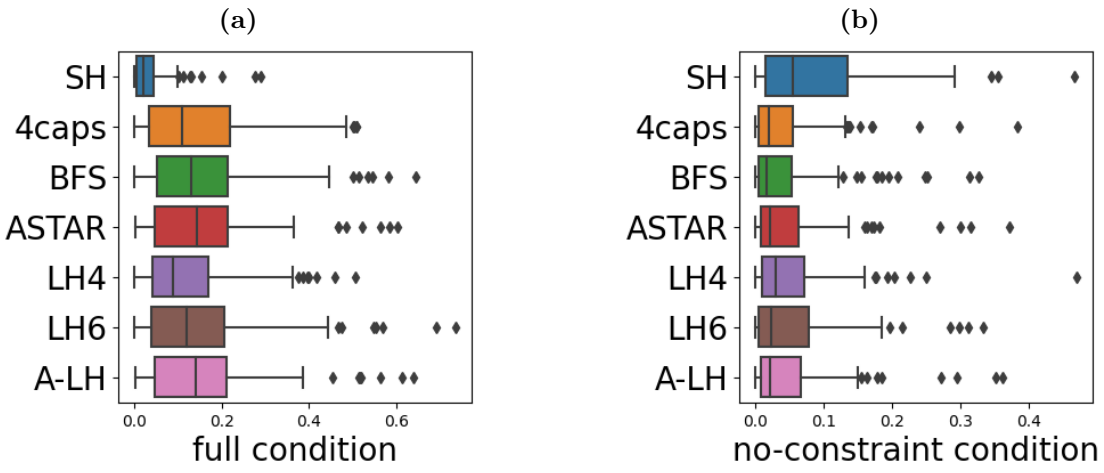


FIGURE 3.5: (a) Individual-level analysis of initial planning times. Panels (a) and (b) show regression scores for the full and no-constraint conditions, and each data point represents an individual participant that is identified as an outlier for that particular planner.

subset of the data provided by a single participant, which yields regression scores indicating the extent to which each planner or structural parameter predicts the responses of that participant. Distributions of these regression scores across individuals are shown in Figure 3.5a and Figure 3.5b. Consistent with Table 3.1, the individual level analysis suggests that the adaptive lookahead and A* planners provide the best account of the full condition, and that Start Hierarchy provides the best account of the no-constraint condition. In the full condition, A* and the adaptive lookahead planner account for the responses of some individuals relatively well (regression scores around 0.6), but in the no-constraint condition no regression score for any individual exceeds 0.5. The results therefore suggest that none of the models provides a good account of individual performance in the no-constraint condition.

3.7 Discussion and Conclusion

I applied a set of planners to the TOL task and evaluated their ability to predict actions and response times collected in a new behavioral experiment. Prior work on the TOL task often asks participants to form a complete plan before acting [36], and in this condition I found that an adaptive lookahead planner provides the best account of both actions and response times. This planner allows the size of the search tree to depend on the difficulty of the current instance, and the good performance of this planner suggests

that people flexibly navigate a speed-accuracy tradeoff when approaching sequential decision-making tasks.

The differences I observed between the full and no-constraint conditions confirm that people's problem solving strategies depend on task requirements, but my planner evaluation did not provide a consistent picture about performance in the no-constraint condition. The adaptive lookahead planner provided the best account of action selection in this condition, but my analysis of response times found that none of the planners was more predictive than a simple structural parameter (Start Hierarchy). It may not be surprising that removing task constraints increases variability and makes experimental data more difficult to model, but my results suggest that more work is needed to develop a satisfying account of human performance in this condition. In this study, I used a regression model to account for the impact of condition and order independent of the current adaptive lookahead planner. However, it is important to note that the planner is highly adaptable and can capture these effects by incorporating adjustable components. For example, the condition effect could be controlled by adjusting the decision threshold, such that a larger threshold in the full condition induces deeper thinking depth. Additionally, the order effect could be modeled as a more accurate heuristic estimation as participants gain more experience. These components could also potentially be adjusted to model the various degrees of suboptimality observed in human problem-solving. Therefore, exploring these possibilities is a crucial future direction in this field of research.

I presented a simple initial analysis of individual participants that revealed substantial variability, and future work can model individual differences more directly by introducing individual-level parameters to the models. For example, the success of the current adaptive lookahead model motivates future versions of the model that allow the decision threshold to vary across individuals.

When using the planning-based approach to model human behavior, it is important to consider not only the planner but also the alignment between human mental representations and the problem representation used by the planner. At least two representations of the TOL task can be considered. My analysis treated pick-and-put as a single action, but an alternative modeling approach treats pick and put as two separate actions. I evaluated both representations and found that both of them led to similar conclusions,

but more targeted experiments may be able to reveal which of the two is closer to the representation used by people. Similarly, my models used the goal-counting heuristic, but I also evaluated other general heuristics derived from widely used relaxations such as the delete-relaxation [21], and found that these alternative heuristics produced similar results in my setting. Future studies, however, can consider experiments that aim to distinguish which of these heuristics provide the best account of human behavior.

This work might be criticized for only considering the initial action and planning time, which is a limited approach to comparing planning algorithms with human responses in sequential decision making. However, my results show that even predicting human initial action or planning time is harder than expected. While the following planning stages are very likely to depend on the initial planning stage, identifying a promising model to mimic human behavior for the initial planning stage is a good start towards a complete model for predicting full observations.

Perhaps the most general message from my work is that the planning-based approach to human problem solving deserves to be revisited. My results suggest that even relatively simple tasks such as the Tower of London continue to present challenges for cognitive models, and combining ideas from both cognitive psychology and AI planning continues to be a promising way to address these challenges.

3.8 Summary

In this chapter, I investigate **RQ1**: "Which algorithm best emulates human responses, including both action selection and response times, in sequential decision-making tasks?" I use automated planning algorithms to simulate human decision-making in terms of action selection and planning time, and found that the novel adaptive lookahead planner performed better than the other algorithms I considered.

Two potential directions for future exploration stand out as particularly noteworthy. Firstly, simulating human behavior encompasses multiple levels, as proposed by Mattar et al. [15]. The initial stage involves action selection, followed by reaction time. Moreover, a more sophisticated model should strive to replicate intricate data such as eye tracking or even more complex datasets like neuroimaging data. While my model may not fully capture processes like eye tracking or neural activity within the human brain,

it could be valuable in the context of human-agent interaction. It serves as a practical tool for agents to approximate and predict human behavior.

Secondly, the conventional approach in computational cognitive modeling involves incorporating specific parameters to account for individual differences, as demonstrated in prior work by Callaway and colleagues [19, 63, 64]. However, my primary aim is to use this model as a general-purpose tool for AI systems targeting diverse populations. The concept of personalized modeling, such as for a personal AI assistant, entails the complex task of tailoring the model and fine-tuning personalized parameters. It presents another interesting challenge which is not the focus of this thesis. For an in-depth discussion of this topic (i.e. how to adapt the adaptive lookahead planner to fit individual's behavior), please refer to Chapter 6.1.3.

Up to this point, we've developed a model (i.e. the adaptive lookahead planner) capable of capturing human action selection and response times in problem-solving tasks. However, a pivotal question remains: is this model useful in the context of human-agent interaction? Can I develop algorithms that leverage this model to deduce and enhance human behaviors to optimize system performance? In the subsequent two chapters, I address these question by applying the adaptive lookahead planner to goal recognition tasks, aiming to address these inquiries. In goal recognition, human-like models are particularly valuable as they allow us to estimate the likelihood of observations. A more accurate model has the potential to yield superior estimations, thereby enhancing the overall performance of a goal recognition system.

Chapter 4

Timing Information in Goal Recognition[1]

4.1 Introduction

Consider, once more, the warehouse setting, where two potential packages await preparation by a human worker. One of these packages consists of a single item, requiring straightforward packaging, while the other presents a more complex packing challenge. In the event that the robot assistant detects a worker thinking for an extended period, it becomes important for the robot to infer the worker's intention, enabling it to offer more effective assistance.

As this example suggests, timing information might be helpful for goal inference. Real-world interactions are embedded in time and timing information is almost always available. Current goal recognition algorithms, however, mostly focus on actions only and rarely take auxiliary information such as timing into consideration [37, 39, 120, 145, 146]. In this chapter, I propose a new goal recognition framework that can exploit observed planning times, and evaluate it using both synthetic and human data.

The problem of goal recognition is the task of inferring an actor's real goal given a sequence of observations and a set of possible goals. Early approaches to this problem often used a plan library to perform goal inference and matched the sequence of

[1]This chapter is adapted from the published article: "Goal Recognition with Timing Information" Proceedings of the International Conference on Automated Planning and Scheduling. 2023

observations with a library of historical observations associated with each goal candidate [38, 147]. Later, Ramirez and Geffner proposed a generative approach that uses planning algorithms over planning models and is known as plan recognition as planning (PRP) [37, 148]. For more detailed description of work in goal recognition, please refer to chapter 2.2.

A small amount of work in AI and cognitive science has explored how auxiliary information can be used to infer the mental states of others. Singh et al. [149] used gaze information for intention recognition and found that gaze can help to reveal the hidden goals of players in a boardgame. Gates et al. [150] developed a Bayesian model that aims to capture how people use response times when inferring the preferences of an actor who is observed to make a single decision. In this chapter, I generalize the same underlying idea by exploring how timing information can be used in situations where actors generate rich sequences of actions, not just one-shot decisions. Perhaps closest to our own approach is the work of Avrahami-Zilberbrand et al. [151], who developed a plan-recognition algorithm that incorporates constraints on action durations. My work also highlights the role of time but focuses specifically on planning times that reflect the effort exerted by the actor when selecting actions. Fagundes et al. [152] also talk about timing constraints, but do not use these constraints to disambiguate goal recognition.

Figure 4.1 and Figure 4.2 illustrates two cases in which planning times are useful for goal recognition. In the Sokoban example (Figure 4.1), the current position of the worker is shown in color and the grey workers show the trajectory the worker followed to reach this position. The actor is a real-time planner that performs a look-ahead search using Manhattan distance as a heuristic, and because the computational resources of the actor are limited it is not guaranteed to choose the optimal trajectory. Given the information in Figure 4.1, goals A and B may seem equally likely because the observed trajectory is consistent with optimal paths to both goals. But if we observe in addition that the actor spent a relatively long time at the position shown, B now seems the more likely goal because A is easily achieved with a single push to the left, whereas the actor has to push the box away and then back to achieve B.

In Figure 4.1, timing information breaks a tie between two goals that seem equally likely based on actions alone, but there may also be cases where timing information reverses the conclusion that would follow from actions alone. Figure 4.2 shows an example based

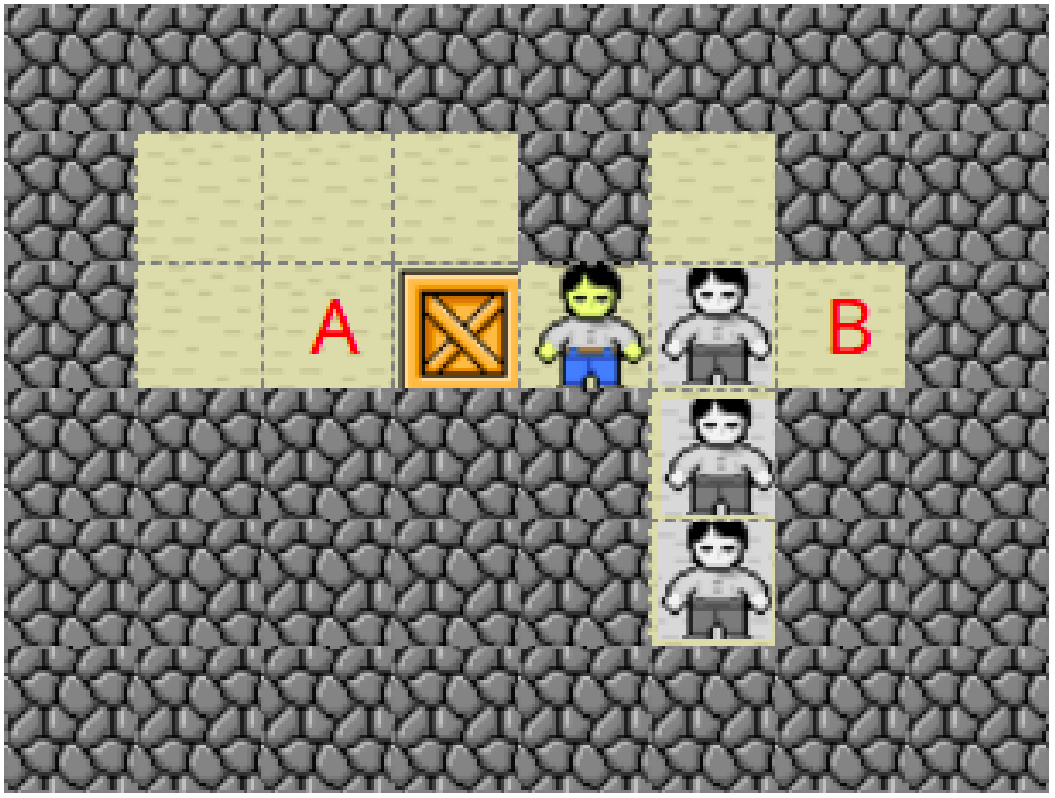


FIGURE 4.1: Timing information can break a tie between two goals. In this Sokoban example, observing the actor stop and think at the position shown with blue jeans suggests that the actor's goal is to push the box to B rather than A.

on a navigation task. Here the observed action sequence suggests that B is the likely goal because this sequence is consistent with an optimal path to B but not A. But if we see that the actor spends a long time at the location shown, we might conclude that A is the actual goal because there would be no reason for the actor to stop and think if the goal were B rather than A.

Because timing information has received little attention in the literature, standard goal recognition benchmarks do not include this information. Most existing agent models do not produce useful planning times because they either allocate a constant amount of planning time for each step or do not consider this factor at all [120, 145]. I, therefore, use an adaptive lookahead planner (A-LH) inspired by human behaviour to generate human-like timing information along with action sequences for standard goal recognition benchmarks. I also use the same planner to develop timing-sensitive goal recognition algorithms.

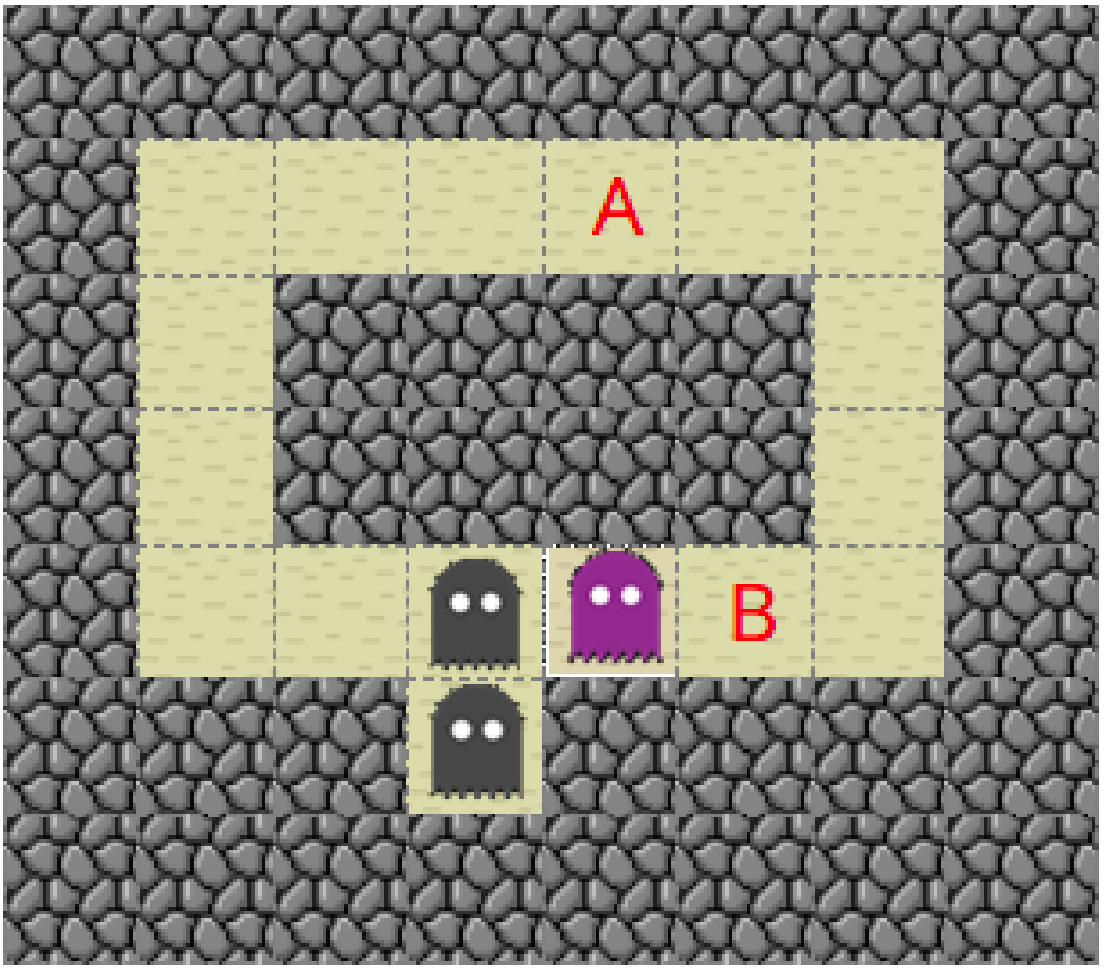


FIGURE 4.2: Timing information can reverse the inference that would follow from actions alone. In this navigation example, a protracted pause at the position shown in purple suggests that the goal may be A rather than B.

To preview some of my results, I find that the extent to which timing information helps in goal recognition depends on how closely the agent model assumed matches the agent actually generating the observations. The adaptive lookahead planner draws on the extensive response-time modelling literature in cognitive science [17, 153], but it is not intended to capture all of the reasoning strategies that humans may use. Instead, the planner relies on two simple assumptions about human planners: (a) people carry out a forward search to make a decision (i.e. they are not reflex agents), and (b) planning time depends only on the current state and the true goal. Assumption (b) does not assume that people only consider the current state, as humans typically anticipate the consequence of future moves. Instead, the assumption is that the planning time for one move does not depend on the planning times for previous moves.

This chapter makes a sequence of four contributions. First, I formally introduce a goal

recognition framework that incorporates timing information and a novel goal recognition algorithm that can exploit this information. Second, inspired by the cognitive science literature, I develop a real-time agent model, and use it to generate observations with timing information for standard goal recognition benchmarks. Third, I use existing goal recognition datasets to show that timing information can be helpful when my framework can exploit an accurate model of timing. Finally, I show that the proposed goal recognition algorithm can exploit timing information in sequences generated by humans, and also accounts for human inferences in a behavioral study of goal recognition.

The next section reviews the relevant literature on goal recognition, online planning agents and response time modelling. I then present the adaptive lookahead planner, formulate the problem of goal recognition with timing information, and present an algorithm that addresses this problem. Finally, I present a set of synthetic and behavioural experiment results and discuss prospects for future work.

4.2 Background

4.2.1 Theory of Mind and Goal Recognition

For a comprehensive introduction to Theory of Mind and goal recognition, please see Chapter 2. The subsequent paragraphs provide a brief review.

People's ability to infer the mental states of others is known as *Theory of Mind* [154], which is a classic topic in cognitive science. Many behavioural and neural studies have been done in this area while its computational basis has been extensively explored in the last two decades [117, 155]. In recent years, industry labs have paid increasing attention to this field [156, 157] because AI systems that interact with humans (e.g. self-driving cars) must be able to figure out the goals and intentions of human users.

In the literature on computational cognitive science, Baker *et. al.* developed a Bayesian model of theory of mind and showed that it makes human-like judgements when inferring people's goals and beliefs [117]. Jara-Ettinger [3] further suggested that theory of mind can be formalised as inverse reinforcement learning and involves inferring people's internal model of the world and their reward functions given some observed actions.

The automated planning community has proposed a variety of models for efficiently solving the goal recognition problem [37, 39]. It is still unknown whether these models are able to account for human goal-recognition abilities but these models can inspire hypotheses about how people carry out goal recognition. Integrating ideas from cognitive science and automated planning literature is therefore a promising way to develop computational models of human goal recognition.

4.2.2 Online Planning Algorithms and Suboptimal Behavior

When dealing with complex tasks with limited reasoning time, it is often impractical for both humans and agent models to find a full plan from the current state to a goal state. Unlike classical planning algorithms, online planning algorithms do not aim to find a full plan but rather focus on choosing which single action should be executed at the current state. A prominent approach used to develop online planning algorithms is Monte Carlo Tree Search (MCTS), which has achieved striking success at playing Go [68]. MCTS has also been explored as a model of human problem solving [59, 61].

Current algorithms in the field of goal recognition usually assume full rationality, i.e. optimal behaviour for both the actor model and the observer model [37]. In contrast, the cognitive literature suggests that people often depart from optimality [117, 150]. Masters and Sardina [120] explore how goal-recognition systems can reason about irrational agents, but their approach has not yet been directly connected with research in cognitive science.

4.2.3 Response-time Modeling

An extensive literature in psychology treats response times as a sign of underlying cognitive mechanisms. A prominent approach in this area focuses on one-shot decision making, and assumes that the decision-maker continually samples evidence about the available response options until some decision criterion is reached. This "evidence accumulation" framework is widely used to account for both reaction times and choice probabilities [17].

There are a variety of evidence accumulation models that make different assumptions, and in recent years psychologists have explored which model gives the best account of

behavioural data from perceptual decision-making tasks [17, 153]. Little work has been done, however, in applying the evidence accumulation framework to sequential decision-making problems. Solway and Botvinick [58] take a step in this direction by showing how an evidence integration mechanism can be combined with a model-based tree search. Their work, however, focuses on simple two-step plans that are significantly simpler than those used in standard AI planning benchmarks. Ho et al. [158] also consider sequential decision-making problems, and use value iteration to account for human reaction times.

4.3 Framework

In this section, we first describe an adaptive lookahead planner that aims to produce human-like planning times by incorporating concepts from the evidence accumulation literature. This planner closely aligns with the A-LH model introduced in Chapter 3, with the notable inclusion of a distinct stopping mechanism to accommodate non-deterministic thinking time. Additionally, it adopts an average update approach instead of a maximal update, a modification aimed at better mirroring the planning characteristics exhibited by humans. We then propose a formal framework for modelling and solving the problem of goal recognition with timing information.

4.3.1 Adaptive Lookahead Planner

The goal recognition algorithms proposed later require models of planning times, and the datasets used to evaluate these algorithms must include planning times in addition to actions. Standard AI planning algorithms do not generate human-like response times, and we therefore developed a new online planning algorithm named adaptive lookahead planner inspired by ideas from the evidence accumulation literature [17, 153].

Given a problem with a goal hypothesis g and start state s_0, we carry out the tree search described below until the goal state is found or the stop signal is triggered (see Algorithm 2). The search tree starts with the current state s_0, or a subtree with root s_0 from the last planning step if a memory mechanism is included, and the algorithm traverses the tree using the UCB policy [141] until the leaf node is reached (lines 3-5). If the leaf node is the goal state g, the tree search process stops (lines 8-9). Otherwise, the node is expanded by generating all possible successor states except those states visited

Algorithm 2 Adaptive Lookahead Planner 2

Input: goal hypothesis g, tree T from last run (if memory mechanism) or single node tree (root node is the current state s_0)

Output: Number of expanded nodes num (used as a proxy for the planning time for this step), Best subtree T_{best} (or the sequence of actions to g if found)

1: Let $tree = T$, $state = s_0$, $num = 0$
2: **while** stop criterion does not hold **do**
3: **while** *state* has been expanded **do**
4: $state = TreePolicy(state)$
5: **end while**
6: $num \leftarrow num + 1$
7: $T = Expand(state, T)$
8: **if** *state* is g **then**
9: **return** $num, ExtractSolution(T, g)$ {Extract path from root node to g. The output in this case is the planning time and the solution }
10: **end if**
11: $state \leftarrow s_0$
12: **end while**
13: **return** $num, T_{best} = ChooseBestChild(T)$

previously to avoid generating repeated states. Each successor state is initialized with the estimated cost-to-go and values of all ancestor nodes are then updated by averaging the obtained values of all visits passing through the node (line 7).

After each iteration (expansion), the stop trigger is executed to check if enough information has been collected to make the decision (line 2). The probability of triggering the stop signal is calculated as

$$P_{stop}(s_0, n) = \frac{n}{n + I(s_0)\gamma \exp(-n/I(s_0))}, \tag{4.1}$$

where n is the number of iterations so far and γ is a parameter that controls the depth of the trajectories considered. The state importance $I(s)$ is defined as:

$$I(s) = \frac{v_{s,a}}{(1+\beta)v_{s,a'} - v_{s,a}}. \tag{4.2}$$

Here $v_{s,a}$ and $v_{s,a'}$ denote the cost estimates that result from choosing the best applicable action a and second-best applicable action a' towards a given goal from state s. The denominator of Equation 4.2 is therefore based on the estimated difference in cost between the top two applicable actions, and a small constant parameter β is included in order to avoid zero denominators when the top two applicable actions have the same

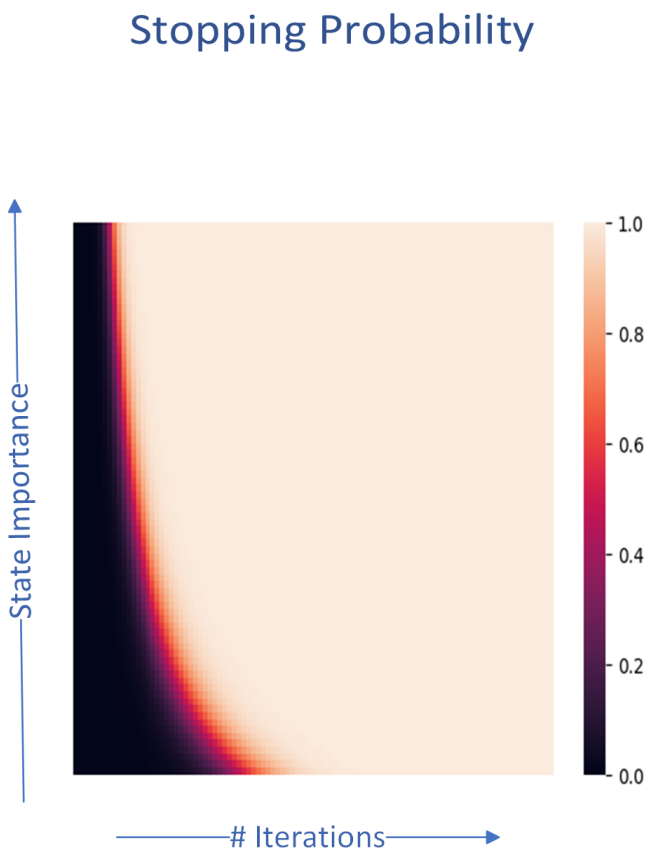


FIGURE 4.3: Stopping probability in Equation 4.1 ($\gamma = 10000$)

costs. When the tree search stops, the agent model returns the number of iterations as the planning time for the current state.

Equation 4.1 specifies the formulation of a probability distribution for generating a stopping signal, which captures the idea that the actor will spend more planning time on states that have two or more applicable actions that seem equally good (or nearly so) while acting relatively fast in states with a dominating action. For an illustration of the stop probability function, refer to Figure 4.3, which depicts how the probability of stopping varies with certain parameters. This approach is broadly consistent with the evidence accumulation literature, which suggests that people tend to keep gathering evidence until one option emerges as the winner [17, 153]. Moreover, similar ideas of state importance have been used to summarise state trajectories over Pacman games [159].

Now I show how the Adaptive Lookahead planner generates human-like planning time using a value-based example. Here I simply assume the cost of action is 0 and the

value denotes the estimated expected utility of the node. In this configuration the best successor state would be the maximal value rather than minimal value, thus the definition of state importance becomes

$$I(s) = \frac{v_{s,a}}{(1+\beta)v_{s,a} - v_{s,a'}}. \tag{4.3}$$

.

In this example we assume $\beta = 0$ and $\gamma = 10$ for simplicity. We start with the current state s_0 (fig 4.4a). In the first iteration we expand the start node by generating all the applicable actions a_1, a_2, a_3 and the initial values by the heuristic function are $5, 3, 1$ respectively (fig 4.4b). Now the state importance $I(s_0) = \frac{5}{5-3} = 2.5$, so the stopping probability $P_{stop} = \frac{1}{1+2.5*10*exp(-1/2.5)} \approx 6\%$. Assume we continue to next iteration. In this iteration we choose expanding s_1 by generating two successor states s_4, s_5 and the value of s_1 is updated to reflect the average gain of all visits $(5+5+8)/3 = 6$, noting that it also includes the value obtained (i.e. 5) in the initial visit (fig 4.4c). Now the state importance $I(s_0) = \frac{6}{6-3} = 2$, and the stopping probability increases to $P_{stop} = \frac{2}{2+2*exp(-2/2)} \approx 21\%$. Assume we still continue, and after next iteration (fig 4.4d) the stopping probability becomes $P_{stop} = \frac{3}{3+1.5*10*exp(-3/1.5)} \approx 60\%$. In the next iteration when we expand s_3 we will use a_3 as the second best action rather than a_2 (fig 4.4e), thus the state importance increases to $6/(6-3) = 2$ and the stopping probability becomes $P_{stop} = \frac{4}{4+2*10*exp(-4/2)} \approx 60\%$. If we still need to continue (very unlucky), we will use UCB rule to choose s_1 to visit then s_5 (fig 4.4f) and this time the state importance remains 2 and the stopping probability becomes 75%. Then assuming the stop signal is triggered we can stop and the planning time for this state is 5.

4.3.2 Problem Formulation

We now formalize the problem of Goal Recognition with Timing information (GRT). For simplicity, we assume a fully-observable deterministic environment, but the framework and goal recognition algorithms introduced later can be extended to partial observability and/or probabilistic settings by choosing appropriate cost-to-go estimators.

The planning domain is a planning problem without a goal, which can be defined as follows.

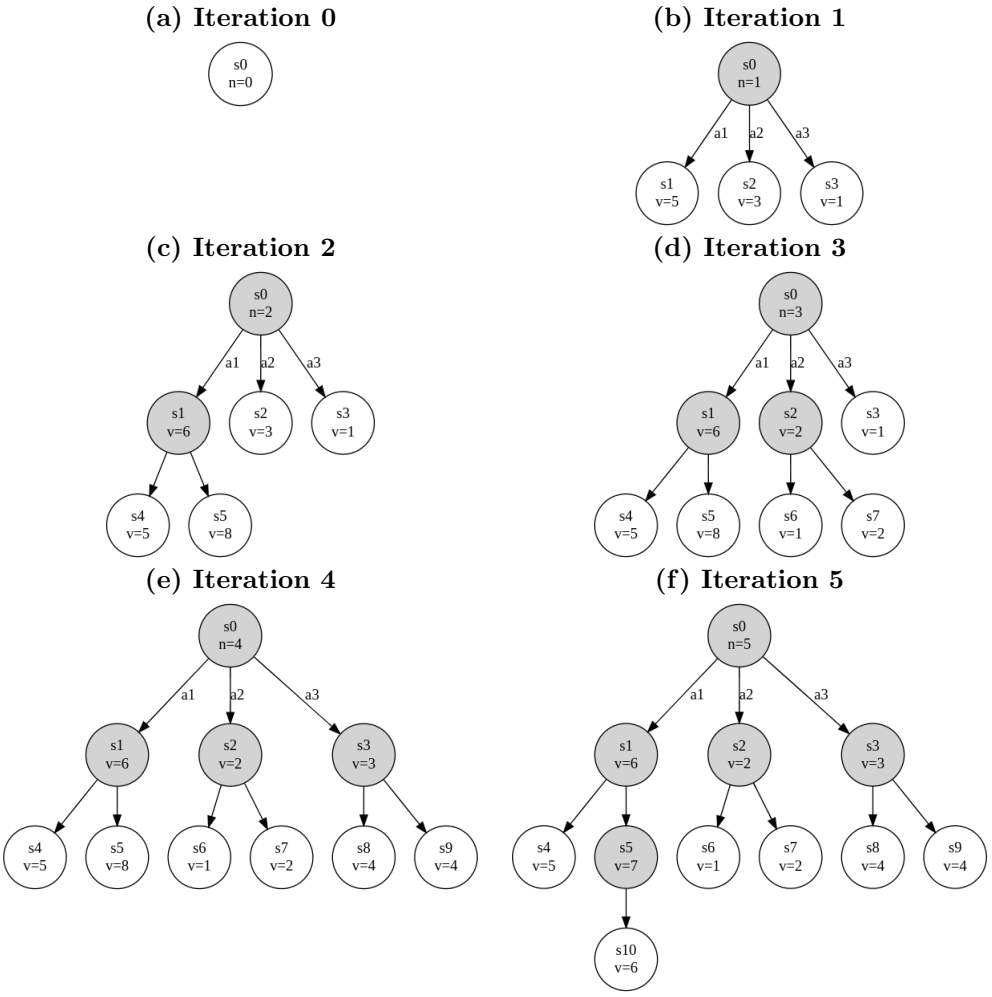


FIGURE 4.4: The Adaptive Lookahead planner run on a value-based example. The dark circles represent the node expanded and the light circles represent the node generated but not expanded yet. The value v of the node is initialized as some heuristic function of the corresponding state and then updated to reflect the average value of all visits passing through the node.

Definition 4.1. A planning domain $D = \langle S, s_0, A, f, c \rangle$ consists of a finite set of discrete states S, an initial state $s_0 \in S$, a finite set of actions A, a state transition function $f : S \times A \to S$ that maps a state-action pair (s, a) into another state s' and a cost function $c : S \times A \to \mathbb{R}$ which specifies the cost $c(s, a)$ incurred when applying action $a \in A$ on state $s \in S$.

A planning problem $D[g]$ is instantiated by adding a goal g to the planning domain D. For a goal recognition problem, we have a set of possible goals along with a sequence of observations in a planning domain.

Definition 4.2. A goal recognition problem with timing information (GRT) is a tuple $\langle D, G, Prior, O \rangle$, where $D = \langle S, s_0, A, f, c \rangle$ is the planning domain, $G = \{g_1, g_2, ..., g_n\}$

is a set of possible goals for the planning domain, *Prior* is the prior probability over G, and O is a sequence of observations $\langle a_0, t_0\rangle, ..., \langle a_m, t_m\rangle$, where $a_i \in A$ is an action, and t_i is a non-negative real number denoting the planning time used to select a_i for execution.

The key difference compared to the classical goal recognition setup introduced in chapter 2 is that we include planning times in the observation sequence.

4.3.3 Timing-sensitive Goal Recognition Algorithm

We assume that actions and planning times only depend on the current state and the true goal (Markovian), and that planning time and action are conditionally independent on states and goals. The first assumption suggests individuals don't need to remember past states; the current state and the goal contain all the relevant information needed to make a decision. The second assumption implies that planning time and the chosen action are determined independently. While they may not be entirely accurate, these assumptions offer a helpful approximation that allows us to estimate the likelihood. Using a uniform prior, we can decompose the likelihood $P(O|g)$ as :

$$\begin{aligned} P(O|g) =& P(\langle a_0, t_0\rangle, ..., \langle a_m, t_m\rangle|g) \\ =& \prod_{j=0}^{m} P(t_j|g, \langle a_0, t_0\rangle, ..., \langle a_{j-1}, t_{j-1}\rangle) \\ & P(a_j|g, \langle a_0, t_0\rangle, ..., \langle a_{j-1}, t_{j-1}\rangle, t_j) \\ =& \prod_{j=0}^{m} P(a_j|g, s_j) \prod_{j=0}^{m} P(t_j|g, s_j) \end{aligned}$$

We call the product $\prod_{j=0}^{m} P(a_j|g, s_j)$ the *action component* and $\prod_{j=0}^{m} P(t_j|g, s_j)$ the *timing component*. The next section explains how we estimate both components, and I then discuss how these components are combined to produce a GRT solution.

Action Component

I follow the PRP approach proposed by Ramírez and Geffner [37] to estimate $\prod_j P(a_j|g, s_j)$. Rather than estimating the probability for each step of the observation sequence, their

approach approximates the full sequence directly as $\prod_j P(a_j|g, s_j) \propto \exp(v^*_{s_0}(g) - v^*_{s_0}(g, O))$, where $v^*_{s_0}(g)$ denotes the optimal (thus smallest) cost-to-go from start state s_0 while $v^*_{s_0}(g, O)$ represents the cost of the best path consistent with current observations O. Their approach uses the full observation trajectory and is computationally expensive, as optimal planning is hard unless approximated with suboptimal planners or suitable relaxations [100]. Thus, we propose a novel method, namely **real-time PRP** via simulation through the agent model we proposed. Compared to the original PRP, real-time PRP assumes the problem to be Markovian and considers each step independently:

$$\prod_j P(a_j|g, s_j) \propto \prod_j \exp(v_{s_j}(g) - v_{s_j}(g, a_j)),$$

where $v_{s_j}(g)$ denotes the approximation of cost-to-go from the state s_j and $v_{s_j}(g, a)$ denotes the approximation of cost-to-go if action a is taken on s_j. This allows for real-time performance instead of computing a full plan as in PRP.

Timing Component

We define expected planning time for state s_j given goal g as $t^*(s_j, g)$, and decision cost (which captures the total effort needed by an actor to choose the move at state s_j when pursuing goal g) as $t(s_j, g)$. In this project, we assume for simplicity that the decision cost $t(s_j, g)$ is identical to t_j, the time recorded in the observation sequence.

We use $\exp(-|t^*(s_j, g) - t(s_j, g)|) = \exp(-|t^*(s_j, g) - t_j|)$ to estimate $P(t_j|g, s_j)$. Two approaches are proposed to approximate the expected planning time $t^*(s_j, g)$:

- **Agent-based**. Given goal g, $t^*(s_j, g)$ is defined as the number of iterations to make the decision at state s_j via simulation by the adaptive lookahead planner described above.
- **Importance-based**. In this approach, $t^*(s_j, g)$ is estimated directly by the state importance $I(s_j)$ defined in the adaptive lookahead planner shown in Equation 4.2.

Note that $t^*(s_j, g)$ and $t(s_j, g)$ may be measured on different scales. $t(s_j, g)$ is typically measured in seconds, whereas $t^*(s_j, g)$ is generated by the timing component of the model and has iterations or importance as units. To map between these different scales, we normalize $t^*(s_j, g)$ by scaling its sum over s_j to match the sum of $t(s_j, g)$.

Combining Components

I use two approaches to combine the action and timing components. The first one adds evidence from the two components and uses the resulting sum to rank the goals. Let $p_t(g)$ be $\log \prod_j P(t_j|g, s_j)$ and $p_a(g)$ be the $\log \prod_j P(a_j|g, s_j)$ for the potential goal g. Then the combined probability of goal g is $\frac{p_t(g)}{\sum_j p_t(g_j)} + w\frac{p_a(g)}{\sum_j p_a(g_j)}$ where w is an adjustable balance factor.

The second approach uses the evidence from the action component to rank the goals, and relies on the timing component only to break ties. In this approach, the timing component cannot reverse the inference suggested by the action component, and can contribute only when the action component does not provide enough information to infer a single most likely goal.

4.4 Synthetic Experiment

This section describes an experiment that uses standard goal-recognition data sets to evaluate whether timing information can improve the performance of goal-recognition algorithms. Existing goal recognition algorithms return the set of most likely goals, and accuracy is typically used as an evaluation metric. Here I use fractional ranking to evaluate the extent to which timing information helps distinguish between equally likely goals. Fractional ranking generates the same mean rank as ordinal ranking but allows for ties. For example, if the likelihoods of 4 potential goals were 0.8, 0.5, 0.5, 0.2, the ordinal ranks would be 1,2,3,4 and the fractional ranks would be 1,2.5,2.5, 4.

For each instance, the performance of an algorithm is measured by the fractional rank assigned by the algorithm to the true goal. The performance on an entire domain is measured by the average performance across all instances of that domain. Given the average fractional rank r_D for an algorithm on each domain D, the overall normalized score for that algorithm is $\sum_D \frac{2r_D}{k_D+1}$, where k_D is the number of potential goals in domain D. Note that $\frac{k_D+1}{2}$ is the expected fractional rank achieved by a random algorithm in the domain D. Overall, lower fractional ranks or normalized scores indicate better performance.

4.4.1 Experiment Configuration

I evaluated goal recognition algorithms on 10 domains from the goal recognition dataset of Pereira et al. [39]. Because this dataset does not include timing information, we used the adaptive lookahead planner to supplement the trajectories with times: for each state s, I ran the adaptive lookahead planner (without memory mechanism) given the real goal g and took the average number of iterations over 100 runs as the planning time for that state. The planning time was recorded while the action chosen by the planner was discarded to ensure that the trajectories remain consistent with the original dataset.

I use the satisfying planner DUAL-BFWS [85] to approximate the optimal cost-to-go in the goal recognition algorithm PRP [37]. For initializing node values in the agent model and computing the importance-based timing component, I use the Fast Forward heuristic function h_{ff} [84]. All experiments were conducted on 4 servers each running Intel®Xeon®Gold 6138 CPU @ 2.00GHz with 4 CPUs, and 8GB of RAM each.

All action costs were set to 1. Constants in the agent model ($\gamma = 10000, \beta = 0.2$) were chosen manually so that the model generated human-like response times in navigation tasks like Figure 4.2. Except when mentioned otherwise, the observation ratio is set to 0.25, which means that I use the first quarter of observations in a trajectory as the input to the goal recognition algorithms. The adjustable weight w is set to 1, which means that we weigh the action and timing components equally.

4.4.2 Experiment Results

Table 4.1 shows the performance of 8 goal-recognition algorithms along with a random baseline.

Action-only Algorithms

Columns rtPRP and PRP in Table 4.1 show the results of real-time PRP and standard PRP (both without a timing component). In Depots, Dwr, Miconic, Driverlog, Ferry, BlocksWorld and Logistics, real-time PRP outperforms PRP. In Sokoban

and EasyIpcGrid, PRP performs better while in IntrusionDetection, both approaches have the same performance. Overall, the normalized score for rtPRP is 6.06, which is slightly better than the score of 6.22 achieved by PRP.

These results suggest that real-time PRP performs similarly to PRP, which implies that computing a full solution might not be necessary for goal recognition even when considering the action component alone.

Effect of Timing Components

When supplied with the agent-based timing component, rtPRP-a and PRP-a receive overall scores of 3.76 and 3.82 respectively, while importance-based timing components increase these scores to 7.48 and 7.10. The scores for importance-based timing components are worse than those for the corresponding algorithms without timing components (6.06 and 6.22). These results indicate that an accurate timing component can substantially increase the performance of both PRP and rtPRP, but that incorporating evidence from an inconsistent timing component using a sum can be harmful.

Using the timing component as a tiebreaker (AF-a) performs worse (4.43) than the sum of evidence algorithms. On the other hand, AF-i (6.16) is slightly better than the action-only algorithm PRP (6.22). These findings imply that the agent-based timing component can sometimes reverse incorrect inferences made by the action component alone, while even importance-based timing components can be helpful for breaking ties between goals. They also suggest that non-linear evidence combination strategies are likely to be superior to the simple sum used by rtPRP-i and PRP-i.

Observation Ratio

To explore whether timing information is especially valuable in scenarios with relatively few observed actions, I ran rtPRP / rtPRP-a on BlocksWorld with observation ratios set to 0.25, 0.5, 0.75 and 1.

Table 4.2 shows that given the timing goal recognition dataset, rtPRP-a has the largest performance boost when fewest observations are available. As expected, timing information appears to be especially valuable when the information conveyed by the action trajectory is relatively minimal.

Algorithm	rtPRP	rtPRP-a	rtPRP-i	PRP	PRP-a	PRP-i	AF-a	AF-i	Random
DEPOTS	4.61	**2.46**	5.14	4.96	3.36	5.32	3.71	5.09	5.5
MICONIC	1.48	**1.15**	1.38	2.18	1.45	1.55	1.90	1.85	3.5
DWR	2.98	**1.57**	3.70	3.38	2.04	3.52	2.50	3.38	3.5
SOKOBAN	3.52	2.05	3.55	2.02	**1.23**	2.50	1.41	2.09	3.5
EASYIPCGRID	3.58	1.74	3.45	3.07	1.44	3.73	**1.31**	3.21	5.5
DRIVERLOG	2.93	1.71	2.98	3.14	**1.54**	3.21	2.18	3.04	3.5
INTRUSIONDETECTION	1.99	3.16	2.13	1.99	3.16	2.13	**1.93**	2.13	8.83
FERRY	1.71	**1.14**	2.64	2.04	1.39	2.52	1.82	2.02	4
BLOCKSWORLD	4.43	2.83	10.08	5.84	**2.54**	9.51	2.88	5.76	11
LOGISTICS	2.18	1.33	4.33	2.36	**1.23**	3.51	1.67	2.33	5.5
Normalized Score	6.06	**3.76**	7.48	6.22	3.82	7.10	4.43	6.16	10

TABLE 4.1: Performance of eight goal recognition algorithms on the timing goal recognition dataset: real-time PRP (rtPRP), real-time PRP with agent-based timing component (rtPRP-a), real-time PRP with importance-based timing component (rtPRP-i), PRP, PRP with agent-based timing component (PRP-a), PRP with importance-based timing component (PRP-i), action first with agent-based timing component (AF-a) and action first with importance-based timing component (AF-i). The best algorithm for each domain is shown in bold. Both AF-a and AF-i use PRP as the action component.

Ratio	Quarter	Half	Three-quarter	Full
rtPRP	2.53	1.5	1.17	1.03
rtPRP-a	1.67	1.17	1.13	1
Difference	0.86	0.33	0.04	0.03

TABLE 4.2: Performance of rtPRP-a and rtPRP on BLOCKSWORLD with different observation ratios.

4.4.3 Discussion

These results suggest that if we want to take advantage of timing information, then we have to access an accurate timing model or at least a good approximation. My experimental results are in line with the findings in theory of mind [154]: if you can construct an accurate model of an actor's mind, then you stand a good chance of correctly inferring their intentions. On the other hand, an inaccurate model is likely to lead to faulty inferences about others.

One possible criticism of my synthetic experimental setup is that the algorithms with the agent-based model timing component rely on the same mechanism used to generate the timing data, and it is therefore not surprising that timing information turns out to be useful to infer the real goal. The next section addresses this concern by demonstrating that the agent-based timing component is still useful when goal inference is performed on human data. My results for synthetic data, however, still make a useful point: they demonstrate that timing information can be used to distinguish between candidate goals

that are not distinguishable based on action sequences alone (as in Figure 4.1), and can even reverse weak inferences based on action sequences alone (as in Figure 4.2).

Over the past decade, several goal recognition algorithms have been developed based on PRP that outperform the original PRP in certain conditions [39, 160]. These alternatives may perform slightly better than PRP in Tables 4.1 and 4.2, but this would not affect my main conclusions. For the behavioral experiments described in the next section, these alternatives would yield the same goal inference as PRP because the action trajectories provide no information about the goal.

4.5 Behavioral Experiments

The major question left open by our synthetic experiment is whether timing information can still be exploited when the process generating planning times is not fully known. In real-world settings, for example, we might aspire to make inferences about the goals of human actors even in the absence of a veridical model of human planning. I therefore developed two behavioral experiments to explore whether the adaptive lookahead planner matches humans closely enough to allow rtPRP-a to exploit timing information when inferring the goals of humans.

4.5.1 Problem-solving Experiment

My first experiment collected human actions and planning times on a series of Sokoban problems. We used these data to ask whether the adaptive lookahead planner can generate human-like planning times, and whether timing information can be exploited when inferring the goals of humans. Our experiment was carried out with approval from Human Ethics Advisory Group at the University of Melbourne.

Experiment Configuration

50 participants (21 females and 29 males with a median age of 27) were recruited using Prolific and asked to complete 24 Sokoban instances each.

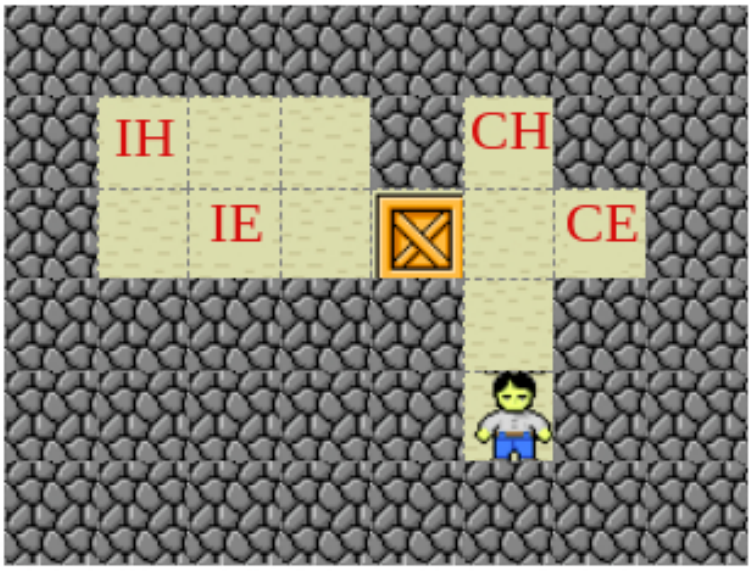


FIGURE 4.5: GRT set 2. One of the 4 potential goal positions is shown in the problem-solving experiment, and the resulting timing is used for goal recognition.

Sokoban is a classic puzzle game where the player must push boxes to designated locations while navigating a maze-like environment. The goal is to successfully move all the boxes to their targets without getting stuck or blocking the path. For simplicity, all of my instances included a single box only. The 24 instances were designed to fall into 6 sets, where each set includes 4 different goal positions located on the same map configuration. One such set is shown in Figure 4.5. The presentation order of all 24 instances was randomized for each participant.

Within each set, the 4 goal positions were chosen as follows. A goal is deemed *intuitive* if the first box push on an optimal path to the goal reduces the distance between the box and the goal, and *counter-intuitive* otherwise. In Figure 4.5, goals IE and IH are intuitive but goals CH and CE are not. Of the two intuitive goals, IE denotes the "easier" goal and IH the "harder" goal, which in some instances can be unreachable. The difficulty is formalized based on the number of nodes expanded by the A* algorithm. Similarly, CH and CE denote the harder and easier of the two counter-intuitive goals. Choosing goals in this way was inspired by results from the psychological literature suggesting that people tend to spend more time planning when the solution length is long and when the solution involves counter-intuitive moves [52, 104].

Human planning times are typically highly variable, and to minimize the variance we use only the initial action and initial planning time to generate goal recognition instances

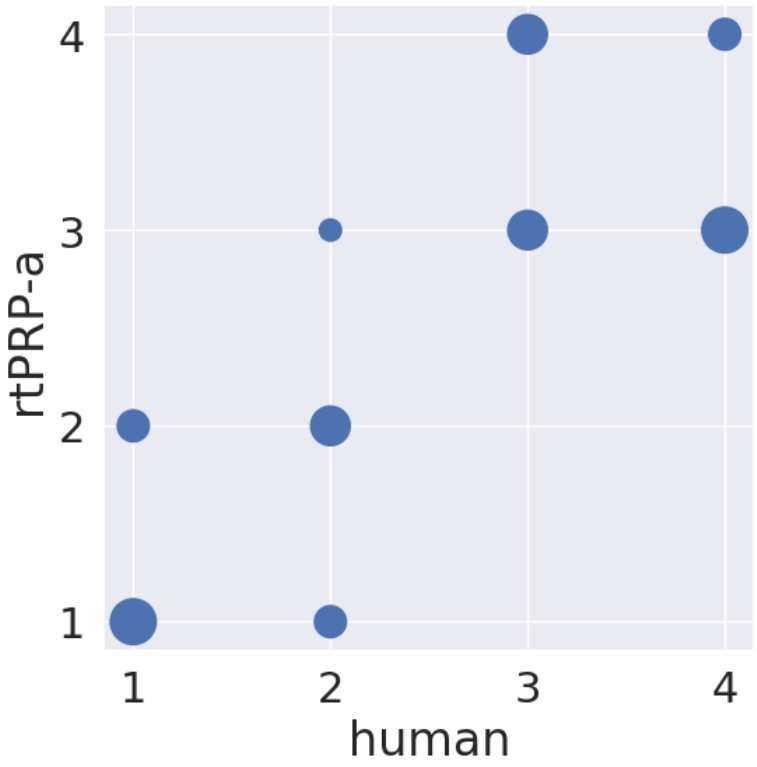


FIGURE 4.6: Comparison between human initial planning time rank and agent model prediction rank within each group. All instances have similar ranks, with a maximum difference of 1.

with timing information (GRT). I generate 4 separate GRT instances for each set, and each GRT instance includes all four goal positions (IE, IH, CE, and CH) as candidates. All instances were designed so that the first move is forced: in Figure 4.5, for example, the agent has no option except to move up on the first move. As a result the initial action provides no information about the goal position, but the time taken before this action is potentially informative.

Results and Discussion

Some of the "hard" goals in the task are actually unachievable, including goals IH and CH in Figure 4.5. I used rtPRP-a as a goal recognition algorithm with both timing and action components and rtPRP as an action-only algorithm.

First, I asked whether rtPRP-a generates human-like planning times. Within each set, I ranked the 4 instances separately by human planning time and by the prediction of rtPRP-a. Figure 4.6 compares these ranks. Most instances lie along the diagonal, which means that rtPRP-a and humans both give the same rank to those instances. When the

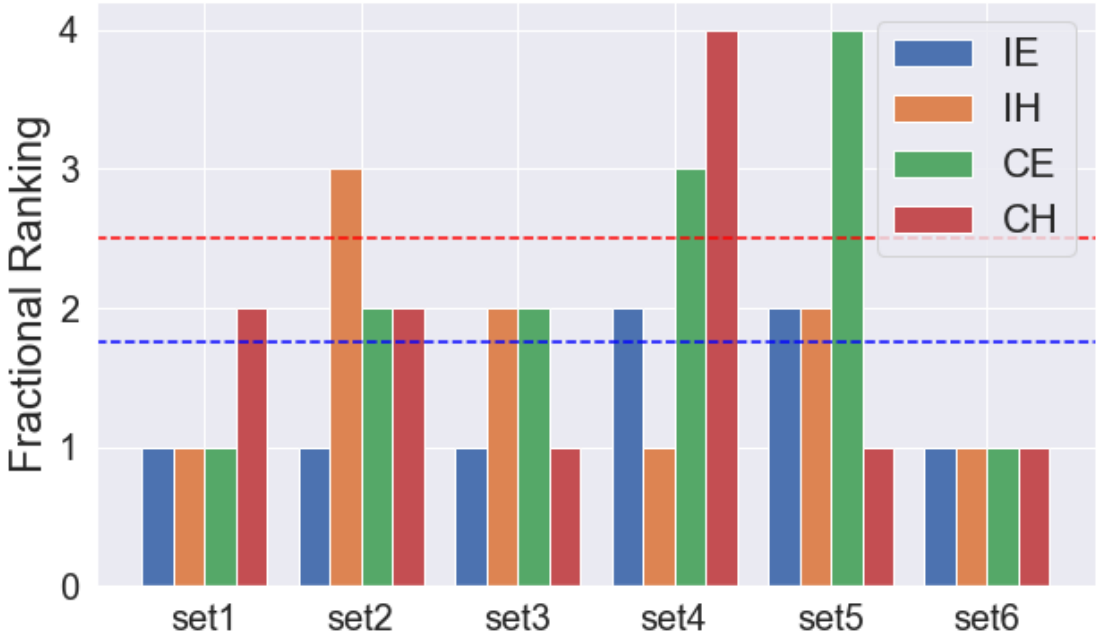


FIGURE 4.7: Performance of rtPRP-a on all GRT instances. The red dotted line denotes the performance of the rtPRP algorithm (2.5) and the blue dotted line denotes the average performance of rtPRP-a (1.75).

rtPRP-a ranking departs from the human ranking, the rank difference for any instance is no more than 1.

I then applied rtPRP-a to the goal recognition task. Because the observation sequences include a single action only, our previous method for aligning t (measured in seconds) with t^* (measured in iterations) no longer applies. I therefore align the two by using ranks relative to the entire set of 24 instances. For example, the median t across this set is mapped to the median t^*.

As presented in Figure 4.7, when choosing among 4 possible goals, rtPRP achieves an average fractional rank of 2.5 (the same as random choice) because by design the initial action is uninformative about the goal. rtPRP-a achieves an average ranking of 1.75, and a paired t-test suggests that the improvement with respect to rtPRP is statistically significant ($t(23) = -3.89, p < 0.001$).

Even though the adaptive lookahead planner is at best a coarse approximation of the strategies used by the experimental participants, my results suggest that this approximation is good enough to be usefully incorporated into my goal recognition framework.

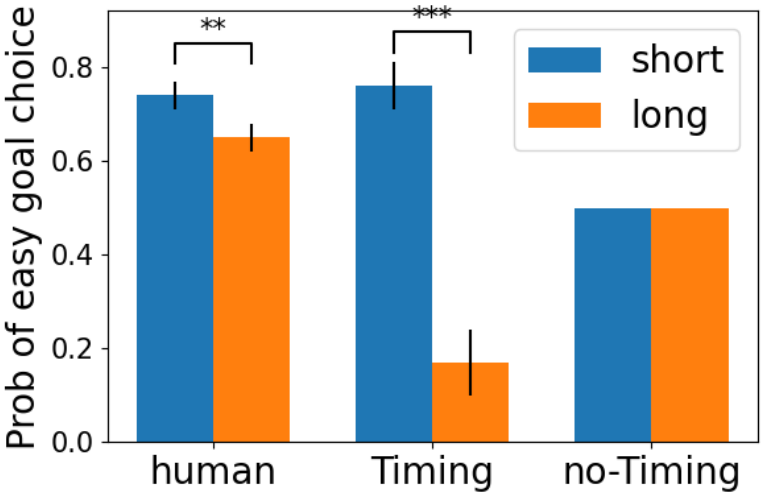


FIGURE 4.8: Human and algorithm responses on the GRT instances. The effect of thinking time is significant for humans ($p < 0.05$) and for rtPRP-a ($p < 0.005$) but there is no effect for rtPRP. Error bars show the standard deviation of the mean.

4.5.2 Goal Recognition Experiment

My work is motivated in part by the idea that humans take timing information into account when faced with goal recognition problems such as those in Figure 4.1. To my knowledge, this idea has not been previously tested, and I therefore designed a second behavioral experiment to verify that timing information can influence human goal recognition.

Experiment Configuration

I designed 13 pairs of goal recognition instances based on the Sokoban domain. One pair used the configuration in Figure 4.1. The instances in each pair included the same map and the same two potential goals. One goal (e.g. A in Figure 4.2) was easy and the other (e.g. B in Figure 4.2) was hard, where "hard" and "easy" are defined as for the previous experiment using A*.

For each member of a pair, participants saw the same sequence of three actions, and the only difference within a pair was the time observed for the third action. For "long" instances, the time associated with the third action was 3 seconds, and for "short" instances the time was only 0.5 seconds. The first two actions were always forced (e.g. in Figure 4.1 an actor who does not backtrack has no option but to move up twice), and the time for both actions was always set to 0.1 seconds.

For each instance, participants observed the sequence of three actions and then indicated whether A or B was more likely to be the goal pursued by the actor. For each pair of instances, I anticipated that participants would be more likely to choose the hard goal in the long version than the short version.

The same 50 participants who completed the problem-solving experiment also completed the goal-recognition experiment, and the goal-recognition experiment was always completed second. As a result participants were familiar with the Sokoban domain by the time they started the goal-recognition task. The presentation order of the 26 goal recognition instances was randomized within participants.

Results and Discussion

Figure 4.8 shows the average probability of choosing the easy goal across all 13 pairs of instances. As predicted, humans are more likely to choose the easy goal given a short instance than when given the corresponding long instance. A paired t-test reveals that this difference between long and short instances is statistically significant, and confirms that human goal inference is sensitive to timing information. rtPRP-a ($t(12) = 5.48, p < 0.001$) shows the same pattern as humans ($t(12) = 4.26, p = 0.001$) but the action-only algorithm rtPRP does not consider timing information and therefore generates identical responses to short and long instances.

Although humans and rtPRP-a are both sensitive to timing information, they respond differently to long instances. Humans prefer to choose easy goals even for long instances, but rtPRP-a is more likely to choose hard goals than easy goals across the set of long instances. This difference may reveal a lack of calibration between rtPRP-a and humans. For example, if the true goal were easy, spending 3 seconds on a single move would be highly anomalous according to rtPRP-a, but is apparently less anomalous according to people. Future work can attempt to better calibrate the predictions of rtPRP-a by aligning human and model planning times across responses to a large set of planning problems.

4.6 Future Directions

My framework opens up a number of additional directions for future work, and here I propose four that seem especially important. More comprehensive and broad discussion can be found in Chapter 6. First, as mentioned in my discussion of the synthetic experiment, a generative approach that makes accurate inferences based on human planning times will need to incorporate an accurate generative model of human planning times. The behavioral experiments suggest that the adaptive lookahead planner is accurate enough to support useful inferences about human planners. This planner, however, is far from a comprehensive account of human planning and future agent models can incorporate additional factors that influence human planning times. For example, future models may be able to capture the notion of action commitment by incorporating a meta-reasoning process about when to stop searching and add the current best action to the execution queue [161]. Future versions of the model can also take a bounded-rationality approach and explicitly incorporate human memory limitations [125].

A second direction is to develop agent models that allow for individual differences. As mentioned, my primary aim is to use this model as a general-purpose tool for AI systems targeting broad populations. However, the behavioral data suggested that planning strategies are highly variable across individuals: some participants seem to compute a complete path to the goal, while others seem to focus only on the next few steps. Future versions of the agent models could therefore include adjustable parameters that reflect individual differences in planning strategies, and the values for these parameters could be inferred on a per-participant basis.

Third, my current analyses assume that decision cost for a move (i.e. the total effort required to select the move) is proportional to the observed time for that move. This assumption holds if an agent is memoryless, and must carry out a fresh search on each move without using any information computed on previous moves. In reality, however, decision costs may be amortized over multiple moves, because humans and other memory-based agents may reuse information (such as search trees) computed on previous moves [61, 134]. This is also among the reasons why I stated previously that the A-LH planner fails to capture the entirety of the human problem-solving process. Future models can therefore consider ways to use observed planning times to infer the total decision cost associated with each move. One possible approach is to model the

total decision cost for a given move as a discounted accumulated sum that incorporates some fraction of the observed times recorded for previous moves.

Finally, although timing information is often informative about the goals of an actor, this relationship may not hold in contexts in which actors use strategies other than forward search to make decisions. In some scenarios, especially when people are dealing with familiar situations, they might act immediately in a reflex way without thinking or reasoning [59]. Whether or not actors carry out forward search could potentially be inferred on the basis of timing information. Future extensions of the agent model could therefore adopt a hierarchical approach that supports two inferential phases: the first phase aims to identify moves for which an actor has relied on forward search, and the second phase uses thinking time of these moves to infer the goal pursued by the actor.

4.7 Conclusion

Goal recognition is an important problem for both AI and cognitive science researchers. Most work in this area considers action sequences only, but I showed that humans are sensitive to timing information and introduced a goal recognition framework that can take timing information into account. To develop and evaluate this framework I introduced an adaptive lookahead planner with a response-time mechanism inspired by the evidence accumulation literature in cognitive science. My results suggest that incorporating an accurate model of timing is a promising way to improve the performance of goal recognition algorithms, and that the adaptive lookahead planner captures human planning closely enough to support useful inferences about the goals pursued by human actors. Because timing information is easy to acquire and generally observable, exploiting this information can potentially provide payoffs across many different settings.

4.8 Summary

In this chapter, I explored **RQ2**: "How can a human-like planning algorithm be leveraged to enhance the performance of methods for goal recognition?". Departing from conventional goal recognition approaches, I introduced a novel goal recognition framework that incorporates timing information and illustrated how the adaptive lookahead

planner can be used to formulate a timing-sensitive goal recognition algorithm within this framework.

In addition, I conducted human experiments to illustrate the practicality of timing-sensitive goal recognition algorithms when interacting with humans. Two findings emerged from the results of these experiments. Firstly, humans might act without realizing the unsolvability of a goal as the actor, and human observers might interpret an actor's true objective as an unattainable goal. Secondly, humans do exhibit sensitivity to timing information, although their behavior does not align perfectly with the predictions of the timing-sensitive goal recognition algorithm. This opens up several questions for investigation: What factors could influence human goal inference? How does solvability come into play? What mechanisms underlie human goal inference? In the upcoming chapter, I use a Bayesian goal inference framework to address these questions.

Chapter 5

Human Goal Recognition as Bayesian Inference[1]

5.1 Introduction

Once again, imagine yourself as a warehouse worker, overseeing the operations while an AI robot assistant is navigating the facility. You observe the robot approaching a secured door, and in this scenario, two potential goals are in play: gaining access to the storage room located behind the locked door or proceeding to the shelf just outside the door. If you notice the robot pausing for an extended period right outside the door, it could suggest an intent to enter the locked room, even though this goal is currently unachievable. In this situation, you have the capability to assist by using your private key to grant access. However, if the robot simply continues past the door without stopping, you might infer that its goal is to visit the shelf, a goal that is readily achievable.

As this example illustrates, in Chapter 4 I have shown that people's ability to infer the intentions of others may be influenced by factors such as timing information in addition to observed actions [149, 150]. Furthermore, individuals can sometimes infer goals that the actor cannot currently achieve. However, most existing goal recognition focus on actions alone, neglecting the broader context, and they struggle to handle situations involving unsolvable goals [37, 39, 120, 145, 146]. In this chapter, I draw on behavioral

[1]This chapter is adapted from the article accepted by AAMAS23: "Human Goal Recognition as Bayesian Inference: Investigating the Impact of Actions, Timing, and Goal Solvability."

experiments to explore how goal recognition in humans is influenced by three kinds of information: actions, timing, and goal solvability.

Goal recognition is the problem of inferring an actor's real goal given a sequence of observations and a set of possible goals. Two notable approaches that draw on Bayesian inference [162] have emerged in the literature. In 2009, Baker et al. [9] introduced the inverse planning Bayesian model, aimed at simulating human plan recognition by modeling human Theory of Mind formally as planning. Around the same time, Ramírez and Geffner [8] independently proposed a generative approach that uses planning algorithms over planning models and is known as plan recognition as planning (PRP).

Beyond actions alone, a small group of researchers in AI and cognitive science have explored how additional sources of information help to convey what others are thinking. Singh et al. [149] used gaze data to infer people's intentions and discovered that gaze can help uncover the hidden goals of players in a board game. Gates et al. [150] developed a Bayesian model that explains how people use response times as a cue to preferences in one-shot decision making situations. In Chapter 3 I generalized the underlying idea and explored how timing information can be used in situations where actors generate rich sequences of actions, not just one-shot decisions. While both Berke et al. [19] and I report that people are sensitive to timing information (in Chapter 4), there have been no comprehensive attempts to understand the extent to which timing affects human goal inferences.

Beyond actions and timing, the solvability of candidate goals provides a third relevant cue that may influence people's goal inferences. It seems plausible that people tend to assume that actors are working towards achievable goals, because actors often have accurate beliefs and actors are unlikely to waste effort working towards goals that they believe to be unachievable. To the best of our knowledge, however, there has been very little work on the impact of solvability in goal-recognition scenarios. Psychological studies of solvability judgments generally focus on tasks like unscrambling anagrams, [163, 164], and planning scenarios have received little attention. I therefore consider solvability in addition to actions and timing information, and develop an experiment that aims to understand how these three factors influence goal inference in humans.

Figure 5.1, Figure 5.2, and Figure 5.3 suggest how the three factors can be studied using goal-recognition tasks within the domain of Sokoban. In all cases the actor is required

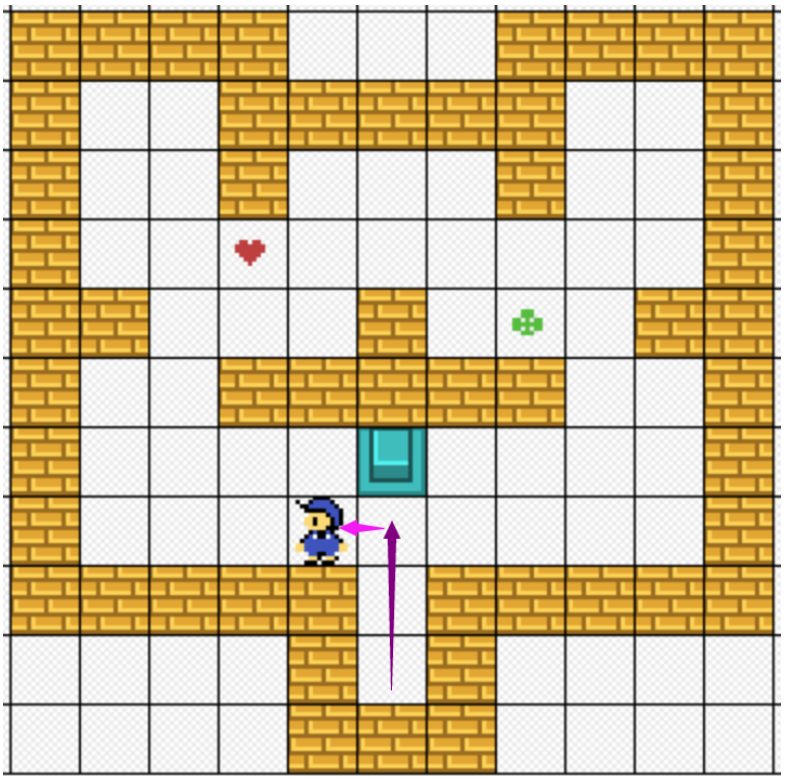
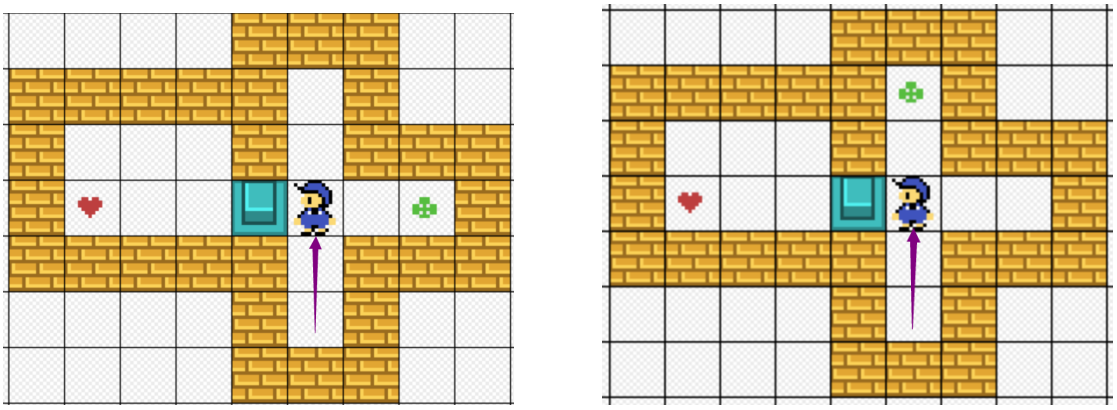


FIGURE 5.1: An *action* map. The red goal is achievable but the green goal is not, and the actor moves left at the key step.



(a) **(b)**

FIGURE 5.2: (a) An *easy-goal* map. The red goal is easy to achieve but the path to the green goal is more complex. At the key move (not shown) the actor pushes the box to the left. (b) A second *easy-goal* map. The red goal is easy to achieve but the green goal is not achievable. The key move (again not shown) involves a push to the left.

to push a box towards a goal, and the observer must infer which of two candidate goals the actor is working towards. Figure 5.1 is used to study the effect of observed actions. If the actor moves left at the key step shown as a pink arrow, people typically infer that the goal must be the green club, but had the actor moved right instead the red heart would be more probable. Figures 5.2a and 5.2b feature two identical maps with distinct

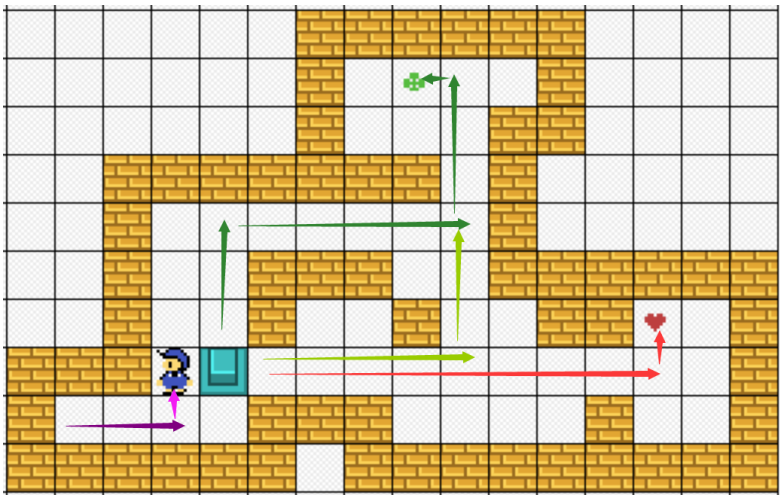


FIGURE 5.3: A *competing-path* map. There is one good path (red arrows) to the red goal and two good paths (green arrows) to the green goal. The actor moves up at the key step.

green goal positions, with Figure 5.2a representing a solvable green goal and Figure 5.2b an unsolvable one. The probability assigned to the red goal may increase when the green goal is unsolvable rather than solvable. In Figure 5.3, there is a single viable path towards the red heart but two possible paths towards the green club. If the agent thinks for a long time before taking the key step shown as a pink arrow, one possible inference is that the goal is green and the agent is deciding which of the two paths to pursue. In contrast, the red goal provides no plausible explanation of an extended pause before the key step.

The evidence available in conventional goal recognition tasks includes one or more observed actions, but I also consider scenarios using maps similar to those in previous figures but where no actions are observed. We refer to these instances as *prior* instances, because they probe expectations in advance of observing any actions. These prior instances allow us to investigate how solvability influences goal-recognition when other sources of information are absent. For instance, in Figure 5.1, in the absence of any observations, individuals may exhibit a slight preference for the solvable goal (the red heart). Previous Bayesian models of goal recognition typically assume a uniform prior [9, 37, 38, 148], but a small body of recent work has explored how actions observed in previous instances shape the priors that observers apply to new goal-recognition instances [165, 166]. Here I take a different approach, and explore how the prior reflects structural properties such as solvability and solution complexity rather than previously-observed sequences of actions.

To preview my results, I find that solvability influences people's goal-recognition judgments when no actions have been observed, but that this factor may be subsumed by a more general notion of solution complexity. When actions are observed, however, solvability appears to play a minimal role, and people's goal-recognition inferences are shaped instead by actions (as a primary factor) and timing information (as a secondary factor). I evaluate a suite of formal models and find that human goal inference is well-captured using Bayesian inference, and in particular that a Bayesian model which incorporates an online planner provides a good account of human judgments.

This chapter makes several contributions. First, I carry out a comprehensive behavioral experiment aimed at thoroughly investigating the factors that influence human goal recognition. This study provides a strong foundation for the development of computational models of human goal inference. Second, I expand upon the adaptive lookahead planner introduced in Chapter 4 by integrating a component that allows the planner to recognize unsolvable goals. Third, I introduce a human-like goal recognition algorithm that relies on Bayesian inference, and show that it provides a good account of human behavior.

5.2 A Bayesian Framework for Goal Recognition

We now formalize the problem of goal recognition and introduce a Bayesian framework for this problem. I follow the notation commonly used in the planning community [8, 38, 148], but the same general approach has been applied in the cognitive science literature [9]. Given we are incorporating timing information, the problem formulation can be found in definition 4.2.

Goal recognition can then be carried out using

$$P(G|O) \propto Prior(G)LL(O,G), \tag{5.1}$$

where $P(G|O)$ is the posterior distribution over goals, $Prior(G)$ is the prior $P(G)$ and $LL(O,G)$ the likelihood $P(O|G)$. Following the framework introduced in the last chapter, we decompose the likelihood $LL(O,G)$ into two components: the timing component $LL_T(O,G) := P(\langle t_0, t_1, ..., t_m\rangle|G)$ and the action component $LL_A(O,G) := P(\langle a_0, a_1, ..., a_m\rangle|G)$, allowing for independent calculations.

While solvability, actions, and timing might all influence human goal inference, a Bayesian perspective suggests a fundamental distinction between solvability and the other two factors. Solvability is an inherent property of the goal and should therefore be captured by $Prior(G)$ within the Bayesian model. In contrast, actions and timing are aspects of O, the observation sequence, and should be incorporated in the likelihood $LL(O, G)$.

Because previous Bayesian accounts of goal recognition usually assume a uniform prior [9, 37, 38, 148], they focus on estimating the likelihood term $LL(O, G)$. Specifically, this involves determining the probability of generating the provided observation sequence O given the goal G. Most goal-recognition models rely on standard planning algorithms that do not handle scenarios in which the goal G is unsolvable. For example, in PRP and following approaches, unsolvable goals are typically filtered out from consideration at the outset.

Some approaches avoid the assumption that the actor is rational (i.e. follow the optimal path) [9, 38, 145] like I did in Chapters 3 and 4, and can therefore estimate the likelihood of an unachievable goal. I go beyond these approaches by using a novel solvability-aware planner that can decide whether a goal is unsolvable based on the adaptive lookahead planner (see Algorithm 3).

As shown in Algorithm 3, the Solvability-aware Adaptive Lookahead commences by setting up the initial conditions, where the current state is defined and a set is prepared to track which states have been explored (line 1). It constructs a tree from the current state (line 4), signaling the start of the search process. The algorithm then enters a loop (lines 2-42), repeatedly examining the search tree to determine the next best move like adaptive lookahead planner we described in Chapter 4. The new solvability-aware mechanism is implemented by keeping track of whether any new states have been discovered, and the downtime counter is resetting if novel states are found (lines 35-41). This counter increases with each loop iteration until the algorithm concludes that no solution exists if the downtime exceeds a predetermined limit (line 42). Through this approach, the algorithm mirrors human decision-making by considering both new possibilities and existing knowledge, all while navigating through the search space efficiently.

Although I provide a limited evaluation of this planner as an account of human planning, my primary focus is on evaluating the Bayesian model of goal recognition that incorporates this planner as a component.

Algorithm 3 Solvability-aware Adaptive Lookahead

Parameters: Stopping temprature β, Exploration constant C, JoS (Judgement of Solvablity) Threshold T **Input**: Search space P, Search goal g, Current state s_0
Output: Solvability of the problem (True/False)

Let $downTime \leftarrow 0, s_c \leftarrow s_0, visitedStates \leftarrow set()$
while $downTime < T$ **do**

  Let $tree \leftarrow Tree(s_c), n \leftarrow 0$ {Construct a tree rooted on state s_c}
  Let $node \leftarrow tree.root, v' \leftarrow -\infty, v'' \leftarrow -\infty$ {v' and v'' denote the best child node value and second best child node value of the root node respectively}

  {use adaptive lookahead to find the next move and record the thinking time as n, stopping probabilitiy is defined as $P_{stop}(s_0, n) = \frac{n}{n+I(s_0)\gamma \exp(-n/I(s_0))}$}
  while *tree.root* is expanded **and** stop not triggered **do**
    while *node* is expanded **do**
      $node \leftarrow UCBSelect(node, C)$
    end while

    if *node.state* is g **then**
      $a \leftarrow softmaxSelect(tree.root), n$
      break
    end if

    $n \leftarrow n + 1$
    for *succ* **in** $P.successors(node.state)$ **do**
      $node.children.add(Node(succ, gc(succ)))$ {Initialize the new generated node using goal counting heuristic}
      $tree.update(gc(succ))$ {Backpropagate the new evidence so values of all ancestor nodes are updated by averaging the obtained values of all visits passing through the node}
    end for

    Update v', v''
    $I(s) = \frac{v'}{(1+\beta)v' - v''}$
  end while

  $a \leftarrow softmaxSelect(tree.root), n$
  $s_c \leftarrow P.execute(s_c, a)$

  if s_c is g **then**
    return True
  end if

  if There is any novel state in *tree* that not in *visitedStates* **then**
    add novel states to *visitedStates*
    $downTime \leftarrow 0$
  else
    $downTime \leftarrow downTime + n$
  end if
end while
return False

5.3 Experiment Configuration

To explore how actions, timing and solvability influence goal recognition and to test competing computational models I conducted a human experiment using the Sokoban domain. Although goal recognition is our primary focus, the experiment began with a planning phase in which participants were asked to solve 23 Sokoban problems. 9 of these problems were unsolvable, and participants could press a specified button at any stage if they believed that the current instance was unsolvable.

Participants then moved on to a goal-recognition phase using the same maps presented in the planning phase. Each instance presented a Sokoban map with two possible goal positions marked as A and B. Participants were asked to infer the actor's goal, and provided responses on a six point Likert scale labeled "very confident A", "fairly confident A", "slightly confident A," "slightly confident B," and so on. For subsequent analyses we mapped these six responses to probabilities {0, 0.2, 0.4, 0.6, 0.8, 1}, where each probability represents the probability of choosing goal B [167]. For example, if three participants chose "very confident A" and two chose "slightly confident B" the average response would be $\frac{3\times0+2\times0.6}{2+3} = \frac{1.2}{5} = 0.24$.

The stimuli for the goal recognition phase belong to one of three types, and included 20 *prior* instances, 40 *observation* instances and 9 *filler* instances. The presentation order of these instances was fully randomized. Identical map configurations and goal positions were used for the prior and observation instances, but the prior instances required participants to infer the actor's goal without having observed any actions. In all *filler* instances participants observed the player pressing the button to declare the instance unsolvable. Responses to these instances will not be analyzed, and they were included only to reinforce the possibility that the goal might be unsolvable.

The *observation* instances included pairs that share identical maps and potential goal positions but differ in a single key step (see Figures 5.1 and 5.3). This key step refers to the first step at which a player who does not backtrack has multiple options. Within each pair, either the action for this step or the response time for the action at this step can vary. There are 20 pairs in total, corresponding to the 20 instances in the *prior* type.

The observation instances can be organized into three subtypes. *Action* pairs differ based on the action taken for the key step (see Figure 5.1). I hypothesize that changing the action at this step will influence human inferences regardless of the solvability of the potential goals.

The remaining two subtypes allow me to study the influence of timing information. *Easy-goal* pairs use maps where one goal is easy to solve and the other goal is either solvable or unsolvable (Figure 5.2a and 5.2b). In this subtype, the thinking time for the key step varies. I hypothesize that increasing the thinking time at this step will decrease participant's confidence that the actor is aiming for the easy goal, because achieving the easy goal should not require a prolonged pause at any stage.

Competing-path pairs (the third and final subtype) include cases in which one goal (e.g. the green goal in Figure 5.3) requires a choice between two possible actions at the key step, but the other goal suggests only one natural action at this step. As for easy-goal pairs, I vary the thinking time observed at the key step. I hypothesize that increasing the thinking time at this step will suggest that the actor is choosing between two paths, and therefore aiming for the competing-path goal rather than the alternative.

For each map configuration, I started with a goal-recognition instance featuring two solvable goals. I then created additional instances by moving each solvable goal in turn to either an adjacent unsolvable position or an unsolvable position with similar properties (e.g. Manhattan Distance from the start position). Figure 5.2a shows an original instance with two solvable goals, and Figure 5.2b is a variant in which the green goal is unsolvable. Manipulating solvability in this way allows me to explore the influence of solvability on human goal inference.

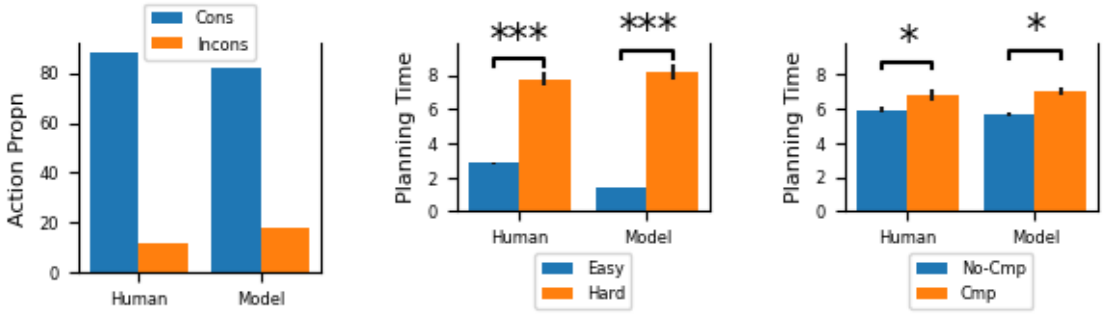
The experiment was pre-registered on AsPredicted (https://aspredicted.org/YRN_Y96) and was approved by Melbourne University Ethic Committee. I recruited 100 standard sample participants (63 females and 37 males with a median age of 28) on Prolific, and 5 were excluded because they had more than 3 abnormal responses in the problem solving phase. For each instance, responses more than 3 standard deviations away from the mean total time and total steps for that instance were considered abnormal.

5.4 Human Problem Solving Behaviour

The problem-solving phase in the experiment serves three primary purposes. Firstly, it aims to validate the effectiveness of our manipulation of observations (i.e. actions or thinking times) in the goal recognition phase. Secondly, it seeks to analyze participants' strategies when faced with an unsolvable goal. Lastly, it involves a comparative assessment of the performance between the solvability-aware adaptive lookahead planner (sA-LH) and human participants across Sokoban instances. We convert the number of iterations generated from sA-LH into seconds by normalizing it, ensuring that the total planning time for all instances is the same across humans and sA-LH.

Across the *action* maps, the majority of participants (88%) make choices that match our manipulation in the goal recognition phase, which is consistent with the model's prediction (82%) as shown in Figure 5.4a. For instance, as shown in Figure 5.1, selecting the green goal represents a choice that is consistent with our manipulation. Across *easy-goal* maps (e.g. Figure 5.4b), participants spend less time on the easy goal, with an average of 2.84 seconds compared to 7.75 seconds for the harder goal. The model's prediction shows a similar trend: 1.41 seconds for the easy goal and 8.24 seconds for the hard goal. Across *competing-path* maps, both human participants and the model show a small but statistically significant difference in planning times for the two goals. Human planning times increase from 5.94 seconds to 6.83 seconds, and the model predicts an increase from 5.68 to 7.04 seconds (see Figure 5.4c). These results indicate that the manipulations in the goal recognition experiments are well-grounded and also suggest that the sA-LH planner provides a good account of human behavior in the Sokoban domain.

We further examined the number of steps taken before participants became aware that unsolvable instances were in fact unsolvable. The results depicted in Figure 5.5 demonstrate a positive correlation between the model's predictions and human responses. The majority of participants demonstrated behavior resembling that of online planners, taking an average of 15.23 steps, indicating that, on average, participants take approximately 15 steps before recognizing the unsolvability of the goal. The model predicted a much higher average of 35.78 steps. This divergence might be attributed to participants' general lack of patience when carrying out online experiments. A minority of participants do recognize goals as unsolvable before carrying out any actions, and failing



(a) **(b)** **(c)**

FIGURE 5.4: (a) Proportion of participant choices for the action in *action* maps. *Cons* means consistent with our manipulation in the goal recognition phase. The model employs softmax action selection with a temperature parameter set to 5. (b) Average Planning time for *easy* and *hard* goals in *easy-goal* maps. The effect of thinking time is significant for both human and model ($p < 0.001$). Error bars show the standard deviation of the mean and planning time measured in seconds. (c) Average Planning time for *competing* and *no-competing* goals in *competing-path* maps. The effect of thinking time is significant for both human and model ($p < 0.05$). Error bars show the standard deviation of the mean and planning time measured in seconds.

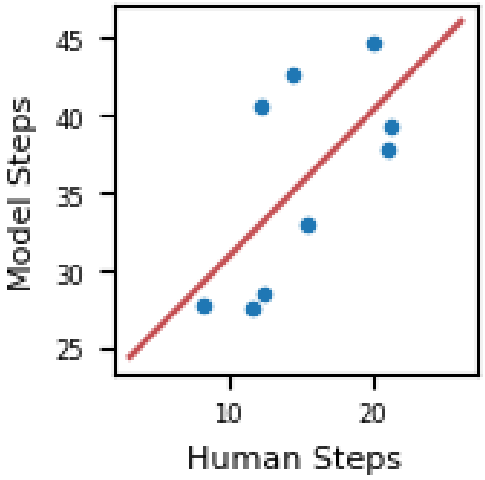


FIGURE 5.5: Number of steps taken in unsolvable instances for humans (x-axis) and the model (y-axis). Human responses and model predictions are strongly correlated ($r(7) = 0.65, p = 0.05$).

to capture the responses of these participants may also contribute to the difference between model predictions and average human responses. Nevertheless, the data strongly suggest that the majority of people should be characterized as online planners in the experimental context.

Model	Model String ($CL \sim$)	Prior	Action	Easy-goal	Competing-path
M0	$(1\|participant) + (1\|map)$	6762.8	6252.6	2591.2	5463.8
M1	$soA + soB + soA * soB + (1\|participant) + (1\|map)$	**6741.3**	6272.5	2597.1	**5439.5**
M2	$obs + (1\|participant) + (1\|map)$	N/A	**4621.3**	**2552.4**	5466.2
M3	$soA + soB + soA * soB + obs + (1\|participant) + (1\|map)$	N/A	4636.9	2558.3	5441.7

TABLE 5.1: Bayesian Information Criterion (BIC) of models in regression analysis. The best model for each set of instances (i.e. each column) is shown using bold. The dependent variable CL is the probability assigned to goal A.

5.5 Human Goal Recognition

I use mixed effects models to fit the human responses in the goal recognition phase. In these models, the variable *CL* represents the confidence level towards goal A, ranging from 0 to 1. The variables *soA* and *soB* correspond to the solvability of goals A and B, respectively, with 1 denoting solvability and -1 denoting unsolvability. In the *action* maps, goal A represents the rightmost goal, while goal B represents the leftmost goal. In *easy-goal* maps, goal A is designated as the easy goal, while goal B is identified as the hard goal. In *competing-path* maps, goal A signifies the no-competion goal, while goal B denotes the competing-paths goal. The variable *obs* indicates whether the observation (i.e. action or planning time) is consistent with goal A (1 denotes consistent, -1 denotes inconsistent) if available. The model also includes random effects for participant and map configuration. All p-values subsequently reported are based on log-likelihood ratio tests carried out for *prior* instances and each subtype of *observation* instances. The models and summary of regression results can be found in Table 5.1.

5.5.1 Prior Instances

In *prior* instances, I present a map without any observed actions to determine how solvability or other static properties would influence the human prior $Prior_{\mathcal{H}}(G)$ over the potential goals. My hypothesis is that humans will prefer solvable goals in cases where one goal is solvable and the other is unsolvable. As shown in Figure 5.6a, the overall choice percentage of solvable goals stands at 61.16% (the sum of blue bars), and the average confidence level of choosing solvable goal is 0.59. This result confirms a clear preference for goals that can be solved.

The log-likelihood ratio test of prior instances yields $\chi^2(3) = 44.185, p < 0.001$. Model M1 demonstrates a strong fit, implying that the impact of solvability is evident. Specifically, the 95% Confidence Interval (CI) for regression coefficient of soA is [-0.05, -0.02],

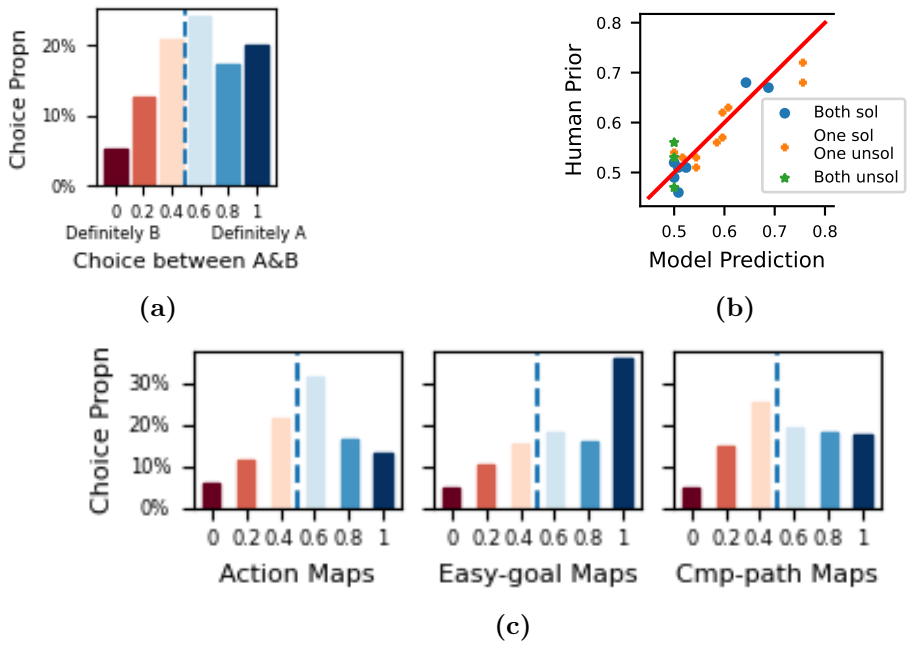


FIGURE 5.6: (a) Response distribution for prior instances where goal A is solvable and goal B is not. Blue bars indicate a preference for solvable goal A while red bars represent a preference for unsolvable goal B. (b) Comparison between human responses and the easiness model. The x-axis represents the model's predicted probability of choosing the easy goal, and the y-axis represents the human prior observed in the experiment. The instances are represented as circles, crosses or stars based on whether neither, one or both goals are unsolvable. (c) Response distribution from Figure 5.6a broken down by the three subtypes.

while the 95% CI of soB is [0.02, 0.05]. These findings confirm our hypothesis —- when one target is solvable, participants are more likely to infer that the solvable target represents the actual goal.

When I look deeper into the differences between various types of scenarios, I notice that distinct map layouts affect how much participants rely on solvability (see Figure 5.6c). Specifically, in the *action* maps, where the primary contrast between the goals is solvability, a consistent pattern emerges: participants tend to lean toward solvable goals. Most participants, however, express only a slightly confident viewpoint. This suggests that even though participants recognize the importance of solvability, the evidence supporting it might not be strong enough to firmly guide their conclusions.

In the *easy-goal* maps, the findings reveal a substantial number of participants who exhibit strong confidence in favor of the target being solvable rather than unsolvable. This finding, however, prompts the question of whether this confidence stems solely from solvability or is influenced by other characteristics within the *easy-goal* maps. As

mentioned already, within these maps the solvable goal coincides with the easier goal. In order to further explore the possible role of easiness, I compared responses to maps that were similar except that the hard goal was solvable rather than unsolvable. I found that solvability itself does not significantly impact human inference; rather, individuals consistently lean towards the easier goal, irrespective of the solvability status of the other goal.

For *competing-path* maps, solvability continues to shape human judgments, but in a different way. Among the responses, 54.74% show a preference for the solvable goal, resulting in an mean confidence level of 0.57. This is even higher than the 0.56 confidence level in the *action* maps. Interestingly, when participants choose a solvable goal, their behavior stands out from when they pick an unsolvable one. While they don't seem very sure about choosing an unsolvable goal, their confidence is more balanced when they opt for a solvable goal.

Our findings suggest that human goal inference is influenced by the difficulty of goals, from easily solvable to inherently challenging scenarios. An unsolvable goal might represent an extreme version of a difficult goal. To test this idea, I developed a simple model called the *Easiness Prior Model* to fit the human prior. In this model, I operationalize the difficulty of each solvable goal g as the sum of the optimal (i.e. shortest) path length $opt(g)$ and a smoothing parameter o (set to 5 in our analyses). This parameter captures the baseline cognitive effort demanded by the task (e.g. effort to process the map, recognize the actor and goal locations, etc). I further assume that unsolvable goals have the same difficulty score ($s = 26$) as the most difficult solvable goal in the experiments. Overall, the difficulty score for goal g is defined as $s_g = o + min(c, opt(g))$. Let s_A and s_B represent the cognitive difficulty values for goals A and B respectively in the prior instances. To reflect the notion that easier goals (with shorter optimal paths) have a higher prior, we use

$$\langle Prior(A), Prior(B)\rangle = \langle \frac{s_B}{(s_A + s_B)}, \frac{s_A}{(s_A + s_B)}\rangle. \tag{5.2}$$

As shown in Figure 5.6b, my model closely aligns with the actual prior probabilities observed in the prior instances (Pearson correlation test: $r(18) = 0.91, p < 0.001$). This

finding suggests that our simple easiness model can effectively mimic human decision-making when no observations are available: preference for solvable over unsolvable goals maybe an instance of a more general preference for easier over harder goals.

5.5.2 Observation Instances

The observation instances consists of pairs that share identical maps and potential goal configurations but differ in a single key step. This key step refers to the first action where a player who does not backtrack has multiple options. Within each pair, either the action for this step or the response time for the action can vary. Each pair also corresponds to a prior instance which shares the same map and goal configurations without including any observations.

There are three specific subtypes within the observation pairs, which also corresponds to three different types of maps in the prior instances. In what follows I consider the three subtypes separately.

Action Pairs

The result confirm my hypothesis: solvability rarely contributes to the final decision in goal choice when actions are informative. Regardless of whether the goal is solvable or unsolvable, the shift in goal preference, compared to the prior (that slightly favors the solvable goal), aligns with the guidance provided by action observations. When the action supports to the unsolvable goal, the confidence level for the solvable goal shifts from 0.56 to 0.24, and when the action supports to the solvable goal, the confidence level for that goal increases to 0.81. I also ran a log-likelihood ratio test to verify the hypothesis (see Table 5.1).

Among the models considered, Model M2 demonstrates the best fit ($(\chi^2(1) = 1638.7, p < 0.001)$), as evidenced by its lowest Bayesian Information Criterion (BIC) value. The 95% CI for the regression coefficient of *obs* falls within the range of [-0.32, -0.3]. Conversely, neither *soA* nor *soB* contributes meaningful information to the confidence level in this context. Notably, Model M1 even exhibits a higher BIC value than the baseline model (M0), indicating that solvability fails to enhance the model fit.

Easy-goal Pairs

Compared to the prior condition, regardless of the time actors take to think about the key steps, human responses shift towards the easy goal in the presence of observations. This shift is evident as the confidence level for the easy goal changes from 0.69 to 0.86 given short thinking time. However, when a long thinking time (consistent with hard goals in our hypothesis) is observed, this shift is somewhat less pronounced (0.69 to 0.75). Additionally, I observed that this pattern remains consistent, irrespective of whether the hard goal is solvable or not.

I performed an identical log-likelihood ratio test using easiness to establish *obs*: short thinking time is consistent with easy goal A (assigned 1) and long thinking time is aligned with hard goal B (assigned -1). The results aligned with my initial intuition: Model M2 exhibits the most favorable fit ($\chi^2(1) = 45.425, p < 0.001$). This underscores the notion that thinking time is useful for inferring the confidence level, while solvability's contribution remains negligible. The 95% confidence interval for the regression coefficient of the intercept spans from 0.26 to 0.34, indicating a strong tendency among participants to favor the easier target choice. Furthermore, the 95% confidence interval for the regression coefficient of *obs* (i.e. long/short thinking time) falls within the range of [0.04, 0.07]. This outcome emphasizes that the manipulation of thinking time can exert a notable influence on the confidence level, contributing to statistically significant variations in participants' goal inference processes.

Competing-path Pairs

Broadly speaking, the patterns observed within the *competing-path* pairs align closely with those of the *easy-goal* pairs. In particular, when participants observe the actions, their preferences shift towards the no-competition goals whether they spend more or less time. Unlike the *easy-goal* instances, the initial distribution of *competing-path* maps is nearly uniform (with confidence level to the no-competition goal of 0.5) as shown in Figure 5.6c. With consistent observations (i.e. short thinking time) favoring the no-competition goal, the confidence level to that goal increases to 0.58. Surprisingly, even with inconsistent observations (i.e. long thinking time), the confidence level still increases to 0.56. This result implies that our definition of consistency (long/short

thinking time) may not be the primary factor observers take into account during goal inference.

I applied the same log-likelihood test using number of competing path as the standard to establish *obs*: short thinking time is consistent with no-competition goal A (assigned 1) and long thinking time is aligned with competing-path goal B (assigned -1). All three models yield significantly better fits than the baseline model, with Model M1 – which considers only solvability – achieving the optimal fit ($\chi^2(2) = 41.442, p < 0.001$) based on BIC. In the comprehensive Model M3, the 95% confidence interval for the regression coefficient of *soB* lies between [-0.07, -0.04], while the intervals for *soA* and *obs* encompass [0.00, 0.04] and [0.00, 0.03] respectively. These results indicate that in this context, the solvability of competing goal B presents a substantial impact on human inferences, while the solvability of the no-competing goal A and the influence of thinking time are comparatively more modest. Increased awareness of solvability of competing goals suggests individuals may allocate more time to plan for these goals, aligning with the assumption in sA-LH and my results in previous two chapters.

5.5.3 Model Comparison

I now evaluate a range of models by comparing them against human goal inference behavior. These models are formulated within the same Bayesian framework (Equation 5.1) but use 3 different priors $Prior(\cdot)$ (uniform prior, easiness prior model shown in Equation 5.2, empirical prior based on our problem solving data) and 5 different likelihoods $LL(\cdot,\cdot)$ (offline-planning likelihood, online-planning likelihood, online-planning likelihood with actions only, empirical likelihood, empirical likelihood with actions only).

For offline-planning likelihood estimation, I adopt the PRP approach outlined by Ramírez and Geffner [37]. This approach is not designed to handle unsolvable goals, but as originally formulated it consistently prioritizes solvable goals ahead of unsolvable goals. All easy-goal and competing-path maps were intentionally designed so that actions would be uninformative about the goal, and in these cases the offline likelihood assigns equal weight to both targets.

The online likelihood is derived from 100 simulations conducted using the solvability-aware adaptive lookahead planner (sA-LH). To minimize variance in our model predictions, I focus solely on the likelihood associated with the key step, as the other two steps are predetermined. Specifically, I need to calculate the action component LL_A and the timing component LL_T separately for each goal and then combine them. For the action component, the likelihood is estimated by dividing the number of action choices made in the simulations by the total number of simulations (i.e. 100). As illustrated by Figure 5.4a, I previously confirmed that sA-LH aligns with human action choices in *action* instances. In the remaining two types of instances, we found that actions still provide valuable information for goal inference. In *action* instances, since the simulated times for both targets are the same, the timing likelihood LL_T effectively makes no contribution. In *easy-goal* and *competing-path* instances, the timing likelihood is computed under the assumption that $LL(\cdot, g)$ follows a Gaussian distribution with a mean determined by the average number of sampled iterations needed to achieve goal g. I further assume that long and short thinking times in the goal-recognition experiment correspond respectively to the average number of iterations generated by sA-LH for hard and easy goals.

The empirical likelihood draws inspiration from the inverse planning approach introduced by Baker et al. [9]. I estimate the empirical action and timing likelihoods in the same way as the sA-LH likelihoods except that the samples are based on human responses collected during the problem-solving phase instead of simulations from sA-LH. For example, the mean and standard deviation for the Gaussian timing likelihood are based on the human responses provided during the problem-solving phase.

For both the online and empirical likelihoods, I consider variants that incorporate only the action component LL_A. These variants are useful for establishing whether timing information is needed to account for our human goal-recognition data. For all log-likelihood calculations, we add a small value of 0.025 to both options to prevent the occurrence of zero probabilities.

Results and Discussion

As shown in Fig 5.7, the Easiness prior with online likelihood (actions only) achieves the best overall performance as measured by the log-likelihood assigned to the entire data set. Comparing the rows of Fig 5.7 suggests that the contribution of the prior

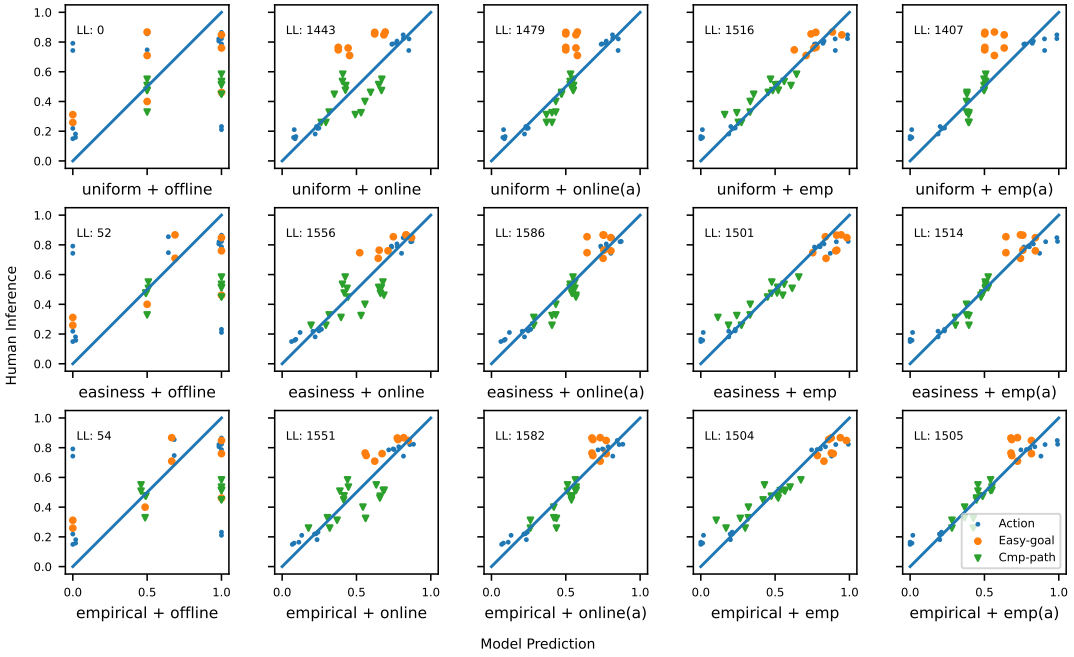


FIGURE 5.7: Comparison between model predictions and human inferences. All model labels show the prior followed by the likelihood: for example, *uniform + emp* is the model with uniform Prior and the empirical likelihood. *emp(a)* and *online(a)* are likelihoods that incorporate actions but not timing information. For readability, log likelihoods (higher is better) are shown as offsets relative to the log likelihood of the *uniform+offline* model.

is important but small. In contrast, comparing the columns reveals that changing the likelihood can have a dramatic effect on model performance.

It is striking that the online likelihoods seems comparable or superior to the empirical likelihoods even though the empirical likelihoods were directly fit to human behavioral data. The online likelihoods are based on the A-LH planner, and the strong performance of these likelihoods suggests that the A-LH planner provides a robust and reliable account of human behavior. In contrast, the offline likelihood performs substantially worse than the online and empirical likelihoods, suggesting that the participants implicitly assumed that the actor in the goal-recognition task relied on an online planning strategy.

Comparing results for likelihoods with and without timing information suggests that timing information is not needed to account for the behavioral data, and that incorporating this information may slightly impair model performance. Although the online likelihoods with and without timing information yield similar levels of performance, the two show distinct patterns across the three map types. Of the two online likelihoods, the action-only version performs worse across easy-goal maps, but better across the other

two map types. This finding suggests that timing information may be beneficial in specific scenarios even though it provided no overall boost in performance across our entire data set.

Although varying the prior does not affect model performance as much as varying the likelihood, it is notable that the Easiness prior model and the empirical prior achieve similar levels of performance. This finding provides additional support for our previous finding (see Fig 5.6b) that the Easiness model is well-aligned with human judgments.

5.6 Related Work

Ramírez and Geffner [37], along with subsequent researchers such as Vered et al. [38] and Masters and Sardina [120], introduced the Plan Recognition as Planning (PRP) approach that uses planning to estimate the likelihood. I evaluated this approach (referred to as the offline likelihood) as a baseline. This approach assumes agent rationality and focuses exclusively on actions, leaving unaddressed the explicit treatment of unsolvable goals.

Berke et al. [19] have explored the influence of timing information on human understanding of others. Their study, however, is not anchored in the domain of goal recognition, and they rely on a domain-specific algorithm for likelihood estimation.

Baker et al. [9] introduced a Bayesian framework for human goal inference and conducted a systematic human experiment demonstrating their model's ability to achieve human-like inference, but did not consider the influence of timing and solvability. They acknowledged the possibility of a non-uniform prior in humans, but did not explore this idea experimentally.

Some recent research has considered non-uniform priors in goal recognition [165, 166]. These approaches, however, focus mainly on incorporating past information into the prior within the context of sequential Bayesian updating. We depart from this approach by investigating how domain-independent factors (i.e solvability and easiness) influence human priors.

5.7 Conclusion

In this study I used a Bayesian framework to systematically investigate the influence of actions, timing, and goal solvability on goal recognition. Through an in-depth analysis of human responses in the Sokoban domain, I found that while actions are typically attributed the highest importance, timing and goal solvability also influence goal recognition, particularly in scenarios where actions offer limited information. Leveraging these insights, I developed a goal recognition model that closely aligns with human inferences, surpassing the performance of existing algorithms.

My work explored role of the prior, which is often overlooked in the literature. My results suggest that humans rely on a prior that incorporates factors such as solvability and perceived goal difficulty. I formulated a model of the prior (the Easiness model) that proved successful in accounting for human responses, both before and after any actions had been observed. However, our model comparison results also indicate that, for the purpose of modeling human inference, uniform priors may remain a viable and pragmatic choice. But this observation should not be extended to numerous real-world scenarios. Consider, for instance, a personalized assistant that relies on a uniform prior; in such cases, the assistant's functionality would be severely limited and ineffective. A final consideration regarding priors is how humans might formulate them. Some readers may argue that prior should be immediately available regardless of the reasoning process and should not require inference of task difficulty. However, in this thesis, prior specifically denotes human inference without directly observing the actor's behavior. This includes information such as goal solvability or other structural parameters unaffected by the actor's behavior. This interpretation of prior diverges from the psychological definition, which usually denotes beliefs formed without a reasoning process.

I extended the Adaptive Lookahead Planner to capture human behavior in the presence of unsolvable goals, and our model comparison suggests that this extended model is useful for estimating the likelihood term required by the Bayesian goal-recognition framework. This planner, however, departs from human behavior in some respects (e.g. by taking more steps before recognizing a goal as unsolvable), and future work should aim to improve it further.

The evaluation of the influence of actions, timing, and solvability suggested that actions (when available) have a dominant influence on people's choices. This finding provides some justification for the standard emphasis on actions within the goal-recognition literature. Nevertheless, my observations also revealed the influence of solvability and timing, particularly in situations where actions are uninformative. My results seem broadly compatible with an information-seeking approach [168] to goal-recognition in which humans focus initially on actions but turn to other factors such as timing and solvability if actions prove uninformative. Future work can explore this information-seeking approach in more detail and compare it with the traditional Bayesian approach.

Finally, I conducted a thorough examination of Bayesian inference and the commonly used mirroring approach (i.e. planning for likelihood estimation) discussed in previous work [9, 37, 38]. The empirical model, which relies on problem-solving data, exhibits a strong alignment with human goal inference. This finding suggests that humans may indeed rely on Bayesian inference and mirroring to carry out goal-recognition. I also introduced a goal recognition model (the model that combines the easiness prior with the online likelihood) that can be implemented independently of human problem-solving data while generating human-like goal inferences. I expect that this model may prove to be useful in a range of downstream applications, including explainable goal recognition [169] and transparent planning, a process focused on selecting actions that effectively convey the actor's intentions to observers [14]. Researchers in these domains may be able to leverage this model to advance the development of more interpretable AI behavior.

5.8 Summary

In this chapter, I considered **RQ3**: How do humans carry out goal inference, and can this inference be captured within a Bayesian framework? My investigation reveals that the Bayesian framework exhibits the capability to capture human inference, with factors such as action, thinking time, and goal solvability contributing to the inference in various scenarios.

So far, we have already covered various scenarios in human-agent interaction in the context of goal recognition. However, many questions remain open and worthy of future exploration. Key areas for consideration include the development of more sophisticated

human-like models, the potential integration of learning approaches into our framework, and the practical application of our research findings in real-world contexts.

In the next chapter, I discuss these questions and propose potential avenues for future research.

Chapter 6

Future Directions

In Chapters 3, 4, and 5, we explored planning algorithms with a focus on mirroring human behavior in action selection and planning time. This investigation serves as a foundational step towards a comprehensive understanding of human problem solving and goal recognition strategies through the lens of automated planning techniques. Given the foundational nature of this research, there exists a breadth of possibilities for deeper exploration.

At the conclusion of each chapter, we briefly introduced potential future research directions. In this section, we will engage in a thorough discussion of these paths, aiming to outline a more detailed and structured research agenda for each area.

6.1 Problem Solving

In my exploration of problem solving, I have developed adaptive lookahead planners that successfully emulate human behavior in terms of choosing actions and timing them, specifically within puzzle configurations such as the TOL and Sokoban domains. The model presented, though efficient, is part of an ongoing effort to develop computational counterparts that closely emulate human problem-solving processes. Its fidelity could be significantly enhanced by evaluating and potentially revising its components, such as the policies for selecting actions and backpropogate values [170].

For example, there are two potential methods for value backpropagation. The first method is max backup, which involves selecting the maximum value from the child nodes

and propagating it back to the parent node. This approach can be particularly effective in scenarios where maximizing performance is the primary objective. Another method is average backup, where the average value of all child nodes is calculated and then used for the parent node's value. This technique might be more suitable in environments where a balanced approach is needed, taking into account various possible outcomes.

Indeed, it would be interesting to investigate which of these two backpropagation methods more accurately mirrors human problem-solving mechanisms. Determining this could be achieved through experimental approaches such as direct measurement, for instance, using self-reporting techniques where participants detail their decision making process. This could provide insight into whether they tend to use maximal outcomes (aligning with max backup) or consider a balance of possibilities (in line with average backup).

Furthermore, identifying the specific scenarios where each method is most applicable could be beneficial. Different problem solving contexts may call for distinct approaches; for instance, max backup might be more suitable in highly competitive or goal-oriented scenarios, while average backup could be better in situations that require risk assessment or dealing with uncertainty.

Beyond a more thorough consideration of component choices, the model also overlooks certain key characteristics inherent in human cognitive processes. First of all, it functions statically, without capturing the progression of learning that naturally occurs with repeated problem solving attempts. Integrating these dynamic elements—specifically, the capacity for individuals to evolve and optimize their approach to problem solving—is a substantial challenge [64].

Moreover, the planner operates on the assumption that each planning session begins from scratch, disregarding any previous planning efforts. In contrast, human decision-making benefits from working memory mechanisms (i.e. previous solutions and strategies can inform and streamline current problem solving endeavors). Integrating this aspect of cognitive recall is crucial, as real-world problem solving often builds upon prior knowledge and planning [171].

Additionally, while my work's goal has been to construct a computational model that

captures human average behavior, the variability in individual problem solving approaches cannot be overstated. Tailoring the model to accommodate individual differences is vital for its application in real-world scenarios [64, 117].

In the following parts, I will illustrate potential methodologies for overcoming these challenges within the framework of adaptive lookahead. This will include strategies for modeling learning effects, working memory mechanisms, and individual behavioral difference, thereby enhancing the model's applicability and precision in simulating human-like problem solving in more realistic settings.

6.1.1 Learning Effects

Learning effects refer to improvements in performance or efficiency that occur as a result of repeated practice or experience with a particular task or activity. Essentially, as individuals become more familiar with a task, they often become better at it. This improvement can be observed in various ways, such as faster completion times, higher accuracy rates, or the ability to perform the task with less cognitive effort. Within my behavioral experiments, learning effects are consistently observed, reflecting their widespread presence across various cognitive tasks.

In the context of cognitive psychology and education, learning effects are associated with how practice or study leads to long-term changes in mental processes and knowledge retention. In artificial intelligence and computational modeling, learning effects can be simulated by algorithms that adjust and optimize their performance as they process more data or encounter similar tasks over time.

Approaches that are centered around learning, such as machine learning algorithms, inherently exhibit learning effects, adapting and refining their performance as they gain more data or experience [69]. In contrast, model-based methods typically rely on static models that do not change over time. Despite this, there are promising strategies that can be integrated into model-based frameworks to mimic learning effects [15].

For instance, with increased experience, individuals can develop richer and more efficient ways of representing a problem. Taking TOL as an example, players may begin to recognize patterns in the puzzle over time. Instead of seeing move one followed by move two, they see a singular operation leading to a desired outcome, effectively compressing

several steps into one mental leap. This cognitive compression, or chunking, mirrors the technique where larger tasks are broken down into meaningful segments or hierarchies [80, 99, 172], each with its own goal leading up to the final objective. This can be modeled by merging states or actions into higher-order chunks within the AI system, reducing the cognitive load and simplifying the decision process. This process is explored as learning macro action and HTNS and also learning heurstics in planning[173–175].

Another potential improvement in representation with experience involves selective attention to relevant information while disregarding what is irrelevant [176]. As individuals become more adept at the TOL task, they learn to focus on the aspects that are crucial for solving the puzzle and ignore distractions. For example, an experienced TOL player might start ignoring the color of the pegs while a beginner might consider all elements of the puzzle. This refined approach is akin to a filter, where the player's cognitive resources are conserved by attending only to the components that directly affect the outcome, such as the positions that need to be filled and the shortest sequence of moves to achieve that end, effectively streamlining the problem solving process [176, 177].

Beyond the development of environment models, the proficiency of a player might also be reflected in the enhancement of heuristic functions [178]. Heuristic functions become more precise with practice, offering improved guidance through the maze of possibilities. As players gain insight into the pattern of the game, these guiding heuristics are refined to more adeptly drive towards successful outcomes. For example, with experience, a TOL player may develop a better understanding of the puzzle, enabling them to more accurately judge the goodness or potential of a particular peg configuration. They might intuitively recognize that certain arrangements of disks are steps toward an optimal solution, while others are further from the goal state.

While we have discussed several model-based explanations for the learning effect, the challenge of systematically incorporating these insights into an automated planning framework persists. Determining whether such an integration can accurately capture human learning effects remains an unresolved question. As the section concludes, we acknowledge this area as a frontier for future research, offering both a challenge and an opportunity to deepen our understanding of learning and problem solving within the crossfield between cognitive science and AI.

6.1.2 Working Memory Mechanism

Learning effects can be viewed as a facet of long-term memory, which is responsible for storing information over lengthy durations. This includes declarative memory, encompassing facts and explicit knowledge, as well as procedural memory, which relates to the methods and processes involved in tasks like puzzle-solving. In addition to these, another critical memory mechanism integral to problem-solving is working memory [179]. This type of memory is engaged in the active manipulation of information necessary for task completion. It temporarily holds data that can be actively used and altered, such as tracking the sequence of moves in a puzzle or integrating search results from previous steps to inform current decisions. For example, in our behavioral experiments, participants exerted considerable effort in planning their first move upon encountering the puzzle. This finding suggests that the cognitive process involved in problem solving does not reset at each step. Instead, participants appear to build upon the results of their previous planning. This indicates a form of cognitive efficiency where prior work is retained and recycled.

Subgoaling is one of the significant concepts in cognitive psychology and AI that discuss the utilization of working memory during problem solving activities [33, 179, 180]. Subgoaling involves breaking down a larger task into smaller, more manageable objectives, which can make complex problems easier to navigate. This method of problem decomposition allows individuals to focus on achieving interim goals, providing a sense of progress and direction that can guide them toward the ultimate solution. Subgoaling, which likely arises from learning effects and the influence of long-term memory—as exemplified by the Tower of London (TOL) scenario discussed previously—is critically underpinned by working memory. This cognitive function is key in maintaining interim objectives and their associated strategies in a readily accessible state, facilitating ongoing cognitive processing and strategy execution. Related research in these areas examines how individuals utilize working memory to dynamically manage and adjust their problem-solving strategies. Studies have shown that individuals with greater working memory capacity are often better at tasks that require subgoaling, as they can juggle more information and potential strategies at once [180].

Like learning effects, understanding how automated planning algorithms can explain subgoaling and working memory mechanisms is still an open question. In the next

part, I introduce an initial attempt to combine model-based search with these cognitive strategies.

Conversion between observed time and total planning effort

In Chapter 4, for simplicity I made the assumption that the planning effort is identical to the time recorded in the observation sequence. This assumption may not strictly hold because people or agents might rely on working memory to reuse information computed in previous planning phases, which means the total planning effort and observed planning time might be different. Here I describe some preliminary research that aims to fill in this gap.

We define total planning effort as the cognitive resources an individual dedicates to make a decision. This effort represents an internal cognitive activity that is not directly measurable from the outside. Initially, this total planning effort is equal to the observed planning time when an individual is first presented with a task. However, it is highly likely that in subsequent states, the total planning effort exceeds the observed planning time. This is because individuals carry forward some of the cognitive effort invested in earlier stages, which is not captured in the time observed for later decisions. Thus, I propose a simple conversion to estimate the total planning effort from observed planning time. To be specific, we can use a discounted accumulated sum to calculate the total planning effort $t(s_i)$ for each state s_i from timing observation sequence $t_i, t_{i-1}, ..., t_1$ as $t(s_i) = t_i + \lambda t_{i-1} + \lambda^2 t_{i-2} + ... + \lambda^{i-1} t_1$, where λ is a constant factor representing the discount for previous observed planning time. For example, if we observe the agent spend 100 time units on the first step, 200 time units on the second step, 50 units on the third step, then the real planning time for the third step is $t = 50 + 200 * 0.8 + 100 * 0.8^2 = 274$ if we choose the discount rate $\lambda = 0.8$. The intuition behind this approach comes from both computer science and cognitive science: from the perspective of online planning algorithms, this mechanism can be considered as reusing the previous subtree [181] and many researchers in cognitive science argue that humans use a similar process to solve complex problems [61, 134]. Under this framework, we can also model the memoryless agent by setting the discount factor to 0.

To test the validity of this conversion method, I looked at the similarity (i.e. cosine distance) between the timing sequence generated by the planner without memory mechanism used in my experiment and the planner with memory mechanism. The memoryless planner generates the the real planning effort (target sequence) while the planner with memory mechanism generates observed planning times that need to be converted into real planning effort. We considered instances where both agents have the same sequence of actions, where only BlocksWorld and EasyIpcGrid fall under this category. In BlocksWorld, this conversion (λ is set to 0.8 empirically) brings up the similarity from 0.75 to 0.96 on 30 instances. In EasyIpcGrid, the similarity increases to 0.98 from 0.59 by the same conversion on 34 instances.

The preliminary findings presented here suggest that our conversion method is a promising approach for estimating the total planning effort from observed planning times. However, further research is needed to refine this model and explore its broader applications. Future studies could investigate the impact of varying the discount factor λ on the accuracy of the conversion, potentially tailoring it to individual differences in cognitive processing speeds and strategies. Additionally, exploring how this model performs across a wider range of tasks and environments would help validate its generalizability and effectiveness.

Another avenue of research could examine the neurological underpinnings of the discounting mechanism, seeking correlations between our model and neural activity observed during problem-solving tasks [182]. This could bridge the gap between computational modeling and cognitive neuroscience, offering deeper insights into how humans naturally integrate past planning efforts into current problem solving activities.

In summary, this subsection lays the groundwork for a novel method of understanding and simulating planning effort in both humans and artificial agents. The potential for this work to contribute to the fields of cognitive science, and artificial intelligence is substantial, and this exploration marks just an initial step in realizing the comprehensive potential of this approach.

6.1.3 Individual Differences

Learning effects demonstrate how an individual's problem-solving strategies evolve over time, even when facing the same task repeatedly. This contrasts with the concept of individual differences, which acknowledges that different people naturally possess distinct approaches to problem solving from the outset [183, 184]. In Chapter 3, I highlighted these individual differences through the lens of the Tower of London task. Furthermore, in Chapter 5, we observed a diverse array of problem-solving behaviors, particularly when participants were confronted with goals that could not be solved. In this section, I consider these inherent individual variations in more detail.

While the primary focus of my thesis does not center on crafting behavior models that predict individual actions with high precision, the significance of such models cannot be overstated in practical applications. The field of cognitive science considers the understanding and modeling of individual differences to be a pivotal issue. Research, such as that conducted by Callaway et al. [64], introduces models that incorporate adjustable parameters tailored to each individual, aligning the model's performance with personal behavior. This approach is increasingly prevalent in computational cognitive science, reflecting a shift toward personalized modeling that captures the unique cognitive profiles of different individuals.

As discussed at the end of Chapters 3 and 4, the adaptive lookahead planner that I proposed is very flexible, capable of reflecting individual differences by integrating customizable components. For instance, individuals may vary in their decision thresholds or in the parameters governing their stopping probabilities during problem solving. Moreover, they may employ distinct problem representations or exhibit preferences for certain states over others. These personal attributes can be effectively captured by the adaptable nature of the lookahead planner, allowing for a more personalized and accurate modeling of individual problem solving strategies.

Although there are various methods to investigate individual differences in the context of problem solving, I propose two particularly promising avenues for exploration. The first approach involves the use of clustering methods. Clustering algorithms are unsupervised learning techniques that organize objects into groups, or clusters, based on their similarities. These techniques have been effectively employed across numerous

disciplines—including data mining, pattern recognition, image analysis, information retrieval, and bioinformatics—to unearth patterns and structures within datasets [185]. Recently, applications of clustering algorithms in cognitive research have provided valuable insights into human individual differences [186–189]. Within the problem solving context, clustering can serve as a powerful pre-processing step to categorize participants into distinct groups based on their approach to tasks. After clustering, it is feasible to apply different types of algorithms, such as online versus offline planning algorithms, to better match and predict the behavior of each group. This approach can yield a more nuanced understanding of the strategic variations individuals employ when faced with complex problems.

A second promising avenue to understanding individual differences in problem solving is to examine a range of cognitive factors, including working memory capacity, attentional control, and others. By employing standardized cognitive tasks designed to assess these abilities [190, 191], we can investigate their direct correlation with parameters in planning algorithms. For instance, an individual's working memory capacity may correspond to the constraints they impose on the number of nodes they consider and the depth to which they search in a given task. By applying standard working memory tests, such as the Operation Span or Reading Span [191], we can explore the potential positive correlation between an individual's working memory capacity and the number of expanded nodes in the model that best fits their behavior. It would be interesting to explore whether specific cognitive capacities can be predictive of, or even dictate, certain algorithmic parameters that individuals naturally employ during problem-solving.

6.2 Goal Recognition

In Chapter 4, I detailed algorithms that factor in human cognitive processes for the purpose of goal recognition. Subsequently, in Chapter 5, I explored the design of algorithms that aim to replicate human approaches to identifying goals. This section will discuss potential directions for future research and refinement within these two domains.

6.2.1 Beyond Simple Lab Tasks

The tasks employed in my research, such as the Tower of London, Sokoban, and others found in goal recognition benchmarks, are laboratory-based and represent a considerable simplification of real-world scenarios. Therefore, a natural progression for future research is to investigate the applicability of the algorithms I have developed in real-world contexts. To this end, I propose a specific real-world problem that could serve as a valuable focus for subsequent research.

Consider the application of the timing-sensitive goal recognition algorithm in a smart home assistant system. This system aims to comprehend the habits and behavior patterns of users to minimize manual intervention in appliance usage. To effectively fulfill this objective, the smart home system must be capable of inferring residents' intentions in real-time and responding with suitable assistance, such as adjusting household appliance settings accordingly. Goal and intent recognition have been increasingly implemented in smart homes by some researchers, as seen in the works of Rafferty et al. [192] and [171]. Wilken and Stuckenschmidt [193] introduced classical planning method (i.e. PRP) into the smart home environment and showed it can achieve better performance than purely statistical approach.

The timing-sensitive goal recognition algorithm can contribute to the inference part thus improve the overall usefulness of the system. The same approach used by [171] can be used to gather sensor data, to convert them into atomic actions and to define a library of goals that might be pursued by the resident. Currently, the sensor system is capable of collecting precise timing information [171]. This data is directly accessible for further analysis. Then we describe problems using PDDL by modelling them as a state space model as presented in Wilken and Stuckenschmidt [193]. This methodology allows the generation of numerous goal recognition tasks that incorporate timing information. These tasks are derived from the sensor data and can be processed using the timing-sensitive goal recognition algorithms discussed in Chapters 4 and 5. Eventually, the system can potentially use inferred goal to offer appropriate help to users.

To see how timing information can enhance the performance of a smart home, consider a system involving a sensor-equipped refrigerator. When a user pauses for an extended period before the refrigerator, say over a minute, this duration, captured by the sensor, typically indicates an intention to prepare something to eat. This inference is based

on the assumption that the user is evaluating contents, contemplating meal options, or organizing ingredients. The smart home system, leveraging this timing information, can offer contextual assistance such as suggesting recipes based on available ingredients, activating kitchen lights, preheating the oven, or playing background music. Conversely, a brief pause, lasting only a few seconds, is interpreted as a simple action, like reaching for a drink. The system can respond by highlighting hydration options or subtly illuminating the water dispenser, which may be especially useful during nighttime. These scenarios exemplify how timing information, by providing deeper insights into user behavior, can transform a smart home system into a more responsive and intuitive entity, adept at catering to the immediate and specific needs of its users. Later, we will discuss a potential framework for integrating knowledge about specific users using the same example.

Before we proceed to the next subsection, it's important to note another limitation of the approach in this thesis, which hinders its application in real-world contexts. We assume explicit reasoning in problem-solving, and we expect the observer to adopt this assumption and conduct simulations to infer the intentions of the actors (i.e., simulation theory). However, as mentioned in the previous section, individuals may engage in reflective reasoning in many real-world scenarios, particularly when tasks are familiar or straightforward. In such cases, if the observer still assumes that the actor is employing explicit reasoning or uses this information to infer planning time, their conclusions may not be accurate (Berke, 2023). Furthermore, it is possible that individuals employ theory theory instead of simulation theory as observers, suggesting that likelihood estimation may not solely result from simulation, as posited in Chapter 5. These aspects warrant further investigation.

6.2.2 Goal Recognition for Human Actor

While the timing-sensitive goal recognition algorithm I proposed demonstrates superior performance over existing algorithms in various scenarios, substantial work remains to enhance its applicability in real-world contexts. In this part, I will outline two potential directions for further development: incorporating methods based on machine learning and devising strategies for personalized prior knowledge acquisition.

Combining Learning-based Method with Goal Recognition

Automated planning methods grounded in model-based approaches have shown promising results in controlled, synthetic benchmarks. However, their performance often diminishes in more complex and unpredictable real-world environments where complete information about the environment or the actors within it is not readily available. Researchers such as Pereira et al. [39], Ramírez and Geffner [37], and Vered et al. [38] have noted these shortcomings. To bridge this gap, there have been efforts to employ reinforcement learning and deep learning frameworks, yet these approaches frequently encounter difficulties in achieving effective generalization across diverse scenarios [3, 112, 194, 195]. Despite these challenges, there are instances where the application of learning methods to planning algorithms has yielded success [111, 196]. Nonetheless, the fusion of learning and planning in the realm of goal recognition remains a significant hurdle. Developing a robust mechanism that effectively integrates these paradigms is crucial for advancing the field and addressing the current limitations faced by goal recognition models in dynamic environments.

In this work, I present a novel learning challenge: the prediction of thinking time. This project marks the first attempt to incorporate timing information into goal recognition, and the process of generating and evaluating timing likelihood emerges as a fresh area of study.

Specifically, in this project, likelihood estimation is conducted via simulations using adaptive lookahead planners. The strength of this model-based method lies in its independence from training datasets, coupled with the ability to integrate theoretical constructs from cognitive science, like drift-diffusion models, directly into the algorithm.

Nonetheless, there remains considerable room for improvement in refining the model to more accurately reflect human response times. Many different factors influence thinking time during problem-solving, and the forward search algorithm employed here relies on a set of assumptions that may not accurately reflect the full breadth of human cognitive activity, particularly in naturalistic contexts. For instance, one assumption is that individuals are entirely focused on the task at hand and perform explicit reasoning, which diverges from common real-world scenarios where multitasking is prevalent. In many scenarios, people might rely on reflective planning instead of explicit reasoning, when the task is familiar or easy. Furthermore,

These essential simplifications, which make the simulation process more manageable, may not always align with the behaviors we see in human subjects. Consequently, the application of machine learning or deep learning techniques to predict thinking time is a promising direction. These methods have the capacity to consider a multitude of factors and can learn to predict outcomes without a precise characterization of the underlying mechanisms.

One potential way to approach thinking time in sequential decision-making relies on time series approach [197]. Here is a research plan to study it through learning techniques:

- **Data Collection:** Collect a dataset of thinking times from individuals engaged in sequential decision-making tasks, ensuring data is sequential and time-stamped.
- **Feature Engineering:** Extract features from the data that may influence thinking time, including both constant features (those that remain unchanged throughout the problem-solving process, like goal configuration and map size) and local features (attributes related to each intermediate state, such as task difficulty and attention level).
- **Time Series Analysis:** Apply time series analysis to understand patterns in thinking times, and select a forecasting model, such as ARIMA [198], LSTM networks [199], or RNNs [200]. These models are particularly adept at handling the temporal dynamics present in goal recognition tasks. For instance, LSTM can effectively capture long-term dependencies in sequences of decisions, which is crucial in understanding how earlier actions and thinking times in a task may influence later thinking times.
- **Model Training and Prediction:** Train the model on historical thinking time data, allowing it to learn and make future predictions.

Using these steps, a learning approach can help uncover the dynamics of thinking time in sequential decision-making, potentially leading to the development of predictive models that could anticipate an individual's decision making pace and improve the design of systems that are sensitive to human timing requirements.

Customized Knowledge

In Chapter 5, we explore a novel dimension often overlooked in previous literature, which is that priors are not always uniform. This is achieved by introducing a domain-independent prior function, which is based on the task's difficulty level. We show the validity of our proposed prior by empirically measuring human priors and incorporating them into the human-like goal inference algorithm. It is crucial to understand that this prior assumes an observer's beliefs about an actor's goal preferences (i.e. prefer to easy goals). However, in practical settings, different actors may exhibit diverse goal preferences, and accurately reflecting these variations is vital for the success of real-world applications. Furthermore, my approach employs a model-based method to estimate likelihood through an adaptive lookahead planner. This approach implies that we assume the actor plans and thinks in a manner similar to the planner, which is a significant assumption. In this section, we explore how to relax these assumptions using a practical example, underscoring the importance of tailoring our approach to individual actors' unique preferences for more effective goal recognition.

Let's consider an example in the context of a smart home assistant again. The assistant observes the residents' behaviors to recognize their goals and adjust the household appliances usage accordingly. Suppose the assistant notices that a resident is turning off the light in the room. A uniform prior might suggest that the goal is always to save energy. However, individual preferences could indicate different goals. For instance, Resident A is environmentally conscious and has their energy-saving intent correctly identified by the assistant. For Resident B, however, who perhaps has a preference for dim lighting and only turns off lights to create a relaxed atmosphere, an overarching assumption by the assistant could misinterpret this action, leading to unintended adjustments like powering down other appliances, which do not align with Resident B's actual intent. This example demonstrates that recognizing unique goal preferences is crucial for the goal recognition system to make accurate inferences and take appropriate actions.

In the smart home scenario, the critical role of behavior patterns (likelihood) in goal recognition should also be acknowledged, separate from individual preferences (prior). Consider a sophisticated thermostat system monitoring two residents, each exhibiting unique behavior patterns: Resident A typically raises the thermostat before exercising, suggesting a desire for a warmer environment during physical activity, and lowers it while

watching TV, indicating a preference for a cooler setting during leisure. On the other hand, Resident B exhibits the opposite behavior, decreasing the thermostat temperature before exercise and increasing it during TV time. These specific actions serve as key indicators, guiding the system to infer the immediate goals of each resident accurately. A standard system might erroneously apply the same model for interpreting these actions to both residents, potentially resulting in inappropriate temperature adjustments. In contrast, by understanding the unique behavior patterns of each individual, the smart thermostat can more accurately predict and respond to their current needs.

To address the challenge of learning individual preferences and behavior pattern in a smart home context, as highlighted in the example, the following research plan can be implemented:

- **Data Collection:** Implement a system for continuous monitoring of resident interactions with the smart home environment. This includes tracking usage patterns of appliances, light usage, and other relevant activities.
- **Behavioral Analysis:** Conduct a thorough analysis of the collected data to uncover patterns indicative of personal preferences or goals. This analysis could be performed using a knowledge-based approach, where human input labels the data, or through an unsupervised learning approach to autonomously discover significant patterns.
- **Preference Profiling:** Based on the analysis, construct detailed profiles that includes both preference (i.e. prior) and behavior pattern (i.e. likelihood estmation) for each individual. In a learning-based approach, these profiles might be represented as vectors. Conversely, in a knowledge-based approach, such as planning, users could be categorized according to different planning strategies or varying hyperparameters in the algorithm.
- **Predictive Modeling:** Develop a predictive model that can anticipate an individual's goals based on their current actions. This model will utilize the created profiles to tailor its predictions to each resident.
- **Continuous Learning:** Establish a feedback loop that allows residents to provide input on the system's interpretations. This feedback is crucial for continually

refining and updating the model, particularly in enhancing the accuracy of the prior distribution and likelihood estimation within the goal recognition algorithm [111].

By following this methodology, the smart home assistant could dynamically learn and adapt to the unique and potentially changing preferences of each resident, leading to more accurate goal recognition and a more personalized user experience.

6.2.3 Human Goal Recognition Mechanism

In Chapter 5, my focus was on demonstrating the application of Bayesian inference in understanding patterns in human goal recognition behavior. However, this framework does not completely capture all observed phenomena. For example, in situations where actions do not provide explicit information, I observed that factors like solvability and timing become more significant. This aligns with the information-seeking model of goal recognition [168], suggesting that humans actively seek out additional information for decision making. This contrasts with the traditional passive Bayesian inference approach, which relies on the availability of all pertinent information. Such observations indicate a need to consider dynamic information-gathering processes in our understanding of human goal recognition, particularly in scenarios involving multiple sources of information.

Information-seeking conceptualizes goal recognition as an active decision making process, where individuals are not mere passive observers but active participants who seek out data to inform their understanding of others' objectives. This paradigm shifts the focus from passive inference to a more dynamic interaction with the environment, where gathering additional information is a strategic component of recognizing goals.

To construct models of human goal recognition based on information-seeking, one could follow these steps [181]:

- **Identifying Relevant Factors for Goal Recognition**: Begin by determining the key factors that influence goal recognition. These might include observable actions, the solvability of a task, and the time spent in decision-making. Understanding which elements are most influential will guide the subsequent stages of model development.

- **Information Priority Assessment**: Conduct behavioral experiments, potentially involving the manipulation of participants' attention and eye tracking technique might be employed to gather data on where and how attention is focused [201]. This step aims to assess the priority of various pieces of information in the context of goal inference, helping to determine the relative weight or importance of each factor in the observer's decision-making process.

- **Computational Modelling**: Develop a sophisticated computational model that simulates the observer's decision-making process in seeking information. This model should intricately factor in the current level of uncertainty, as well as the cost and practicality of acquiring new data. It should also determine which types of information to pursue, taking into account their priority and accessibility. The model must be fine-tuned and validated to align with actual human inference collected in the behavior experiments [168].

Exploring how humans adaptively integrate an actor's preferences and behavior patterns into goal inference is an intriguing avenue for future research. This aspect could be rigorously examined through cognitive experiments by manipulating the preference and behavior patterns of the actor observed. As highlighted by the research of Jara-Ettinger et al. [202], even young children have the capacity to develop varied mental models for others, implying a fundamental human ability in this domain. Studying the specific contexts of goal recognition in which individuals rely on limited information to construct mental models of others is valuable, as it not only enriches our understanding of human behavior but also has potential implications for the advancement of algorithms as elaborated in Section 6.2.2.

Chapter 7

Conclusion

The thesis introduces new methodologies and concepts in problem solving and goal recognition, integrating insights from both automated planning and cognitive science. Central to this work is the development of planners that mimic human cognitive processes, drawing inspiration from cognitive science theories about evidence accumulation. These human-like planners adeptly simulate human action selection and response times across a range of scenarios. Equipped with these planners, we also introduce a new framework of goal recognition that is able to use information beyond action. The usefulness of our approach has been validated in both problem-solving and goal recognition tasks. Additionally, the research advances a Bayesian framework that combines a prior based on goal difficulty with a likelihood derived from an online planner. This combined approach has been shown to accurately forecast human goal inference, highlighting the nuanced nature of human decision-making and the potential of these models to capture such complexities.

The work detailed in this thesis predominantly adopts a human-centered approach, meaning that the underlying concepts and theories are constructed based on the extensive literature in cognitive and social sciences. In contrast to most previous studies within the computer science literature, such an approach is particularly effective in enhancing performance within the context of human-agent interactions, which is supported by the empirical results presented in this thesis. This chapter summarizes the research contributions, insights and findings derived from addressing the main research questions

outlined in Chapter 1 (as shown in Table 7.1), and also discusses the limitations associated with each contribution. All datasets, algorithms and software packages developed for the projects in the thesis can be found at https://github.com/chen-yuan-zhang.

7.1 Human-like Planning Algorithm

To address **RQ1** (see 7.1), I examine how well various existing planning algorithms from AI perform in the Tower of London (TOL) task, with a focus on replicating human actions and response times.

The development of computational models of human problem solving is grounded in the long history of problem-solving research in cognitive science [22, 203]. This background, starting from the early days of AI with the Logic Theorist, is crucial for understanding the evolution and potential of these algorithms in simulating human decision-making in sequential tasks. However, these cognitive models often face challenges when integrated into modern AI systems because they are usually domain-specific and not easy to implement by modern programming language. Therefore, it becomes important to explore whether algorithms from the field of AI could be used to mimic human behavior more effectively in these systems.

The decision to use planning algorithms to mimic human behavior is grounded in several key factors. First, these algorithms are designed to be a general problem solver than other learning-based algorithms in computer science, mirroring the versatility and adaptability of human problem-solving skills. Unlike methods requiring extensive domain specific information, planning algorithms can operate with a broad range of general knowledge, making them more flexible and applicable in varied scenarios. This characteristic is particularly important as it aligns closely with how humans often approach problems: not with exhaustive, specific information, but with adaptable strategies and a general understanding of the task. Lastly, the application of these algorithms in modern AI systems presents an opportunity to create more human-like, intuitive AI solutions. Integrating planning algorithms that replicate human decision-making processes into AI systems can bring these systems closer to natural human thought patterns and problem-solving approaches. This alignment has the potential to improve human-AI collaboration.

Research Question	Objectives	Contribution
RQ1: Which algorithm is most suitable for emulating human responses (both action selection and response times) in sequential decision making tasks?	1. Compare existing model-based algorithms with human responses, focusing on action selection and response times within the context of sequential decision making tasks. 2. Develop a human-like planning algorithm	1. An Adaptive Lookahead Planner that generates human-like actions and planning times
RQ2: How can a human-like planning algorithm be leveraged to enhance the performance of methods for goal recognition ?	1. Develop a framework to integrate auxiliary information into the task of goal recognition. 2. Create a timing-sensitive goal recognition algorithm that uses a human-like planning algorithm for estimating the likelihood. 3. Enhance the overall performance of goal recognition system by incorporating timing information and human-like planning algorithm.	1. A new goal recognition benchmark that extends existing goal recognition benchmarks by incorporating timing information derived from the human-like planning algorithm. 2. A framework for goal recognition that integrates timing information and a novel timing-sensitive goal recognition algorithm.
RQ3: How do humans carry out goal inference, and can this inference be captured within a Bayesian framework?	1. Conduct an in-depth analysis to identify and understand factors that affect human goal recognition. 2. Develop a computational model, grounded in the Bayesian framework, that effectively simulates the human goal inference process.	1. A comprehensive behavioral experiment to thoroughly investigate the factors that influence human goal recognition. 2. A novel planner to identify and respond to unsolvable goals. 3. A human-like goal recognition algorithm that uses Bayesian inference to mimic human goal inference.

TABLE 7.1: Research contributions

In addressing RQ1, the TOL task was transformed into the Planning Domain Definition Language (PDDL). This transformation is key for applying planning algorithms to a cognitive task, providing a method for comparing algorithmic planning with human problem-solving strategies. A behavioral experiment was conducted to gather data for comparing human decision making with predictions from planning algorithms. The experiment, with instructing participants either to find the best path or to find any path, was designed to collect empirical evidence crucial for assessing how closely these algorithms emulate human actions and response times.

In addition to existing planning algorithms, I proposed the adaptive lookahead planner (A-LH), a novel approach designed to mimic the human problem solving process. The uniqueness of this planner lies in its adaptive planning horizon, which optimizes based on the complexity of the task, drawing inspiration from the concepts of evidence integration and human meta-reasoning. The A-LH uses the upper confidence bound (UCB) algorithm for action selection and continues to search until there is a significant difference in the value between the best and the second best actions, as determined by a goal-counting heuristic. This adaptive mechanism allows the A-LH planner to flexibly adjust its search depth in response to different scenarios.

The research found that the adaptive lookahead planner was more effective than multiple competing approaches at predicting human behavior in structured scenarios, suggesting it as a suitable algorithm for emulating human responses in sequential decision making tasks. However, this work is not without its limitations. A notable constraint lies in the current model's ability to fully encapsulate the breadth and diversity of human problem solving strategies, especially in less structured decision making scenarios, such as emergency response, where decision must be made quickly based on incomplete or rapidly changing information. These scenarios often present a wide range of variables and unpredictable elements that human cognition can navigate with remarkable flexibility. The current implementation, while effective in structured tasks like the Tower of London, may not yet fully mirror this aspect of human cognition.

7.2 Timing-sensitive Goal Recognition Algorithm

Goal recognition is the task of identifying an actor's objectives based on their actions and contextual information within a given environment. Current goal recognition algorithms primarily focus on the actor's actions, assuming that these actions are rational. However, additional auxiliary information can be useful, and that actors' behavior may not always be optimal. To address these aspects, I incorporate a human-like planning algorithm (i.e. the adaptive lookahead planner). This integration addresses two aspects of **RQ2** (see 7.1): First, it accounts better for non-optimal behavior than do standard methods. Second, it considers the time spent on planning, which provides insights into the actor's internal cognitive processes.

The core of this work is the introduction of a novel goal recognition framework that incorporates timing information. This framework departs from the traditional approach in AI goal recognition, which predominantly focuses on the actions observed [37, 38, 120, 166]. Recognizing that timing information is a crucial aspect often used in human inference about others' goals, the framework aims to exploit this dimension.

Before introducing the timing-sensitive goal recognition algorithm, I first appended timing data, generated by the adaptive lookahead planner, to standard goal recognition benchmarks [39]. This timing data, produced by the adaptive lookahead planner, simulates human-like thinking times. To effectively use this timing information, I developed the timing-sensitive goal recognition algorithm. The algorithm operates under the assumption that both actions and planning times depend solely on the current state and the true goal. It uses a Markovian approach where planning time and action are considered conditionally independent. The algorithm decomposes the likelihood of an observation sequence into two components: the action component and the timing component. This decomposition allows the algorithm to separately evaluate the likelihood of actions and their associated planning times, which are then combined to produce a comprehensive goal recognition solution.

Subsequently, a synthetic experiment is conducted to evaluate the effectiveness of my approach in addressing RQ2. I use standard goal recognition benchmarks with timing information to compare the performance of the timing-sensitive goal recognition algorithm against existing goal recognition algorithms. The results shows the successful

exploitation of timing information in goal recognition, especially when observed actions are not informative.

The major question left open by our synthetic experiment is whether timing information can still be exploited when the process generating planning times is not fully known. I therefore developed two behavioral experiments to explore whether the adaptive lookahead planner matches humans closely enough to allow the timing-sensitive goal recognition algorithm to exploit timing information when inferring the goals of humans.

In the first experiment, 50 participants completed 24 Sokoban puzzle instances each. The goal was to understand if the adaptive lookahead planner could generate human-like planning times and if timing information could be used for inferring human goals. The results showed a standard goal recognition algorithm achieved an average fractional rank of 2.5 (similar to random choice), while timing-sensitive goal recognitin algorithm performed better with an average rank of 1.75. This finding indicates that the adaptive lookahead planner is not only effective in generating human-like planning times but can also serve as a useful component in goal recognition algorithm.

The second experiment was designed to test if humans consider timing information in goal recognition tasks. The experiment involved 13 pairs of goal recognition instances from the Sokoban domain. Each pair of instances had the same map and two potential goals (one easy and one hard), but varied in the timing observed for the third action (3 seconds for "long" instances and 0.5 seconds for "short" ones). Participants were asked to choose between the two goals after observing a sequence of three actions. The results showed a statistically significant tendency for participants to choose the hard goal more often in the long version than in the short version of each pair. This result suggests a potential alignment between human goal inference processes and the timing-sensitive goal recognition algorithm, thereby providing a key insight in response to RQ3 discussed in the following section.

7.3 Computational Models for Human Goal Recognition

In addressing RQ2, we developed a timing-sensitive goal recognition algorithm designed to infer the goals of human actors. This work, however, did not address the question of how humans themselves perform goal inference. This aspect becomes particularly

critical in scenarios where a human observer is interacting with an AI system. In such cases, it is essential for the AI system to ensure transparency in its behavior [14]. By combining the adaptive lookahead planner with the principles of Bayesian inference, we address **RQ3** (see 7.1) by developing a model of human goal recognition.

In the experiment, participants were presented with pairs of Sokoban puzzle instances, each pair sharing the same map but differing in key aspects, such as action, solvability and thinking time. The aim was to discern how these factors influence goal recognition. The experiments were categorized into action pairs, easy-goal pairs, and competing-path pairs, each designed to test different hypotheses about goal inference. The results revealed that while actions appear to be a primary factor in goal recognition other elements such as the easiness of the goal and the presence of competing paths also significantly influence decision-making. This was particularly evident in situations where the actions were less informative.

The experiment provides a solid foundation for developing computational models that can accurately simulate human goal inference processes. By thoroughly examining how humans infer goals, we gain invaluable insights into what kind of factors should be integrated into computational models of goal inference. In the existing literature, the Bayesian framework is a prevalent tool in goal recognition research across both cognitive science and AI disciplines [37, 117]. However, a common limitation in these studies is the oversight of the role of prior, coupled with likelihood estimations are either based on empirical data [117] or do not adequately account for human factors [37]. In this work, I address these weaknesses by considering the non-uniform prior and enhancing the likelihood estimation to more accurately reflect human goal inference.

Moving away from the uniform prior, my study indicates that human priors depend on factors like solvability and perceived goal difficulty. We developed a prior model based on the goal difficulty, effectively predicting human responses in scenarios with or without observed actions. This easiness model is notable not only for its effectiveness but also for its domain independence, making it a versatile tool that can be applied in many contexts.

For the estimation of likelihood, we considered not only the offline likelihood, which is based on classical planning algorithms commonly used in the planning literature [37, 38], but also evaluated empirical and online likelihood approaches. The empirical likelihood

was derived from the problem-solving data collected from the same group of participants, aligning the model's predictions closely with human inferences and thereby supporting the mirroring approach [38]. Additionally, the online likelihood, sourced from the adaptive lookahead planner, allowed the model to achieve performance on par with the empirical likelihood, effectively matching human behavior as well. This results indicates that it is possible to achieve human-like goal inferences without relying on empirical problem-solving data.

Overall, my work on RQ3 showed a strong correspondence between Bayesian inference models and human goal inference patterns. In addition, I introduced a novel Bayesian goal recognition model that combines the easiness prior with the online likelihood and closely replicates human goal inferences without relying on human problem-solving data and . This model holds potential for applications in explainable goal recognition and transparent planning, offering a path for researchers to develop more interpretable AI systems.

7.4 Final Remarks

This thesis sits at the crossroads of cognitive science and automated planning, showcasing the potential of model-based approaches in modern AI algorithms to describe human behavior. The contributions made in this work take some steps toward advancing our understanding of human-like intelligence. While learning-based approaches have achieved significant success across many domains, the symbolic nature of human reasoning remains a crucial element of human intelligence. Our research underscores the importance of integrating insights from cognitive science into AI systems to move closer to the realization of human-like intelligence.

Appendix A

PDDL files for the Tower of London Task

PDDL files are grounded so that they can be run by LAPKT planners.

A.1 Domain file

```
(define (domain TOL)
  (:requirements :equality)
  (:types

  )

  (:constants

  )

  (:predicates
      (free-loc_0_0 )
      (free-loc_1_0 )
      (free-loc_1_1 )
      (free-loc_2_0 )
```

```
    (free-loc_2_1 )
    (free-loc_2_2 )
    (clear-b_0 )
    (clear-b_1 )
    (clear-b_2 )
    (at-b_0-loc_0_0 )
    (at-b_1-loc_0_0 )
    (at-b_2-loc_0_0 )
    (at-b_0-loc_1_0 )
    (at-b_1-loc_1_0 )
    (at-b_2-loc_1_0 )
    (at-b_0-loc_1_1 )
    (at-b_1-loc_1_1 )
    (at-b_2-loc_1_1 )
    (at-b_0-loc_2_0 )
    (at-b_1-loc_2_0 )
    (at-b_2-loc_2_0 )
    (at-b_0-loc_2_1 )
    (at-b_1-loc_2_1 )
    (at-b_2-loc_2_1 )
    (at-b_0-loc_2_2 )
    (at-b_1-loc_2_2 )
    (at-b_2-loc_2_2 )
)

(:functions

)




(:action move_0
 :parameters ()
```

```
 :precondition (and (and (and (and (and (at-b_0-loc_1_1 ) (at-b_1-loc_2_0 ))
 (at-b_2-loc_1_0 )) (free-loc_2_1 )) (clear-b_0 )) (clear-b_1 ))
 :effect (and
    (not (at-b_0-loc_1_1 ))
    (not (clear-b_1 ))
    (not (free-loc_2_1 ))
    (at-b_0-loc_2_1 )
    (clear-b_2 )
    (free-loc_1_1 ))
)


(:action move_1
 :parameters ()
 :precondition (and (and (and (and (and (at-b_0-loc_1_1 ) (at-b_1-loc_2_1 ))
 (at-b_2-loc_1_0 )) (free-loc_2_2 )) (clear-b_0 )) (clear-b_1 ))
 :effect (and
    (not (at-b_0-loc_1_1 ))
    (not (clear-b_1 ))
    (not (free-loc_2_2 ))
    (at-b_0-loc_2_2 )
    (clear-b_2 )
    (free-loc_1_1 ))
)


(:action move_2
 :parameters ()
 :precondition (and (and (and (and (and (at-b_0-loc_2_1 ) (at-b_1-loc_1_0 ))
 (at-b_2-loc_2_0 )) (free-loc_1_1 )) (clear-b_0 )) (clear-b_1 ))
 :effect (and
    (not (at-b_0-loc_2_1 ))
    (not (clear-b_1 ))
    (not (free-loc_1_1 ))
```

```
        (at-b_0-loc_1_1 )
        (clear-b_2 )
        (free-loc_2_1 ))
    )


    ...

    (:action move_113
     :parameters ()
     :precondition (and (and (at-b_2-loc_2_0 ) (free-loc_1_0 )) (clear-b_2 ))
     :effect (and
        (not (at-b_2-loc_2_0 ))
        (not (free-loc_1_0 ))
        (at-b_2-loc_1_0 )
        (free-loc_2_0 ))
    )

)
```

A.2 Problem file

```
    (define (problem TOLproblem1)
    (:domain TOL)

    (:objects

    )

    (:init
        (free-loc_0_0 )
        (free-loc_1_0 )
        (free-loc_1_1 )
```

```
        (at-b_0-loc_2_0 )
        (at-b_1-loc_2_1 )
        (at-b_2-loc_2_2 )
        (clear-b_2 )
    )

    (:goal
        (and (at-b_1-loc_2_2 ) (and (at-b_2-loc_2_1 ) (at-b_0-loc_2_0 )))
    )




)
```

Appendix B

PDDL files for the Sokoban Task

PDDL files are grounded so that they can be run by LAPKT planners.

B.1 Domain file

```
(define (domain Sokoban)
  (:requirements :equality)
  (:types
      object
  )

  (:constants

  )

  (:predicates
      (at-loc_0_8 )
      (box-loc_0_8 )
      (clear-loc_0_8 )
      (at-loc_0_9 )
      (box-loc_0_9 )
      (clear-loc_0_9 )
```

```
    ...

    (at-loc_7_10 )
    (box-loc_7_10 )
    (clear-loc_7_10 )
)

(:functions

)




(:action move-loc_0_8-loc_1_8
 :parameters ()
 :precondition (and (clear-loc_1_8 ) (at-loc_0_8 ))
 :effect (and
    (not (at-loc_0_8 ))
    (not (clear-loc_1_8 ))
    (at-loc_1_8 )
    (clear-loc_0_8 ))
)



(:action move-loc_1_8-loc_0_8
 :parameters ()
 :precondition (and (clear-loc_0_8 ) (at-loc_1_8 ))
 :effect (and
    (not (at-loc_1_8 ))
    (not (clear-loc_0_8 ))
    (at-loc_0_8 )
    (clear-loc_1_8 ))
)
```

```
...


(:action move-loc_7_10-loc_7_9
 :parameters ()
 :precondition (and (clear-loc_7_9 ) (at-loc_7_10 ))
 :effect (and
    (not (at-loc_7_10 ))
    (not (clear-loc_7_9 ))
    (at-loc_7_9 )
    (clear-loc_7_10 ))
)


(:action push-loc_0_10-loc_1_10
 :parameters ()
 :precondition (and (and (clear-loc_2_10 ) (at-loc_0_10 )) (box-loc_1_10 ))
 :effect (and
    (not (at-loc_0_10 ))
    (not (box-loc_1_10 ))
    (not (clear-loc_2_10 ))
    (at-loc_1_10 )
    (box-loc_2_10 )
    (clear-loc_0_10 ))
)


(:action push-loc_2_10-loc_1_10
 :parameters ()
 :precondition (and (and (clear-loc_0_10 ) (at-loc_2_10 )) (box-loc_1_10 ))
 :effect (and
    (not (at-loc_2_10 ))
```

```
        (not (box-loc_1_10 ))
        (not (clear-loc_0_10 ))
        (at-loc_1_10 )
        (box-loc_0_10 )
        (clear-loc_2_10 ))
    )


   ...

    (:action push-loc_7_10-loc_7_9
     :parameters ()
     :precondition (and (and (clear-loc_7_8 ) (at-loc_7_10 )) (box-loc_7_9 ))
     :effect (and
        (not (at-loc_7_10 ))
        (not (box-loc_7_9 ))
        (not (clear-loc_7_8 ))
        (at-loc_7_9 )
        (box-loc_7_8 )
        (clear-loc_7_10 ))
    )

)
```

B.2 Problem file

```
(define (problem instance_30001)
    (:domain Sokoban)

    (:objects

    )
```

```
    (:init
        (box-loc_3_5 )
        (at-loc_6_5 )
        (clear-loc_0_8 )
        (clear-loc_0_9 )
        ...
        (clear-loc_7_10 )
    )

    (:goal
        (box-loc_1_1 )
    )

)
```

Appendix C

Pre-registration Documents

C.1 Pre-registration document for the Tower of London experiment (Chapter 3)

1. *Have any data been collected for this study already?*

   No, no data have been collected for this study yet.

2. *What's the main question being asked or hypothesis being tested in this study?*

   Our primary goal is to evaluate a set of computational models and ask which provides the best account of human initial planning time.Our secondary goal is to test whether initial planning times (and best computational model) will be different when we explicitly ask participants to form a complete plan.

3. *Describe the key dependent variable(s) specifying how they will be measured.*

   Initial planning time. (The response time before the first move.)

4. *How many and which conditions will participants be assigned to?*

   Participants will be randomly assigned to two conditions. In the "classical planning condition" participants are asked to form a complete plan to the target configuration before making their first move. In the "default" condition participants are just asked to solve the task without any further instruction.

5. Specify exactly which analyses you will conduct to examine the main question/hypothesis.

- We will compute Pearson correlations between model "thinking times" (which correspond to the number of nodes expanded by a model) and human initial planning times (measured in seconds).
- We will run a regression analysis similar to that described by Berg to explore the relationship between initial planning time and the structural parameters considered by them (minimum moves, start hierarchy, goal hierarchy, number of paths, number of move choices)
- To test the hypothesis that response times in the classical planning condition will be longer than in the default condition, we'll use mixed models with random effects for participant and instance:

$$M1 : response_{time}\ condition + (1|participant) + (1|instance)$$

and

$$M0 : response_{time}\ (1|participant) + (1|instance)$$

We'll use a likelihood ratio test to compare M1 with M0.

6. *Describe exactly how outliers will be defined and handled, and your precise rule(s) for excluding observations.*

   We will exclude observations with abnormal response times. For each instance, responses more than 3 standard deviations away from the mean initial planning time for that instance will be considered abnormal.

7. *How many observations will be collected or what will determine sample size? No need to justify decision, but be precise about exactly how the number will be determined.*

   For each condition, we will have 120 participants (thus 240 overall). Each participant will solve 39 randomly chosen instances, and there are 117 instances in total. Each instance will therefore be completed by around 40 participants per condition.

8. *Anything else you would like to pre-register? (e.g., secondary analyses, variables collected for exploratory purposes, unusual analyses planned?)*

   Nothing else to pre-register.

C.2 Pre-registration document for the Sokoban experiment (Chapter 5)

1. *Have any data been collected for this study already?*

   No, no data have been collected for this study yet.

2. *What's the main question being asked or hypothesis being tested in this study?*

   Our goal is to test how solvability and observation affect human goal recognition, and to evaluate computational models based on how well they capture human inference. To this end, we have developed three qualitative hypotheses: first, that people rely on a prior that favors solvable goals and use solvability as a cue before any actions have been observed; second, observed action can dominate people's goal inference even when some of the goals are unsolvable ; and third, observed thinking time affects human goal inference, but to a weaker extent than observed action.

3. *Describe the key dependent variable(s) specifying how they will be measured.*

   The key dependent variable in this study is a participants' response indicating their preference between two potential goals. These responses will be provided on a 6-point Likert scale, and we'll treat this scale as an interval scale.

4. *How many and which conditions will participants be assigned to?*

   All participants will respond to the same instances, which will be presented in a random order. Prior to encountering these instances, participants will engage in a problem-solving phase involving Sokoban puzzles. The data from this problem-solving phase are mostly relevant to a different project that focuses on evaluating models of human planning, and analyses for this other project have been outlined in a separate preregistration document.

   For this project, we will use data from the problem-solving phase to see how closely our goal-recognition manipulations align with actual human problem solving behavior. In addition, we will use these data to derive a "rationality" parameter for each participant, and this parameter may be used when analyzing goal recognition data.

5. Specify exactly which analyses you will conduct to examine the main question/hypothesis.

   The stimuli for the goal recognition phase belong to one of three types. The first type is a map without any observed actions. The hypothesis is that humans will prefer solvable goals in these cases. To investigate the influence of solvability on human responses, specifically measured by confidence level, we will conduct a log-likelihood ratio test using the following regression models:

   M0: CL (1—participant) + (1—map)

   M1: CL (1—participant) + (1—map) + soA + soB + soA * soB

   In these models, CL represents the confidence level, which ranges from -2.5 to 2.5. The variables soA and soB indicate the solvability of goal A and goal B, respectively (1 denotes solvable, -1 denotes unsolvable).

   The second type consists of pairs that share identical maps and potential goal configurations but differ in a single key step. This key step refers to the first action where a player who does not backtrack has multiple options. Within each pair, either the action for this step or the response time for the action can vary.

   To assess the influence of solvability and observation on human responses, we will conduct log-likelihood ratio tests using the following regression models:

   $$M0 : CL\ (1|participant) + (1|map)$$

   and

   $$M1 : CL\ (1|participant) + (1|map) + soA + soB + soA * soB$$

   and

   $$M2 : CL\ (1|participant) + (1|map) + obs$$

   and

   $$M3 : CL\ (1|participant) + (1|map) + soA + soB + soA * soB + obs$$

   Where obs indicates whether the observation (i.e. action or planning time) is consistent with goal A (1 denotes consistent, -1 denotes inconsistent)

The third type consists of filler instances in which participants observe the player pressing the button to declare the instance unsolvable. We may run exploratory analyses of responses to these filler items, but have not preregistered any hypotheses about the fillers.

After analyzing the human data independently, we will assess a set of models by contrasting them with human inferences. The models comprise a set of Bayesian models with distinct priors (uniform prior v.s. Solvability weighted prior) and likelihood computations (sample based likelihood v.s. offline likelihood, strong sampling vs weak sampling). This evaluation will involve performing a regression analysis between the posterior of each model and the likert scales collected during the experiment.

We will also explore the influence of individual differences on goal recognition. Specifically, participants will be categorized as rational or non-rational based on their performance in the problem-solving task. We will then examine whether different goal recognition models align with these categories.

6. *Describe exactly how outliers will be defined and handled, and your precise rule(s) for excluding observations.*

   We will exclude responses that deviate more than 3 standard deviations from the mean time, considering them abnormal during the analysis. Furthermore, if a participant has more than 3 abnormal responses, we will exclude all of their responses from the analysis.

7. *How many observations will be collected or what will determine sample size? No need to justify decision, but be precise about exactly how the number will be determined.*

   We will recruit 100 participants with the aim of retaining at least 90 per instance after exclusions.

8. *Anything else you would like to pre-register? (e.g., secondary analyses, variables collected for exploratory purposes, unusual analyses planned?)*

   Nothing else to pre-register.

Bibliography

[1] Thomas B Sheridan. Human–robot interaction: status and challenges. *Human factors*, 58(4):525–532, 2016.

[2] Manuela Veloso, Joydeep Biswas, Brian Coltin, and Stephanie Rosenthal. Cobots: robust symbiotic autonomous mobile service robots. In *Proceedings of the International Joint Conference on Artificial Intelligence*. Citeseer, 2015.

[3] Julian Jara-Ettinger. Theory of Mind as inverse reinforcement learning. *Current Opinion in Behavioral Sciences*, 29:105–110, 2019.

[4] Alan M Leslie, Ori Friedman, and Tim P German. Core mechanisms in Theory of Mind. *Trends in cognitive sciences*, 8(12):528–533, 2004.

[5] Gergely Csibra and György Gergely. The teleological origins of mentalistic action explanations: a developmental hypothesis. *Developmental science*, 1(2):255–259, 1998.

[6] Rebecca Saxe. Against simulation: the argument from error. *Trends in cognitive sciences*, 9(4):174–179, 2005.

[7] Julian Jara-Ettinger, Hyowon Gweon, Laura E Schulz, and Joshua B Tenenbaum. The naïve utility calculus: Computational principles underlying commonsense psychology. *Trends in cognitive sciences*, 20(8):589–604, 2016.

[8] Miquel Ramírez and Hector Geffner. Plan recognition as planning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2009.

[9] Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[10] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. XAI—explainable artificial intelligence. *Science robotics*, 4 (37):eaay7120, 2019.

[11] Hyowon Gweon, Judith Fan, and Been Kim. Socially intelligent machines that learn from humans and help humans learn. *Philosophical Transactions of the Royal Society A*, 381(2251):20220048, 2023.

[12] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. Plan explanations as model reconciliation: moving beyond explanation as soliloquy. *arXiv preprint arXiv:1701.08317*, 2017.

[13] Tathagata Chakraborti, Anagha Kulkarni, Sarath Sreedharan, David E Smith, and Subbarao Kambhampati. Explicability? Legibility? Predictability? Transparency? Privacy? Security? the emerging landscape of interpretable agent behavior. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 86–96, 2019.

[14] Aleck M MacNally, Nir Lipovetzky, Miquel Ramirez, and Adrian R Pearce. Action selection for transparent planning. In *Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems*, pages 1327–1335, 2018.

[15] Marcelo G Mattar and Máté Lengyel. Planning in the brain. *Neuron*, 110(6): 914–934, 2022.

[16] Tirtharaj Dash, Sharad Chitlangia, Aditya Ahuja, and Ashwin Srinivasan. A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Scientific Reports*, 12(1):1040, 2022.

[17] Roger Ratcliff, Philip L Smith, Scott D Brown, and Gail McKoon. Diffusion decision model: current issues and history. *Trends in cognitive sciences*, 20(4): 260–281, 2016.

[18] Nitzan Shahar, Tobias U Hauser, Michael Moutoussis, Rani Moran, Mehdi Keramati, Nspn Consortium, and Raymond J Dolan. Improving the reliability of model-based decision-making estimates in the two-stage decision task with reaction-times and drift-diffusion modeling. *PLoS computational biology*, 15(2):e1006803, 2019.

[19] Marlene Berke, Abigail Tenenbaum, Ben Sterling, and Julian Jara-Ettinger. Thinking about thinking as rational computation. *PsyArXiv*, 2023.

[20] Hector Geffner. Computational models of planning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 4(4):341–356, 2013.

[21] Blai Bonet and Hector Geffner. Heuristic search planner 2.0. *AI Magazine*, 22(3): 77–77, 2001.

[22] Allen Newell, Herbert A Simon, et al. *Human problem solving*, volume 104. Prentice-hall Englewood Cliffs, NJ, 1972.

[23] John E Laird. *The Soar cognitive architecture*. MIT press, 2019.

[24] Sashank Varma and Marcel Adam Just. 4CAPS: an adaptive architecture for human information processing. In *AAAI Spring Symposium: Between a Rock and a Hard Place: Cognitive Science Principles Meet AI-Hard Problems*, pages 91–96, 2006.

[25] Qing Rao and Jelena Frtunikj. Deep learning for self-driving cars: chances and challenges. In *Proceedings of the International Workshop on Software Engineering for AI in Autonomous Systems*, pages 35–38, 2018.

[26] Dedre Gentner and Julie Colhoun. Analogical processes in human thinking and learning. *Towards a theory of thinking: Building blocks for a conceptual framework*, pages 35–48, 2010.

[27] Toshihiko Matsuka. Modeling human learning as context dependent knowledge utility optimization. In *Advances in Natural Computation: First International Conference, ICNC 2005, Changsha, China, August 27-29, 2005, Proceedings, Part I 1*, pages 933–946. Springer, 2005.

[28] Mehdi Keramati, Amir Dezfouli, and Payam Piray. Speed/accuracy trade-off between the habitual and the goal-directed processes. *PLoS computational biology*, 7(5):e1002055, 2011.

[29] Dirk Ruiz and Allen Newell. Tower-noticing triggers strategy-change in the Tower of Hanoi: a Soar model. Technical report, Carnegie Mellon University Dept of Psychology, 1989.

[30] Kenneth Kotovsky, John R Hayes, and Herbert A Simon. Why are some problems hard? evidence from Tower of Hanoi. *Cognitive psychology*, 17(2):248–294, 1985.

[31] Marilyn C Welsh, Trey Satterlee-Cartmell, and Michelle Stine. Towers of Hanoi and London: contribution of working memory and inhibition to performance. *Brain and cognition*, 41(2):231–242, 1999.

[32] John R Anderson and Scott Douglass. Tower of Hanoi: evidence for the cost of goal retrieval. *Journal of experimental psychology: learning, memory, and cognition*, 27(6):1331, 2001.

[33] Francesco Donnarumma, Domenico Maisto, and Giovanni Pezzulo. Problem solving as probabilistic inference with subgoaling: explaining human successes and pitfalls in the Tower of Hanoi. *PLoS computational biology*, 12(4):e1004864, 2016.

[34] John Slaney and Sylvie Thiébaux. Blocks World revisited. *Artificial Intelligence*, 125(1-2):119–153, 2001.

[35] Louise Helen Phillips, VE Wynn, S McPherson, and KJ Gilhooly. Mental planning and the Tower of London task. *The Quarterly Journal of Experimental Psychology Section A*, 54(2):579–597, 2001.

[36] W Keith Berg, Dana L Byrd, Joseph PH McNamara, and Kimberly Case. Deconstructing the tower: parameters and predictors of problem difficulty on the Tower of London task. *Brain and Cognition*, 72(3):472–482, 2010.

[37] Miguel Ramírez and Hector Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2010.

[38] Mor Vered, Gal A Kaminka, and Sivan Biham. Online goal recognition through mirroring: humans and agents. In *The Annual Conference on Advances in Cognitive Systems*, 2016.

[39] Ramon Pereira, Nir Oren, and Felipe Meneguzzi. Landmark-based heuristics for goal recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[40] Andreas Junghanns and Jonathan Schaeffer. Sokoban: enhancing general single-agent search methods using domain knowledge. *Artificial Intelligence*, 129(1-2): 219–251, 2001.

[41] Petr Jarušek and Radek Pelánek. Human problem solving: Sokoban case study. *Technická zpráva, Fakulta informatiky, Masarykova univerzita, Brno*, 2010.

[42] Petr Jarušek and Radek Pelánek. Difficulty rating of Sokoban puzzle. In *STAIRS 2010*, pages 140–150. IOS Press, 2010.

[43] Patrik Haslum, Adi Botea, Malte Helmert, Blai Bonet, Sven Koenig, et al. Domain-independent construction of pattern database heuristics for cost-optimal planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 7, pages 1007–1012, 2007.

[44] Zhao Yang, Mike Preuss, and Aske Plaat. Transfer learning and curriculum learning in Sokoban. In *Artificial Intelligence and Machine Learning: 33rd Benelux Conference on Artificial Intelligence, BNAIC/Benelearn 2021, Esch-sur-Alzette, Luxembourg, November 10–12, 2021, Revised Selected Papers 33*, pages 187–200. Springer, 2022.

[45] Zhao Yang, Mike Preuss, and Aske Plaat. Potential-based reward shaping in Sokoban. *arXiv preprint arXiv:2109.05022*, 2021.

[46] Daniel Reisberg. *The Oxford handbook of cognitive psychology*. OUP USA, 2013.

[47] Carmen M Laterell. What is problem-solving ability. *LATM Journal*, 1(1):1–12, 2013.

[48] Alberto Colorni, Marco Dorigo, Francesco Maffioli, Vittorio Maniezzo, Giovanni Righini, and Marco Trubian. Heuristics from nature for hard combinatorial optimization problems. *International Transactions in Operational Research*, 3(1): 1–21, 1996.

[49] Iris Van Rooij, Cory D Wright, and Todd Wareham. Intractability and the use of heuristics in psychological explanations. *Synthese*, 187(2):471–487, 2012.

[50] Leo Gugerty. Newell and Simon's logic theorist: historical background and impact on cognitive modeling. In *Proceedings of the Human Factors and Ergonomics*

Society Annual Meeting, volume 50, pages 880–884. SAGE Publications Sage CA: Los Angeles, CA, 2006.

[51] Sashank Varma. *A computational model of Tower of Hanoi problem solving*. PhD thesis, 2006.

[52] James N MacGregor and Yun Chu. Human performance on the traveling salesman and related problems: a review. *The Journal of Problem Solving*, 3(2):2, 2011.

[53] Mary L Gick and Keith J Holyoak. Analogical problem solving. *Cognitive psychology*, 12(3):306–355, 1980.

[54] Pedro A Tsividis, Thomas Pouncy, Jaqueline L Xu, Joshua B Tenenbaum, and Samuel J Gershman. Human learning in Atari. *AAAI spring symposium series*, 2017.

[55] Adel Saadi, Ramdane Maamri, and Zaidi Sahnoun. Behavioral flexibility in belief-desire-intention (bdi) architectures. *Multiagent and Grid Systems*, 16(4):343–377, 2020.

[56] John E Laird. *The Soar cognitive architecture*. MIT press, 2012.

[57] Frank E Ritter, Farnaz Tehranchi, and Jacob D Oury. ACT-R: a cognitive architecture for modeling cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 10(3):e1488, 2019.

[58] Alec Solway and Matthew M Botvinick. Evidence integration in model-based tree search. *Proceedings of the National Academy of Sciences*, 112(37):11708–11713, 2015.

[59] Ionatan Kuperwajs, Bas Van Opheusden, and Wei Ji Ma. Prospective planning and retrospective learning in a large-scale combinatorial game. In *The Conference on Cognitive Computational Neuroscience*, pages 13–16, 2019.

[60] Sylvain Gelly and David Silver. Monte Carlo Tree search and rapid action value estimation in computer Go. *Artificial Intelligence*, 175(11):1856–1875, 2011.

[61] Moritz JF Krusche, Eric Schulz, Arthur Guez, and Maarten Speekenbrink. Adaptive planning in human search. *BioRxiv*, page 268938, 2018.

[62] Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Mańdziuk. Monte Carlo Tree search: a review of recent modifications and applications. *Artificial Intelligence Review*, 56(3):2497–2562, 2023.

[63] Frederick Callaway, Falk Lieder, Priyam Das, Sayan Gul, Paul M Krueger, and Tom Griffiths. A resource-rational analysis of human planning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2018.

[64] F Callaway, B van Opheusden, S Gul, P Das, P Krueger, F Lieder, and T Griffiths. Human planning as optimal information seeking. *Manuscript in preparation*, 2021.

[65] Falk Lieder and Thomas L Griffiths. Resource-rational analysis: understanding human cognition as the optimal use of limited computational resources. *Behavioral and brain sciences*, 43:e1, 2020.

[66] Kenji Doya. *Bayesian brain: Probabilistic approaches to neural coding*. MIT press, 2007.

[67] Matthew Botvinick and Marc Toussaint. Planning as inference. *Trends in cognitive sciences*, 16(10):485–488, 2012.

[68] Jim X Chen. The evolution of computing: Alphago. *Computing in Science & Engineering*, 18(4):4–7, 2016.

[69] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553):436–444, 2015.

[70] Hector Geffner. Model-free, model-based, and general intelligence. *arXiv preprint arXiv:1806.02308*, 2018.

[71] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5): 206–215, 2019.

[72] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*, pages 270–279. Springer, 2018.

[73] Tim Miller. Explanation in artificial intelligence: insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[74] David Gunning. Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency, nd Web*, 2(2), 2017.

[75] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017.

[76] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[77] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[78] Anthony R Cassandra. A survey of POMDP applications. In *Working notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes*, volume 1724, 1998.

[79] Dongyan Chen and Kishor S Trivedi. Optimization for condition-based maintenance with semi-markov decision process. *Reliability engineering & system safety*, 90(1):25–29, 2005.

[80] Ronald Edward Parr. *Hierarchical control and learning for Markov decision processes*. University of California, Berkeley, 1998.

[81] Maria Fox and Derek Long. PDDL2. 1: an extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research*, 20:61–124, 2003.

[82] Patrik Haslum, Nir Lipovetzky, Daniele Magazzeni, and Christian Muise. An introduction to the planning domain definition language. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(2):1–187, 2019.

[83] Blai Bonet and Héctor Geffner. HSP: heuristic search planner. *AIPS-98 Planning Competition*, pages 60–72, 1998.

[84] Jörg Hoffmann and Bernhard Nebel. The FF planning system: fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, 2001.

[85] Nir Lipovetzky and Hector Geffner. Best-first width search: exploration and exploitation in classical planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[86] Nir Lipovetzky and Hector Geffner. Width and serialization of classical planning problems. In *Proceedings of the European Conference on Artificial Intelligence*, pages 540–545, 2012.

[87] Alonso H Vera and Herbert A Simon. Situated action: a symbolic interpretation. *Cognitive science*, 17(1):7–48, 1993.

[88] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial intelligence*, 49(1-3):61–95, 1991.

[89] William Cushing, Daniel S Weld, Subbarao Kambhampati, K Talamadupula Mausam, and Kartik Talamadupula. Evaluating temporal planning domains. In *Proceedings of the International Conference on Automated Planning and Scheduling*, pages 105–112, 2007.

[90] Amanda Coles, Andrew Coles, Maria Fox, and Derek Long. Forward-chaining partial-order planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 20, pages 42–49, 2010.

[91] Patrick Eyerich, Robert Mattmüller, and Gabriele Röger. Using the context-enhanced additive heuristic for temporal and numeric planning. In *Towards Service Robots for Everyday Environments: Recent Advances in Designing Service Robots for Complex Tasks in Everyday Environments*, pages 49–64. Springer, 2012.

[92] Dorit Dor and Uri Zwick. Sokoban and other motion planning problems. *Computational Geometry*, 13(4):215–228, 1999.

[93] Gilbert Laporte. The traveling salesman problem: an overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2):231–247, 1992.

[94] Clair E Miller, Albert W Tucker, and Richard A Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4): 326–329, 1960.

[95] Marco Caserta and Stefan Voß. A hybrid algorithm for the DNA sequencing problem. *Discrete Applied Mathematics*, 163:87–99, 2014.

[96] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3): 52–57, 2002.

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

[98] Mark K Ho, David Abel, Carlos G Correa, Michael L Littman, Jonathan D Cohen, and Thomas L Griffiths. People construct simplified mental representations to plan. *Nature*, 606(7912):129–136, 2022.

[99] Bradley Hayes and Brian Scassellati. Autonomously constructing hierarchical task networks for planning and human-robot collaboration. In *IEEE International Conference on Robotics and Automation*, pages 5469–5476. IEEE, 2016.

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

[101] Cordell Green. Application of theorem proving to problem solving. In *Readings in Artificial Intelligence*, pages 202–222. Elsevier, 1981.

[102] Sharlene D Newman, Patricia A Carpenter, Sashank Varma, and Marcel Adam Just. Frontal and parietal participation in problem solving in the Tower of London: fMRI and computational modeling of planning and high-level perception. *Neuropsychologia*, 41(12):1668–1682, 2003.

[103] Geoff Ward and Alan Allport. Planning and problem solving using the five disc Tower of London task. *The Quarterly Journal of Experimental Psychology Section A*, 50(1):49–78, 1997.

[104] Allen Newell, Herbert Alexander Simon, et al. *Human problem solving*, volume 104. Prentice-hall Englewood Cliffs, NJ, 1972.

[105] Seçkin Yılmaz and Vasif V Nabiyev. Comprehensive survey of the solving puzzle problems. *Computer Science Review*, 50:100586, 2023.

[106] Smaragda Markaki and Costas Panagiotakis. Jigsaw puzzle solving techniques and applications: a survey. *The Visual Computer*, pages 1–17, 2022.

[107] Michal Gregor, Katarína Zábovská, and Vladimír Smataník. The zebra puzzle and getting to know your tools. In *IEEE International Conference on Intelligent Engineering Systems*, pages 159–164. IEEE, 2015.

[108] Jan M Wiener, Simon J Büchner, and Christoph Hölscher. Taxonomy of human wayfinding tasks: a knowledge-based approach. *Spatial Cognition & Computation*, 9(2):152–165, 2009.

[109] Reuth Mirsky, Kobi Gal, Roni Stern, and Meir Kalech. Goal and plan recognition design for plan libraries. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–23, 2019.

[110] Franz A Van-Horenbeke and Angelika Peer. Activity, plan, and goal recognition: a review. *Frontiers in Robotics and AI*, 8:643010, 2021.

[111] Artem Polyvyanyy, Zihang Su, Nir Lipovetzky, and Sebastian Sardina. Goal recognition using off-the-shelf process mining techniques. In *Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems*, pages 1072–1080, 2020.

[112] Mattia Chiari, Alfonso Emilio Gerevini, Francesco Percassi, Luca Putelli, Ivan Serina, and Matteo Olivato. Goal recognition as a deep learning task: the grnet approach. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 33, pages 560–568, 2023.

[113] Nusrat Jahan Tithi and Swakkhar Shatabda. A convolutional neural network for goal recognition. In *2023 26th International Conference on Computer and Information Technology (ICCIT)*, pages 1–6. IEEE, 2023.

[114] Chris Frith and Uta Frith. Theory of Mind. *Current biology*, 15(17):R644–R645, 2005.

[115] Renée Baillargeon, Rose M Scott, Zijing He, Stephanie Sloane, Peipei Setoh, Kyong-sun Jin, Di Wu, and Lin Bian. *Psychological and sociomoral reasoning in infancy.* American Psychological Association, 2015.

[116] Martin Doherty. *Theory of Mind: how children understand others' thoughts and feelings.* psychology press, 2008.

[117] Chris L Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B Tenenbaum. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):0064, 2017.

[118] Chris Baker, Rebecca Saxe, and Joshua Tenenbaum. Bayesian Theory of Mind: modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

[119] Russell Stuart and Peter Norvig. Artificial intelligence: a modern approach (global edition). *Harlow: Pearson*, 2016.

[120] Peta Masters and Sebastian Sardina. Goal recognition for rational and irrational agents. In *Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems*, pages 440–448, 2019.

[121] Allen Newell and Herbert Simon. The logic theory machine–a complex information processing system. *IRE Transactions on information theory*, 2(3):61–79, 1956.

[122] Michael E Atwood and Peter G Polson. A process model for water jug problems. *Cognitive Psychology*, 8(2):191–216, 1976.

[123] Stellan Ohlsson. The problems with problem solving: reflections on the rise, current status, and possible future of a cognitive research paradigm. *The Journal of Problem Solving*, 5(1):101–128, 2012.

[124] Milica Mormann, Jonathan Malmaud, Alexander Huth, Christof Koch, and Antonio Rangel. The drift diffusion model can account for the accuracy and reaction time of value-based choices under high and low time pressure. *Judgment and Decision Making*, 5(6):437–449, 2010.

[125] Herbert A Simon. Bounded rationality. In *Utility and probability*, pages 15–18. Springer, 1990.

[126] John R Anderson. A rational analysis of human. *Varieties of memory and consciousness: Essays in honour of Endel Tulving*, page 195, 1989.

[127] Michael L Anderson and Tim Oates. A review of recent research in metareasoning and metalearning. *AI Magazine*, 28(1):12–12, 2007.

[128] Satohiro Tajima, Jan Drugowitsch, Nisheet Patel, and Alexandre Pouget. Optimal policy for multi-alternative decisions. *Nature neuroscience*, 22(9):1503–1511, 2019.

[129] Christian Lebiere and John R Anderson. A connectionist implementation of the ACT-R production system. In *Proceedings of the Annual Conference of the Cognitive Science Society*, pages 635–640, 1993.

[130] John E Laird, Allen Newell, and Paul S Rosenbloom. Soar: an architecture for general intelligence. *Artificial intelligence*, 33(1):1–64, 1987.

[131] Marta Kryven, Max Kleiman-Weiner, Joshua Tenenbaum, and Suhyoun Yu. Planning ahead in spatial search. *PsyArXiv*, 2022.

[132] Björn Meder, Jonathan D Nelson, Matt Jones, and Azzurra Ruggeri. Stepwise versus globally optimal search in children and adults. *Cognition*, 191:103965, 2019.

[133] Mehdi Keramati, Peter Smittenaar, Raymond J Dolan, and Peter Dayan. Adaptive integration of habits into depth-limited planning defines a habitual-goal–directed spectrum. *Proceedings of the National Academy of Sciences*, 113(45):12868–12873, 2016.

[134] Bas Van Opheusden, Gianni Galbiati, Zahy Bnaya, Yunqi Li, and Wei Ji Ma. A computational model for decision tree search. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2017.

[135] Christoph P Kaller, Josef M Unterrainer, Benjamin Rahm, and Ulrike Halsband. The impact of problem structure on planning: insights from the Tower of London task. *Cognitive Brain Research*, 20(3):462–472, 2004.

[136] Christoph P Kaller, Benjamin Rahm, Lena Köstering, and Josef M Unterrainer. Reviewing the impact of problem structure on planning: a software tool for analyzing tower tasks. *Behavioural brain research*, 216(1):1–8, 2011.

[137] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[138] Herbert A Simon. Experiments with a heuristic compiler. *Journal of the ACM*, 10(4):493–506, 1963.

[139] Manuel Heusner, Thomas Keller, and Malte Helmert. Understanding the search behaviour of greedy best-first search. In *Proceedings of the International Symposium on Combinatorial Search*, volume 8, pages 47–55, 2017.

[140] Vadim Bulitko, Ilya Levner, and Russ Greiner. Real-time lookahead control policies. In *Joint Workshop on Real-Time Decision Support and Diagnosis Systems, AAAI*, 2002.

[141] Levente Kocsis and Csaba Szepesvári. Bandit based Monte Carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

[142] Miquel Ramirez, Nir Lipovetzky, and Christian Muise. Lightweight Automated Planning ToolKiT. http://lapkt.org/, 2015. Accessed: 2020.

[143] Joshua R De Leeuw. jsPsych: a Javascript library for creating behavioral experiments in a web browser. *Behavior research methods*, 47(1):1–12, 2015.

[144] Guillem Francés, Miquel Ramirez, and Collaborators. Tarski: an AI planning modeling framework. https://github.com/aig-upf/tarski, 2018.

[145] Tan Zhi-Xuan, Jordyn L Mann, Tom Silver, Joshua B Tenenbaum, and Vikash K Mansinghka. Online Bayesian goal inference for boundedly-rational planning agents. *arXiv preprint arXiv:2006.07532*, 2020.

[146] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design for non-optimal agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[147] Nate Blaylock, James Allen, et al. Corpus-based, statistical goal recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 3, pages 1303–1308, 2003.

[148] Shirin Sohrabi, Anton V Riabov, and Octavian Udrea. Plan recognition as planning revisited. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3258–3264, 2016.

[149] Ronal Singh, Tim Miller, Joshua Newn, Liz Sonenberg, Eduardo Velloso, and Frank Vetere. Combining planning with gaze for online human intention recognition. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pages 488–496, 2018.

[150] Vael Gates, Frederick Callaway, Mark K Ho, and Thomas L Griffiths. A rational model of people's inferences about others' preferences based on response times. *Cognition*, 217:104885, 2021.

[151] Dorit Avrahami-Zilberbrand, Gal Kaminka, and Hila Zarosim. Fast and complete symbolic plan recognition: allowing for duration, interleaved execution, and lossy observations. In *Proceedings of the AAAI Workshop on Modeling Others from Observations, MOO*, pages 25–19. Citeseer, 2005.

[152] Moser Silva Fagundes, Felipe Rech Meneguzzi, Rafael H Bordini, and Renata Vieira. Dealing with ambiguity in plan recognition under time constraints. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems, 2014, França.*, 2014.

[153] Gabriela Tavares, Pietro Perona, and Antonio Rangel. The attentional drift diffusion model of simple perceptual decision-making. *Frontiers in neuroscience*, 11: 468, 2017.

[154] Alan M Leslie. Pretense and representation: the origins of Theory of Mind. *Psychological review*, 94(4):412, 1987.

[155] Michael Rescorla. The Computational Theory of Mind. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2020 edition, 2020.

[156] Neil C Rabinowitz, Frank Perbet, H Francis Song, Chiyuan Zhang, SM Eslami, and Matthew Botvinick. Machine Theory of Mind. *arXiv preprint arXiv:1802.07740*, 2018.

[157] Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. *arXiv preprint arXiv:1802.09640*, 2018.

[158] Mark K Ho, David Abel, Jonathan D Cohen, Michael L Littman, and Thomas L Griffiths. The efficiency of human cognition reflects planned information processing. *arXiv preprint arXiv:2002.05769*, 2020.

[159] Dan Amir and Ofra Amir. Highlights: summarizing agent behavior to people. In *Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems*, pages 1168–1176, 2018.

[160] Luísa RA Santos, Felipe Meneguzzi, Ramon Fraga Pereira, and André Grahl Pereira. An LP-based approach for goal recognition as planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11939–11946, 2021.

[161] Tianyi Gu, Wheeler Ruml, Shahaf S Shperberg, Eyal Solomon Shimony, and Erez Karpas. When to commit to an action in online planning and search. In *Proceedings of the International Symposium on Combinatorial Search*, volume 15, pages 83–90, 2022.

[162] Eugene Charniak and Robert P Goldman. A Bayesian model of plan recognition. *Artificial Intelligence*, 64(1):53–79, 1993.

[163] Sascha Topolinski, Giti Bakhtiari, and Thorsten M Erle. Can I cut the Gordian tnok? the impact of pronounceability, actual solvability, and length on intuitive problem assessments of anagrams. *Cognition*, 146:439–452, 2016.

[164] Laura R Novick and Steven J Sherman. On the nature of insight solutions: evidence from skill differences in anagram solution. *The Quarterly Journal of Experimental Psychology Section A*, 56(2):351–382, 2003.

[165] Kin Max Gusmao, Ramon Fraga Pereira, and Felipe Meneguzzi. Inferring agents preferences as priors for probabilistic goal recognition. *arXiv preprint arXiv:2102.11791*, 2021.

[166] Peta Masters, Michael Kirley, and Wally Smith. Extended goal recognition: a planning-based model for strategic deception. In *Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems*, pages 871–879, 2021.

[167] Spencer E Harpe. How to analyze Likert and other rating scale data. *Currents in pharmacy teaching and learning*, 7(6):836–850, 2015.

[168] Donald O Case and Lisa M Given. *Looking for information: a survey of research on information seeking, needs, and behavior*. Emerald Group Publishing, 2016.

[169] Abeer Alshehri, Tim Miller, and Mor Vered. Explainable goal recognition: a framework based on weight of evidence. *arXiv preprint arXiv:2303.05622*, 2023.

[170] Thomas Keller and Malte Helmert. Trial-based heuristic tree search for finite horizon MDPs. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 23, pages 135–143, 2013.

[171] Joseph Rafferty, Chris D Nugent, Jun Liu, and Liming Chen. From activity recognition to intention recognition for assisted living within smart homes. *IEEE Transactions on Human-Machine Systems*, 47(3):368–379, 2017.

[172] Quentin JM Huys, Níall Lally, Paul Faulkner, Neir Eshel, Erich Seifritz, Samuel J Gershman, Peter Dayan, and Jonathan P Roiser. Interplay of approximate planning strategies. *Proceedings of the National Academy of Sciences*, 112(10):3098–3103, 2015.

[173] Hankz Hankui Zhuo, Derek Hao Hu, Chad Hogg, Qiang Yang, and Hector Munoz-Avila. Learning HTN method preconditions and action models from partial observations. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2009.

[174] Jette Randlov. Learning macro-actions in reinforcement learning. *Advances in Neural Information Processing Systems*, 11, 1998.

[175] Stefan O'Toole, Miquel Ramirez, Nir Lipovetzky, and Adrian R Pearce. Sampling from pre-images to learn heuristic functions for classical planning. In *Proceedings of the International Symposium on Combinatorial Search*, volume 15, pages 308–310, 2022.

[176] Jon Driver. A selective review of selective attention research from the past century. *British journal of psychology*, 92(1):53–78, 2001.

[177] Ronald Hübner, Marco Steinhauser, and Carola Lehle. A dual-stage two-phase model of selective attention. *Psychological review*, 117(3):759, 2010.

[178] Jonathan Cagan and Kenneth Kotovsky. Simulated annealing and the generation of the objective function: a model of learning during problem solving. *Computational Intelligence*, 13(4):534–581, 1997.

[179] David Z Hambrick and Randall W Engle. The role of working memory in problem solving. *The psychology of problem solving*, pages 176–206, 2003.

[180] Jennifer Wiley and Andrew F Jarosz. How working memory capacity affects problem solving. In *Psychology of learning and motivation*, volume 56, pages 185–227. Elsevier, 2012.

[181] Edward J Powley, Peter I Cowling, and Daniel Whitehouse. Information capture and reuse strategies in Monte Carlo Tree search, with applications to games of hidden information. *Artificial Intelligence*, 217:92–116, 2014.

[182] Mark D'Esposito and Bradley R Postle. The cognitive neuroscience of working memory. *Annual review of psychology*, 66:115–142, 2015.

[183] Royce R Ronning, Donald McCurdy, and Ruth Ballinger. Individual differences: a third component in problem-solving instruction. *Journal of Research in Science Teaching*, 21(1):71–82, 1984.

[184] Carl P Duncan. Recent research on human problem solving. *Psychological Bulletin*, 56(6):397, 1959.

[185] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017.

[186] Anja F Ernst, Marieke E Timmerman, Bertus F Jeronimus, and Casper J Albers. Insight into individual differences in emotion dynamics with clustering. *Assessment*, 28(4):1186–1206, 2021.

[187] Daniele Asioli, Ingunn Berget, and Tormod Næs. Comparison of different clustering methods for investigating individual differences using choice experiments. *Food Research International*, 111:371–378, 2018.

[188] Joeri Hofmans and Etienne Mullet. Towards unveiling individual differences in different stages of information processing: a clustering-based approach. *Quality & Quantity*, 47:455–464, 2013.

[189] Robert S Siegler. Individual differences in strategy choices: good students, not-so-good students, and perfectionists. *Child development*, pages 833–851, 1988.

[190] Breda Cullen, Brian O'Neill, Jonathan J Evans, Robert F Coen, and Brian A Lawlor. A review of screening tests for cognitive impairment. *Journal of Neurology, Neurosurgery & Psychiatry*, 78(8):790–799, 2007.

[191] Sian L Beilock and Thomas H Carr. When high-powered people fail: working memory and "choking under pressure" in math. *Psychological science*, 16(2):101–105, 2005.

[192] Joseph Rafferty, Liming Chen, Chris Nugent, and Jun Liu. Goal lifecycles and ontological models for intention based assistive living within smart environments. 2015.

[193] Nils Wilken and Heiner Stuckenschmidt. Combining symbolic and statistical knowledge for goal recognition in smart home environments. In *IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pages 26–31. IEEE, 2021.

[194] Leonardo Amado, Reuth Mirsky, and Felipe Meneguzzi. Goal recognition as reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9644–9651, 2022.

[195] Yunxiu Zeng, Kai Xu, Quanjun Yin, Long Qin, Yabing Zha, and William Yeoh. Inverse reinforcement learning based human behavior modeling for goal recognition in dynamic local network interdiction. In *AAAI Workshops*, pages 646–653, 2018.

[196] William Shen, Felipe Trevizan, and Sylvie Thiébaux. Learning domain-independent planning heuristics with hypergraph networks. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 574–584, 2020.

[197] Dozdar Mahdi Ahmed, Masoud Muhammed Hassan, and Ramadhan J Mstafa. A review on deep sequential models for forecasting time series data. *Applied Computational Intelligence and Soft Computing*, 2022, 2022.

[198] Paul Newbold. Arima model building and the time series analysis approach to forecasting. *Journal of forecasting*, 2(1):23–35, 1983.

[199] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation*, 31 (7):1235–1270, 2019.

[200] Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*, 2017.

[201] M Marcos, Ruth García-Gavilanes, Emad Bataineh, and Lara Pasarin. Using eye tracking to identify cultural differences in information seeking behavior. *Many people, many eyes: aggregating influences of visual perception on user interface design. CHI'13: Extended Abstracts on Human Factors in Computing Systems*, 2014.

[202] Julian Jara-Ettinger, Joshua Tenenbaum, and Laura Schulz. Not so innocent: reasoning about costs, competence, and culpability in very early childhood. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 35, 2013.

[203] Herbert A Simon and Allen Newell. Human problem solving: the state of the theory in 1970. *American Psychologist*, 26(2):145, 1971.