RMIT UNIVERSITY

Goal Recognition and Deception in Path-Planning

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

Peta Louise Masters

Bachelor of Computer Science (Honours), RMIT University

School of Science

College of Science, Engineering and Health

RMIT University

February, 2019

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Peta Louise Masters
February, 2019

Acknowledgments

I would like to thank Sebastian Sardina, my supervisor since Honours, the best and most generous teacher at RMIT (to my knowledge :o). Thank you for teaching me always to think harder and to look more carefully and more critically, even if it risks finding out that everything I did so far was wrong. You told me at the start of this that, if I worked hard, I might be able to create a tiny pimple on the giant sphere of human knowledge. This is our pimple.

To my associate supervisor, Lin Padgham: thank you for your wisdom and support and for being a beacon to me where I'd certainly otherwise have stumbled along this path. How lucky am I to have had such a shining example.

Thanks to Lisa, Emma and Claire in HDR for taking care of all the administrative stuff, more complex than anything I encountered in AI. Without you, I'd still be trying to work out what a "confirmation" was and whether I needed to buy myself a white dress for it. And to my co-conspiritors in the 'Robot Lab', Chaminda, Behrooz, Mohammed and absent friends, Shahriar and Marco: I won't say we've exactly had fun but I will say that our time together has been unforgettable!

Finally, thanks to Ann. What you have had to put up with! OMG.

Research Output

Some of the material in this thesis has appeared (or will shortly appear) in the following publications.

- P. Masters and S. Sardina. Goal recognition for rational and irrational agents. *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 440-448, 2019.
- P. Masters and S. Sardina. Cost-based goal recognition in navigational domains. *Journal of Artificial Intelligence Research (JAIR), 64*, pages 197-242, 2019.
- P. Masters and S. Sardina. Cost-based goal recognition for the path-planning domain. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, sister conference, pages 5329-5333, 2018.
- P. Masters and S. Sardina. Deceptive path-planning. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4368-4375, 2017.
- P. Masters and S. Sardina. Cost-based goal recognition for path-planning. *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 750-758, 2017, (**winner of the Pragnesh Jay Modi Best Student Paper Award**).

Our Python-based simulator and test environment is available for public use and may be downloaded from `https://tinyurl.com/p4sim`.

Dedication

To my father Schera Morris Masters and my good friend Angela Langfield, who between them funded the purchase of my first computer: a 'top-of-the-range' 8-bit Commodore 128. Without that, nothing.

"We run things; things don't run we."

–Miley Cyrus

Contents

List of Figures

List of Tables

Listings

Notation and Conventions

This document uses the following technical conventions.

Notation	**Explanation**	**Notes**
α, β	Constants	Greek characters
$c : E \mapsto \mathbb{R}$, *costdif*$(\cdot)$	Functions	lower case with brackets
D_c, $costdif_{RG}$	Identifiers	subscript
$n_1, n_2, \pi^i, \pi^{i+1}$	Indices	subscript, superscript
i, j, k	Integers/counters	letters i, j, k
A*, Moving-AI	Names of models/systems	sans-serif
Π^*	Optimal (and Klene star as usual)	asterisk
π, Π	Path, set of paths	special case
$Prob$	Prior probability distribution	
$\vec{o} = o_1, \ldots, o_k$	Sequences (but see also π)	vector
$o \in \vec{o}$	o occurs somewhere in the sequence $\vec{o}$	
$s \in S, g \in G \setminus \{g'\}$	Set members	lower case
$S, G \subset N$	Sets	upper case
$\mathcal{P}, \mathcal{D}_c$	Tuples	maths font

Abbreviations

This document uses the following abbreviations.

AI	artificial intelligence
ARMOR	assistant for randomized monitoring over routes
CCTV	closed circuit television
BDI	belief, desire, intent
DPP	deceptive path-planning
FSA	finite state automaton
FTP	first truthful point
GPS	global positioning system
GR	goal recognition
HMI	human-machine interaction
LDP	last deceptive point
PAIR	plan, activity and intent recognition
POMDP	partially observable Markov decision process
PROTECT	port resilience for operational/tactical enforcement to combat terrorism
R&G	Ramirez and Geffner (their model of goal recognition)
RMIT	Royal Melbourne Institute of Technology
RMP	radius of maximum probability
STRIPS	Stanford Research Institute Problem Solver
V&K	Vered and Kaminka (their model of goal recognition)
VIP	very important person

Abstract

This thesis argues that investigation of goal recognition and deception in the much studied and well-understood context of path-planning reveals nuances to both problems that have previously gone unnoticed. Contemporary goal recognition systems rely on examination of multiple observations to calculate a probability distribution across goals. The first part of this thesis demonstrates that a distribution with identical rankings to current state-of-the-art can be achieved without any observations apart from a known starting point (such as a door or gate) and where the agent is now. It also presents a closed formula to calculate a radius around any goal of interest within which that goal is guaranteed to be the most probable, without having to calculate any actual probability values. In terms of deception, traditionally there are two strategies: dissimulation (hiding the true) and simulation (showing the false). The second part of this thesis shows that current state-of-the-art goal recognition systems do not cope well with dissimulation that does its work by 'dazzling' (i.e., obfuscating with hugely suboptimal plans). It presents an alternative, self-modulating formula that modifies its output when it encounters suboptimality, seeming to 'know that it does not know' instead of 'keep changing its mind'. Deception is often regarded as a 'yes, no' proposition (either the target is deceived or they are not). Furthermore, intuitively, deceptive path-planning involves suboptimality and must, therefore, be expensive. This thesis, however, presents a model of deception for path-planning domains within which it is possible (a) to rank paths by their potential to deceive and (b) to generate deceptive paths that are 'optimally deceptive' (i.e., deceptive to the maximum extent at the lowest cost).

CHAPTER 1

Introduction

"The road up and the road down are one and the same."

–Heraclitus

Goal recognition and deception are two sides of the same coin: the yin and yang of observable behaviour. Surveillance, privacy; exposure, protection; curiosity, concealment. These are problems whose solutions are useful, important and inextricably linked. Goal recognition involves determining an agent's intent by observing her behaviour; deception involves the observed agent in executing behaviour that conceals her intent. The relationship is symbiotic. Consider a crime-writer, working on a new plot. Why bother coming up with new twists and turns, why bother to throw in all those red herrings, if not for a reader continually trying to work out what will happen next? Consider an athlete feinting left before she moves right. Why feint at all, if not for an opponent observing the behaviour and attempting to recognise which way she will run?

This thesis studies the problems of goal recognition and deception in the context of path-planning. Given the initial location of an agent, a set of possible goals and an incomplete sequence of places where the agent has already been, goal recognition is the problem of determining which of those goals is the agent's most likely destination. Conversely, deceptive path-planning takes an initial location, a destination and a set of possible goals and finds a path such that a goal recognition system (i.e., an interested observer) is *unable* to determine the destination of an agent travelling along that path.

These are simplified versions of equivalent problems studied in the setting of general task-planning. At a technical level, we adapt Ramirez and Geffner's seminal framework for goal recognition (2009, 2010) to the more restrictive path-planning environment within which we develop specialised solutions that minimise computational effort. We analyse and improve on the probability distribution formulas used in support of two contemporary cost-based goal recognition frameworks (Ramirez & Geffner, 2010; Vered, Kaminka, & Bi-

ham, 2016) to better deal with agents whose behaviour is—or appears to be—suboptimal. Finally, we invert our probabilistic goal recognition solution as a basis for a deceptive path-planning model, which also incorporates concepts from the general theory of deception developed by Bell and Whaley (Bell, 2003; Bowyer, 1982; Whaley, 1982).

Our specific contributions are the provision of faster approaches to computing an agent's most likely goal, both online and offline, and a theory of deception that facilitates a principled approach to understanding and using deceptive path-planning. This work may be used not only in obvious contexts such as surveillance, privacy-protection and warfare ("I want to know where your tanks are going but I don't want you to know where my tanks are going!") but also for applications such as games programming, intelligent transport systems and human-machine interaction (so-called 'human-in-the-loop' processing), whether that involves cooperative teamwork or adversarial reasoning (or both).

Essentially, we seek answers to the following questions.

1. What gains can be made, computationally, when adapting state-of-the-art plan recognition techniques to the special case of goal recognition in path-planning?

2. How can we improve on the response of goal recognition systems when the observed agent's behaviour is (apparently) irrational?

3. How can we define and generate measurably deceptive paths?

1.1 Our Approach

The dominant approach to goal recognition involves observing an agent's behaviour, then matching her actions against a sequence of similar actions stored as a plan in a plan library; the 'goal' of the plan is assumed to be the goal of the agent (Demolombe & Hamon, 2002). Development and maintenance of a plan library is time-consuming (plans must usually be hand-crafted by a domain expert or converted from one format to another to facilitate comparison) and the libraries are domain-specific (Sukthankar, Geib, Bui, Pynadath, & Goldman, 2014). To overcome some of these problems, a recent trend has emerged, thanks largely to the innovative work of Ramirez and Geffner (2009; 2010), whereby, instead of creating and storing plans in advance, they are generated for comparison on-the-fly using automated planning. Ramirez and Geffner's model builds on the insight, independently arrived at by others (Baker, Saxe, & Tenenbaum, 2009; Pattison & Long, 2010), that an agent's most likely goal can be determined by reference to the principle of rational action, that is, by considering how closely the agent's observed behaviour conforms to an optimal plan for each possible goal.

Operationally, Ramirez and Geffner's approach (2010) involves computing the cost of the best plan that differs from observed behaviour. The calculation is cumbersome and several authors (e.g., Escudero-Martin, Rodriguez-Moreno, & Smith, 2015; Vered et al.,

2016) have substituted a more economical alternative but without proof of its equivalence. We supply this proof; we also define (and demonstrate the impact of) the corner case where it does not apply (pp.50, 53, 80).

Many other modifications and extensions have been considered in relation to Ramirez and Geffner's original framework, often using navigational domains as part of a simplifying assumption to assist explanations (Escudero-Martin et al., 2015; Keren, Gal, & Karpas, 2014; Sohrabi, Riabov, & Udrea, 2016; Vered et al., 2016, etc.). In our work, we focus on those domains as an end in themselves, exploring how we can build on existing scholarship under the particular constraints of path-planning to generate new, efficient solutions. In most cases, goal recognition solutions are geared to handling observations in either an 'online' or an 'offline' process. The offline process considers a full set or sequence of observations in one batch, whereas the online process handles them incrementally, often in real-time as they occur. The solutions we present in Part I—'single-observation recognition' (p.42) and the 'radius of maximum probability' (p.61)—do not depend on the observation sequence so manage to achieve the advantages of an online solution without the computational cost of accumulating and incrementally processing observations.

Traditionally, goal recognition can be categorised into three types: 'keyhole' recognition, whereby the observed agent behaves just as if she were not being watched; 'intended' recognition, in which she actively attempts to reveal her goal; and 'deceptive' or adversarial recognition, where the agent deliberately misleads or obfuscates. Until recently, adversarial recognition has been somewhat neglected in the literature (Carberry, 2001). The work that has been done has tended to focus either on domain-specific deceptivity or anomaly detection (Kott & McEneaney, 2007). This omission is exacerbated by the fact that goal recognition systems (particularly cost-based systems of the type outlined above) are typically predicated on an assumption of rationality, which carries with it an in-built assumption of honesty (since there is little difference, computationally, between an agent that is irrational and one that is rational but deceptive).

Importantly, a goal recognition system rarely knows in advance which type of recognition it is dealing with. This means that a system which assumes it is dealing with keyhole or intended recognition may actually be dealing, unknowingly, with a deceptive agent. In this thesis, we present analysis (p.104) of probability distribution formulas used in support of two state-of-the-art goal recognition frameworks (that of Ramirez and Geffner (2010) and Vered et al. (2016), both of which assume rationality and keyhole recognition) and find that, faced with suboptimality, both formulas return anomalous solutions. Nevertheless, our work builds on the insights underlying those formulas to arrive at an alternative model capable of handling suboptimality in a principled way (p.110).

The literature relating to deception largely focuses on its prevention. This is probably partly due to the ethical considerations which often arise at the mere mention of "deception" (Carson, 2010). Concerns have lately become increasingly pronounced with

anxieties about privacy (Keren, Gal, & Karpas, 2016), debate about how and when it is 'OK' for robotic carers to deceive their vulnerable human charges (Arkin, Ulam, & Wagner, 2012), and so on. We are also now seeing the emergence of adversarial elements, not only in predictable spheres such as network security but, more surprisingly, on the roads in our (perhaps imminent) driverless cars (McClean, Stull, Farrar, & Mascareñas, 2013) and even in the context of machine learning where facial recognition systems can be deceived into thinking that a photograph of one person is a photograph of somebody else entirely (Sharif, Bhagavatula, Reiter, & Bauer, 2016). Elsewhere, game-theoretic accounts dominate (Kott & McEneaney, 2007), in which, as warned by Castelfranchi (2000), deception arises almost incidentally as a preferred strategy for maximising value (e.g., Hespanha, Ateskan, Kizilocak, et al., 2000).

Notwithstanding the body of work on deception, we have been unable to identify a systematic computational approach that is generally applicable to core navigational domains. In Part II of this thesis, we present a model for deceptive path-planning which tackles deception as an active pursuit (p.116). In doing so, we have reached outside computer science to the work of military strategists, Bell and Whaley, who under a pseudonym produced a general theory of deception or "Cheating" (Bowyer, 1982). From them we take our terminology and basic strategies—in particular 'simulation' (showing the false) and 'dissimulation' (hiding the real)—which we reinterpret in a path-planning framework.

Practically, we treat deception as an inversion of probabilistic goal recognition, used as a black box to stand in for the observer that we want to deceive. Thus, our model is agnostic with respect to any particular goal recognition system. In general, however, recognising that ambiguity is a problem for goal recognition, we implicitly acknowledge how useful it is as a tool for deception. Furthermore, the use of probabilities is, in itself, significant. In goal recognition, a probability distribution allows solutions to be ranked. Clearly, if goals can be ranked for likelihood, they can similarly be ranked for *un*likelihood; if we know how unlikely the observer believes each goal to be, we know how unlikely they believe the real goal to be. Thus, probabilistic goal recognition provides us with a principled mechanism for measuring not only how likely a goal is, based on the observed plan, but also that observed plan's potential to deceive.

Goal recognition and deception are mirror images of one another. They are challenging topics, no doubt, acknowledged markers in the development of intelligent human behaviour (e.g., Alloway, McCallum, Alloway, & Hoicka, 2015). By considering them in the context of path-planning—one of the longest-studied and best-understood problems in computer science (Russell & Norvig, 2013, p.109)—this thesis aims not only to develop specialised solutions but also to advance our understanding of the topics themselves.

1.2 General Assumptions and Recurring Themes

This thesis returns to certain issues repeatedly and makes general assumptions in the following key areas.

1.2.1 Theory of Mind

Our work involves reasoning about an agent's intent; necessarily, it makes the strong preliminary assumption that she has one. Here, we depend on the 'theory' (because how can we know?) that similar processing is going on in other people's minds to that which seems to be going on inside our own; that is, that they too are forming intentions, making plans, arriving at beliefs including, quite possibly, false beliefs.

It is an assumption made by people every day: so-called 'folk' or 'common sense' psychology which suggests our actions are the result of logical reasoning and rational thought. It little matters that behavioural economists have argued long—and lately popularly (Kahneman, 2011)—that this is only a partial explanation for how people make decisions. The notion has the great advantage of being efficiently computable. It is at the root of the belief-desire-intent (BDI) model of agent programming, built on the work of Dennett (1987), Bratman, Israel, and Pollack (1988) and Rao and Georgeff (1991) and, as computer scientists, we persist with it pragmatically because it works.[1] Furthermore, the approach promises to become increasingly accurate over time: BDI is a dominant paradigm for AI agents; more and more agents are incorporated into human-AI teams; goal recognition systems observe team behaviour—and increasingly that behaviour is being decided (thanks to the growing number of agents in the team) on a BDI model.

The assumption of intentionality is important: it provides a 'goal-recogniser' with something to measure against and a deceiver with something to disguise: "How closely is the agent following the plan for this goal? Very closely? Then she is likely going there! Not very closely? Then we can forget about that goal and consider something else."

1.2.2 Rationality

> *If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action.* (Newell, 1982, p.102)

Under Newell's definition, path-planning is a clear expression of rational action. It is an assumption on which contemporary cost-based goal recognition relies; though also one that can make it vulnerable to deception.

[1] Electricians take a similarly pragmatic approach to electrical circuits. Electricity does not flow clockwise but it is convenient to 'believe' that it does because the assumption enables us to calculate (correctly) how much current will end up at each node. Similarly, the assumption that people reason about their intentions as a prelude to taking action, helps us to predict (often correctly) what they are likely to do.

The implication is that a rational agent will take the most direct approach and, since in path-planning accessibility is measured by cost, rationality can be taken as just another way of saying 'cost-sensitivity'. This is not to suggest that an agent might not prefer paths that are aesthetically pleasing or routed, for example, via service stations, nor that the agent might not have altruistic reasons for taking one path instead of another; but, if necessary, these considerations can all be incorporated into an agent-specific—or agent-*type*-specific—cost model. Certainly, if the agent in question (whether doing the observing or being observed) is computerised (as opposed to human), the assumption that it is acting with reference to some sort of cost model is not unreasonable.

Although we are aware of alternative approaches that better match the sort of bounded rationality found in resource-constrained agents such as humans (e.g., Halpern, 1998), we have not attempted to incorporate that reasoning into our account. We assume that agent behaviour may be slightly suboptimal but that a rational agent's intention is to behave more or less optimally. Thus, except where otherwise noted, we use the term 'rationality' synonymously with 'cost-sensitivity' and when we say that a path is 'fully rational' we mean that it is optimal in terms of cost.

An agent's behaviour may be both rational *and* suboptimal, as explored in Part II.

That said, it may sometimes *appear* that rationality and optimality are at odds. This is because there is more than one reason for an agent to engage in suboptimal behaviour: she (or it) may actually be irrational (e.g., drunk or broken), in which case there is no discrepancy; but it may also be that she is operating under a different cost model from the one we expected (i.e., if we knew the correct cost model, we would see that she is behaving cost-efficiently after all); or her behaviour may be deliberately deceptive (a deceptive agent may rationally choose an apparently suboptimal course of action precisely because the more cost-efficient route would lead to detection: behaviour that is simultaneously suboptimal *and* cost-sensitive, the situation we explore in Part II of this thesis).

1.2.3 About the Domain

By path-planning, we do not mean 'motion-planning'. Except where otherwise stated, this thesis focuses on core navigational domains of the type used for route-planning, such as graphs (in the discrete domain) or Euclidean space (in the continuous domain), which define only location and cost (often synonymous with distance). That is, our domain of interest supplies the minimum amount of information necessary for us to be able to solve a shortest path (or cheapest path) problem, which, in turn—as discussed in Section 1.2.2 above—is a minimum requirement for the assessment of rationality on which we depend.

Space in the real world is unlike any other dimension. We experience it directly and, as a result, we understand it well. It has very particular constraints. For example, the only available actions are movements; and movement can only occur between neighbouring

locations. Actions are deterministic (if I move to A, I am sure to be at A, and would be very surprised to find myself, instead, at B) and observations are complete in that they capture the full state (if you know that I am at A, there is nothing more to know). This is different even from other domains that might be used for motion-planning or trajectory prediction. It is a 'bare bones' or 'core' account; it excludes richer notions such as pose or heading, velocity, acceleration, clearances, fuel reserves, localisation and so on.

It may, of course, be the case that other areas of planning, or other particular planning problems, share the above constraints. Indeed, we discuss the relationship between our goal recognition model for path-planning and adaptations that would make it suitable for general task-planning in Section 3.4.2 (p.84). However, we take the view: (a) that core navigational domains in themselves present an abundance of worthwhile applications for goal recognition and deception; and (b) that reducing the problem to its simplest form allows us to focus on the main issues first, which leads to improved understanding of the key issues and subtleties. We leave elaborations on the core problem to be studied as future work.

In any event, the domain is not necessarily as restrictive as it at first appears.

> *The spatial domain [is] particularly suitable as a medium for conveying knowledge, since its properties are universal to different cognitive systems. Thus, the spatial domain can be used particularly well as the source domain for metaphors with a non-perceivable or abstract target domain. In this way, the properties of physical space can be used as a vehicle for conveying non-spatial concepts.*
>
> (Freska, 1991, p.362)

One objective of this thesis is to use path-planning to explore the concepts of goal recognition and deception in precisely this way.

1.3 Outline of Contributions

Our main contributions are as follows.

1. **Single-observation goal recognition for path-planning domains**. Assuming that the agent's starting point is known (because all entry points are monitored or because it is the only entry point into the domain), we adapt cost-based probabilistic techniques designed for task-planning and demonstrate an alternative formula for use in the path-planning domain which is observation-independent. That is, it enables an observer to determine where an agent is going without knowing where she has been (p.55). Previous best practice in this space would require the observed agent's prior path to be monitored as comprehensively as possible. This solution not only saves time and computational effort; it makes feasible the pre-computation of a sort of 'heatmap' of probabilities for each starting point in a given domain (p.57) from

which, given just one observation—her current location—the agent's most likely destination can be retrieved in constant time.

2. **Radius of Maximum Probability.** Based on the single-observation formula, we demonstrate calculation of a cost-radius within which any particular goal is the most probable without actually calculating any probabilities; all that is needed are the optimal costs from the starting location to each goal and between goals (p.61). In practical terms, calculation of the radius of maximum probability contributes to the emerging field of goal recognition design (i.e., the problem of designing an environment in which goal recognition is easy to perform) in that it can be used to identify optimal surveillance points: an agent observed within the radius of maximum probability for a protected goal is most likely heading for that goal. Alternatively, being quick to calculate, it can also be used on-the-fly for spot checks or in dynamic domains (e.g., Raffe, Zambetta, & Li, 2012), where precalculation cannot be employed.

3. **A self-modulating probability distribution formula for goal recognition.** Contemporary cost-based goal recognition models make an assumption of rationality. Our analysis of two state-of-the-art systems (Ramirez & Geffner, 2010; Vered et al., 2016) faced with behaviour that appears to be irrational (but which may occur when an agent's behaviour is fully rational but *deceptively* suboptimal) exposes anomalies and potential vulnerabilities (p.104). We develop a self-modulating formula which lifts the rationality assumption (p.110). It recognises the appearance of irrationality and reduces its level of confidence accordingly, returning meaningful results whether the agent is rational, irrational or deceptively suboptimal.

4. **A model of deception as path-planning.** We present a model that defines deceptive path-planning in terms of deceptive magnitude, density and extent (p.116) and introduces the notion of a 'last deceptive point' (p.121), which can be used to develop economical strategies for solving the deceptive path-planning problem. Drawing on insights from goal recognition, game theory and disciplines outside computer science such as military strategy and psychology, this model concretises deception—a notoriously elusive, abstract concept—in the context of path-planning: one of the best understood and most widely studied topics in AI.

5. **Proof of equivalence between Ramirez and Geffner's cost difference formula and a more economical alternative**. Ramirez and Geffner's probabilistic seminal framework (2010) for goal recognition in task-planning involves computing the cost of the best plan that 'differs from observed behaviour'. Several authors have suggested that it can be simplified by substituting instead the cost of the best plan per se (e.g., Escudero-Martin et al., 2015; Kaminka, Vered, & Agmon, 2018; Sohrabi et al., 2016; Vered et al., 2016). We provide a sound theoretical basis for

this substitution by demonstrating the precise cases where the simplified formula and Ramirez and Geffner's original formula return identical results (p.50), where they differ (p.53) and the potential impact of those differences (p.80).

1.4 Thesis Outline

The rest of the thesis is organised as follows.

- In Chapter Two, we provide a literature review which sets out the background to this thesis. First, we review the landscape with respect to goal recognition and deception in path-planning; we then focus on those works which are central to our approach.
- The thesis then divides into two parts.
  - Part I (Chapter Three) deals with goal recognition in the context of path-planning. Here, we demonstrate the computational gains that can be made when applying cost-based goal recognition in path-planning domains using our novel techniques of single-observation recognition and the radius of maximum probability.
  - Part II (Chapter Four) deals with deception. It looks at improvements that can be made when dealing with apparently irrational (potentially deceptive) agents and at ways to actively develop measurably deceptive paths. In this chapter, we present our self-modulating probability distribution formula and describe a model of deception for path-planning.
- In Chapter Five, we present our conclusions. We acknowledge the limitations of the work presented in this thesis and propose avenues for future research.

CHAPTER 2

Literature Review

"The ability to observe, and the ability to see the little things that seem trivial at first, may become amazingly important and meaningful. Out of little observations huge ideas may grow; and if a mind, made receptive by training in the use of the senses, can store away a mass of observations, the time will come when the whole collection can be unrolled, connected together as a great novel is planned, in a compelling pattern that tells us something new*."*

–Harold Gatty

This thesis stands at the intersection of goal recognition, deception and path-planning, topics with long histories both within and outside computer science. This chapter does not (and could not) aim to provide a full survey of all these fields. Rather, it lays out the general landscape then focuses on those works which we believe best contextualise the models of goal recognition and deception that we propose in Chapters 3 and 4.

2.1 Path-Planning

Path-planning is a sub-problem of general task-planning: that branch of artificial intelligence, central to decision-making, which involves "devising a plan of action to achieve one's goals" (Russell & Norvig, 2013, p.366). In planning, goals are conceived as states, that is, combinations of atoms or 'fluents', capable of representing the many possible situations (appropriate to the problem domain) in which an agent might find herself. A planning problem asks: if I am in this situation and would rather be in that situation, what sequence of actions can I take to make it so?

Path-planning, by contrast, is the problem of finding a path from a starting point to a goal through the map or model of a domain. In path-planning, states are reduced

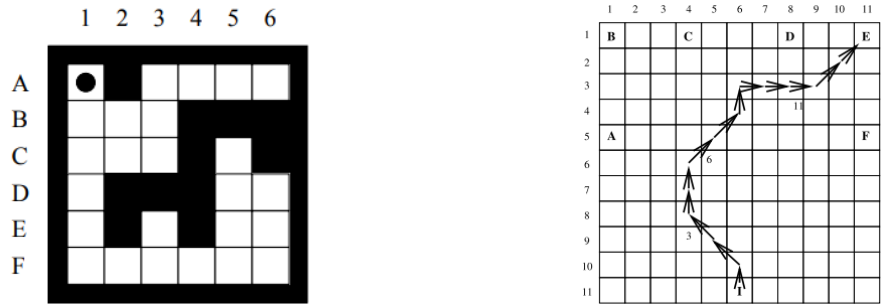


Figure 2.1: Gridworlds from (Tovey & Koenig, 2000) and (Ramirez & Geffner, 2010).†

†Externally-sourced images in this chapter are used in accordance with fair dealing for criticism and review. Copyright remains with the owners.

to locations, actions to movements, and plans to routes from one location to another. In classical path-planning, the map or model—whether it represents the real-world, a game world or other kind of search space—is typically abstracted into a graph or grid (Harabor & Grastien, 2012; Hart, Nilsson, & Raphael, 1968; LaValle, 2006). The question now becomes, what path should I follow to reach my destination?

Path-planning represents a significant simplification of the task-planning problem and—in much the way that people commonly reach for pen and paper to illuminate problems by quickly sketching a flow-chart or a brain-tree—solutions to general task-planning problems are very often evaluated first as *toy* problems in a 'gridworld' domain (e.g., Tovey & Koenig, 2000). This two-dimensional grid (similar to those shown at Figure 2.1), which may be as small as six- or eight-by-eight cells and in which costs are equated with distance, exemplifies the KISS (Keep It Simple, Stupid) approach to problem-solving. By eliminating potentially confounding variables and extraneous noise, the decision points in a computational process are thrown into relief, errors or inconsistencies become visible and the whole process is made more transparent.

The Shortest Path Problem

In a graph-based domain, $\langle N, E \rangle$, where N is a set of nodes and $E \subseteq N \times N$ is a set of edges, the general approach to path-planning is to begin from a nominated start node $n \in N$, expand it (that is, obtain a set of all its neighbouring nodes) and then proceed to expand the newly discovered nodes in some systematic way, repeating the process until a goal is reached. A 'best-first' search algorithm is one that applies an evaluation function to the discovered nodes to decide which of them should be expanded next. Edges between nodes are frequently weighted (according to cost or distance) and, instead of finding any path, the problem becomes one of finding an optimal (i.e., minimum weight, cost or distance) path to goal (Russell & Norvig, 2013).

The shortest path problem has a long history. A comprehensive review of algorithmic solutions appears in (Cherkassky, Goldberg, & Radzik, 1996). Djikstra's algorithm (Dijkstra, 1959), which evaluates nodes based on their minimum cost so far, is still widely

Listing 2.1: A* algorithm.

```
1  Require: starting node, s; goal
2  Returns: path or failure
3
4  openlist ← ⟨s, f(s), null⟩
5  while openlist is non-empty
6     n ← node on openlist with lowest f(node)
7     if n is the goal then
8        constructpath(n,parent)
9        return path
10    else
11       closedlist ← n
12       neighbours ← getNeighbours(n)
13       for neighbour in neighbours
14          if neighbour not on closedlist then
15             openlist ← ⟨neighbour, f(neighbour), n⟩
16          else
17             if f(neighbour) < f(neighbour − on − closedlist)
18                remove neighbour from closedlist
19                openlist ← f(neighbour)
20 return failure
```

used as the basis for route-planning algorithms across road networks (e.g., Bast et al., 2015). Even in continuous domains, Djikstra and its variants are commonly used because, in practical applications, the space must usually be discretised to a grid or a graph to make the problem tractable (LaValle, 2006).

The A* algorithm with which path-finding in games is said to be "synonymous" (Millington & Funge, 2009, pg.215) is one of Djikstra's variants (see Listing 2.1). A* considers not only the cost so far, but also an heuristic, which estimates the cost from the current node to the goal, enabling a more directed search using the well-known function $f(n) = g(n) + h(n)$, where $n \in N$ is the node being evaluated, $g(n)$ is the minimum known cost from the start node to n, and $h(n)$ the estimated optimal cost from n to a goal (Hart et al., 1968). Provided that the heuristic $h(n)$ is consistent[1] and not an over-estimate, A* is guaranteed to find an optimal path.

Jump Point Search (Harabor & Grastien, 2012) is also guaranteed to be optimal. It has recently become a dominant extension to A* for searching uniform-cost grids, where the symmetry of that particular environment can be exploited to massively reduce the number of nodes that need to be evaluated.

No matter how the search is conducted, the conventional path-planning problem is typically constrained by a requirement to return not just *any* path but one that is shortest, cheapest, fastest, or otherwise maximises some value or minimises some cost. To our knowledge, however, it has not previously been used to maximise deceptivity.

[1] A consistent heuristic is one that accommodates triangle inequality, that is, the sum of any two sides of a triangle must always be greater than or equal to the length of the third side or, put another way: the shortest path between two points is a straight line; breaking that line cannot make the path shorter.

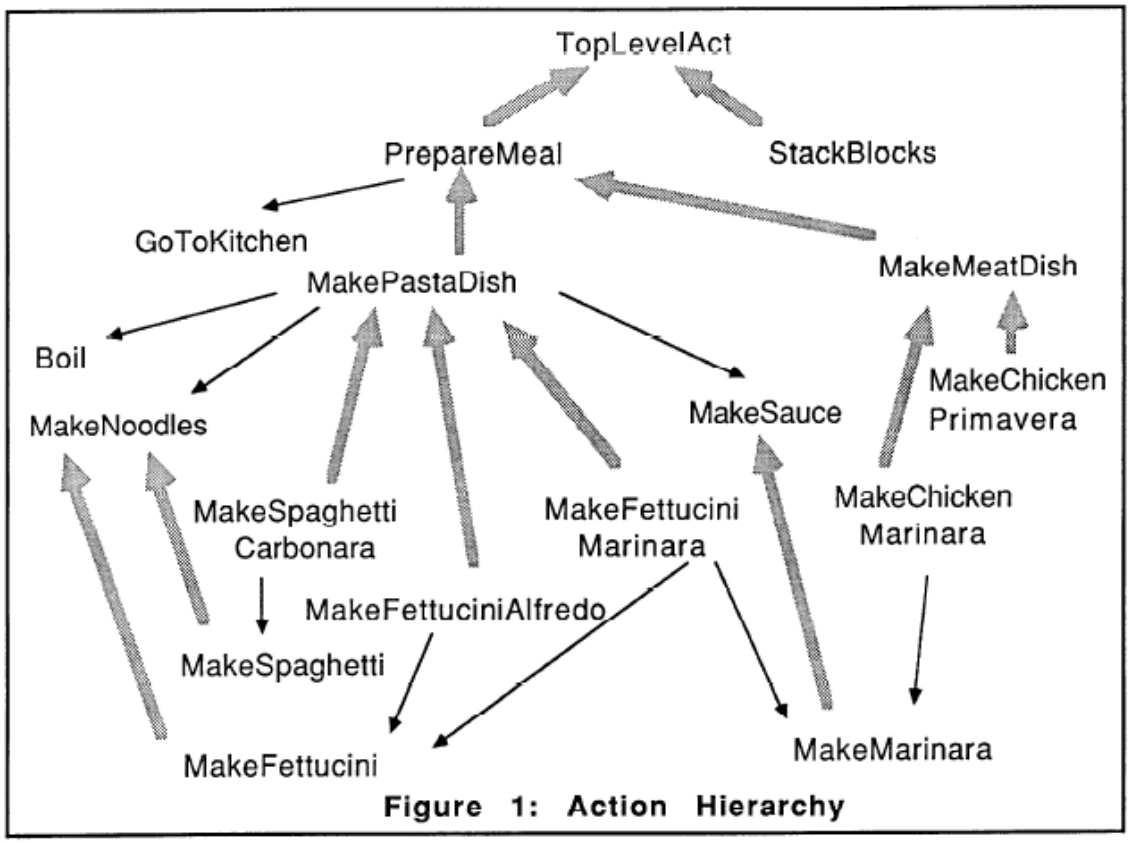


Figure 2.2: A traditional plan, as expressed by Kautz and Allen (1986, p.34).

2.2 Goal Recognition

Goal recognition is the problem of determining an agent's intent by observing her behaviour. It falls within the scope of plan, activity and intent recognition (PAIR) and concerns that aspect of the PAIR problem which is interested in the final or 'top level' goal, rather than the plan or subgoals that must be negotiated on the way to achieving it (Blaylock & Allen, 2006).

PAIR has a long history in computer science (Carberry, 2001) and its growing number of navigational and motion-related applications range from adversarial reasoning for games and the military (Kott & McEneaney, 2007) to the monitoring of residents in smart homes for the cognitively impaired (Roy, Bouzouane, Giroux, & Bouchard, 2011). With the advent of autonomous robotic vehicles, such uses have become increasingly important as an adjunct to trajectory prediction (e.g., Wiest, Höffken, Kreßel, & Dietmayer, 2012), to recognise driving goals such as lane-changing (Firl & Tran, 2011; Graf, Deusch, Seeliger, Fritzsche, & Dietmayer, 2014) and more broadly to predict the behaviour of detected objects, such as other vehicles, cyclists and pedestrians (Kooij, Schneider, Flohr, & Gavrila, 2014).

The first formal articulation of the plan recognition problem is widely attributed to

Kautz and Allen (1986), whose approach is illustrated in Figure 2.2, where up-arrows indicate task-specialisation and down-arrows indicate task-decomposition. Their solution, using minimum set cover over a task network representation, did not accommodate uncertainty and the desirability of a probabilistic solution quickly gave rise to alternative models based on Dempster-Schaffer theory (Bauer, 1995; Carberry, 1990) and Bayesian Theory (Charniak & Goldman, 1991). Numerous other probabilistic solutions have followed, ranging from probabilistic grammers (Geib & Goldman, 2009) to POMDPs (Baker, Saxe, & Tenenbaum, 2011; Pynadath & Marsella, 2005; Ramırez & Geffner, 2011), amongst others (e.g., Bui, 2003; Mirsky & Gal, 2016). Sukthankar et al. (2014) provide a recent survey of contemporary approaches.

The traditional—and still prevalent—technique involves matching a sequence of observations to a sequence of actions stored as a plan in a *plan library* (Sukthankar et al., 2014). The winning plan is the one that best matches the observations and, given that each plan sets out to achieve some goal, having identified the plan, the observer has implicitly identified the goal (Demolombe & Hamon, 2002). However, the difficulty of having to either acquire or hand-code the plans makes it desirable to perform plan recognition without them (Hong, 2001), particularly in real-world navigational domains such as a sea or a city, where the number of plans whereby an agent might move from point A to point B is potentially infinite (Pattison & Long, 2013).

2.2.1 Plan Recognition as Planning

Rather than endure the overhead of generating, storing and searching through numerous plans that might turn out to be redundant, Ramirez and Geffner (2009) proposed the use of a classical planner to generate plans, as needed, relative to a 'domain theory' (i.e., a planning domain). This innovation makes it possible for plan recognition to leverage advances made by the planning community. It relies on a key insight that the probability of a plan can be linked to its cost. Appealing to the principle of rational action, an agent is assumed to be taking the optimal (for which read minimum cost/maximum utility) or *least sub-optimal* (Ramirez & Geffner, 2010) path to goal. Ramirez and Geffner's 2009 and 2010 papers are central to this thesis and we discuss them more fully in Section 2.5.

While Ramirez and Geffner were among the first to recognise the potential of making plan recognition a planning problem, they were not alone. Cognitive scientists Baker et al. (2009) independently arrived at a similar conclusion. They characterise their methodology as 'Bayesian inverse planning' and equate it with theory of mind. They are interested in modelling human cognition and their experiments compare calculated predictions (again based on the principle of rational action) with human inference as expressed by participants in their studies. Their findings show that, initially, predictive calculations accurately model the human inference process. However, once a goal has been identified as the most probable, human observers are reluctant to relinquish it. That is, they continue

to subscribe to an initially assumed intention even in the face of subsequent contradictory observations and seem to seek any plausible reason to justify anomalous behaviour (Baker et al., 2011).[2] This implies that, once a human observer has been persuaded of a false goal, there may be a period of grace in which to pursue the *real* goal while that observer 'readjusts her thinking'. In devising deceptive paths, a strategy that aims to be as deceptive as possible for *as long as possible* implicitly seeks to benefit from this effect.

2.2.2 Extensions and Alternatives

In most applications, goal recognition tasks must be performed 'online' (Blaylock & Allen, 2006). That is, the objective is to identify an agent's goal *while the plan is being carried out* (certainly before it terminates and anyway as quickly as possible), working from observations that are delivered incrementally. It is a purpose not explicitly considered by Ramirez and Geffner (2010) and one for which more efficient models have been sought.

Concurrently with work undertaken preparatory to this thesis, Vered and Kaminka have developed a body of work, geared particularly towards online recognition in a field closely related to ours: that of motion-planning in continuous domains (Vered & Kaminka, 2017; Vered et al., 2016; Vered, Pereira, Magnaguagno, Kaminka, & Meneguzzi, 2018). Vered et al.'s 2016 formulation of cost-based goal recognition solves the potential problem of identifying continuous observations—which may consist of points (for a stationary agent) or trajectories (for an agent in motion) by defining both (or either) as a function of the time during which each observation is made. Candidate paths are conceived as similarly time-segmented subpaths concatenated together, effectively discretised, which allows for direct comparison between the path segments and the observations.

Kaminka et al. (2018) present a somewhat different model, which extends to task-planning and is unified to cover both discrete and continuous environments. They achieve this by treating continuous paths as a sequence of continuous subpaths between waypoints. Whereas transitions in a discrete task-planning environment are effected by applying an action a in a particular state s (and so reaching some new state s'), here instead the equivalent action is conceived as resulting in a continuous transformation from s to s'. The granularity of these transformations, captured as a sequence of indices $I = \{1, \ldots, m\}$, is assumed to be determined at runtime and, although I is potentially infinite in the continuous case, it is easy to see that in the special case where it is binary (i.e., where $I = \{0, 1\}$), the model applies equally to discrete transitions. The authors point out that the continuous environment has demonstrable advantages when dealing with goal recognition that corresponds to real-world path-planning. This is because, although a continuous environment can always be discretised to an appropriate level of granularity

[2] Famously, in the case of the D-Day landings, the German high command was convinced that the allied landings would occur in Calais. Even after troops had been observed actually landing in Normandy, the German 15th army remained stationed at Calais: commanders were seemingly unable to relinquish the assumed intention of a Calais landing, preferring to explain the conflicting evidence as a decoy operation.

to accommodate known locations, once the discretisation has been decided, newly added locations may become indistinguishable. For example, in an environment discretised as a grid, observations at two different locations can end up inside the same cell so that, computationally, they appear to be at the *same* location.

Vered and Kaminka consider goal recognition in relation to types of motion-planning beyond route-planning, such as drawing analysis. They characterise goal recognition as 'goal mirroring' (that is, the empathetic human response to observations, whereby we imagine ourselves in the observed situation and assume the observed agent is behaving as we would) and have an interest in uncovering the 'heuristic' (i.e., probability distribution) that best corresponds to human reasoning. Thus, although they build on the formula at the centre of Ramirez and Geffner's non-probabilistic model (2009)—which (effectively) *subtracts* the optimal cost of reaching a goal from the optimal cost of reaching that same goal given the observations that have already been made—the formula they use to derive a probability distribution takes, instead, the *ratio* of those two terms: an heuristic known to be a good match with human goal reasoning (Bonchek-Dokow & Kaminka, 2014).

In focusing on the mechanics of online goal recognition, Vered and Kaminka (2017) save time by re-using the calculated cost of the 'path prefix' (that is, the observed path so far) rather than repeatedly calculating its entire cost.[3] The authors propose two further mechanisms to help reduce computational costs. In online goal recognition, observations are processed incrementally and Vered and Kaminka's model synthesises the path prefix with the path suffix at every step. On receiving a new observation, the system first checks whether the agent appears still to be approaching the same goal that was previously deemed 'most probable'. If so, recalculation is skipped. If not, the system checks the agent's trajectory (i.e., angle of movement) since the previous observation. If the agent appears to have 'turned away' from a goal by an angle greater than some given threshold, that goal is pruned from the candidate set making future probability calculations (across one less goal with each pruning) faster and faster. The claims relate to task-planning (albeit in a continuous domain) but the examples relate to motion-planning and it is unclear how the agent's 'trajectory' would be assessed in alternative domains.

Vered et al. (2018) combine goal mirroring with the use of landmarks. Classical planning can be computationally expensive in itself and Pereira, Oren, and Meneguzzi (2017) have extended Hoffmann, Porteous, and Sebastia (2004)'s work on landmarks (i.e., actions that *must* occur for a goal to be achieved) so avoiding both the cost of planning and the expense of a plan library. Pereira et al.'s model treats landmarks as facts that must be satisfied (rather than actions that must occur). Briefly, an ordered list of landmarks for each goal is first extracted from the domain; then, by comparing observations with this set, impossible goals can be pruned so that only the achievable goals remain. Vered et

[3] In fact, using Ramirez and Geffner's probability distribution formula instead of Vered and Kaminka's ratio-based alternative, it is unnecessary to calculate the cost of the path-prefix at all (see Section 3.1, p.42).

al. (2018) extend this approach to provide an online solution for discrete *and* continuous domains that generates a probability distribution by taking the ratio of 'landmarks achieved' against 'total landmarks' for each goal. In order to apply the notion of landmarks in a continuous domain, the authors characterise them as regions around a goal, some part of which must be traversed in order to reach that goal. They denote these regions as (rectagonal) bounding boxes based on visibility (i.e., absence of obstacles) between enclosed points and the goal. Although a 'blocks world' example is discussed in this work, experimental evaluation is confined to navigational domains.

Escudero-Martin et al. (2015) also avoid the expense of planning, in their case by generating plan graphs based on the domain; and they save time by estimating costs rather than calculating them precisely. As discussed more extensively in the final section of this literature review (p.35), Ramirez and Geffner (2010) introduce the notion of 'cost difference', a formula that compares the optimal cost of a plan that incorporates observed actions with the optimal cost of one that does not. Escudero-Martin et al.'s probabilistic solution to goal recognition involves first deriving a plan graph from the problem domain, from which to generate cost estimates for each goal. The graph is then pruned based on observed actions and a second set of cost estimates is generated so arriving at an *estimated* cost difference, which can be plugged into Ramirez and Geffner's probability distribution formula directly. The resulting distribution can be used as-is or as an intermediate step after which only the 'most likely' goals are considered further under some other process.

Ramirez and Geffner's solution (2010) does not natively accommodate 'unreliable' observations and Sohrabi et al. (2016) extend their work by introducing weights to allow for the possibility that observations may be noisy or missing. They demonstrate that, by taking the average of multiple results from a top-k planner (i.e., one that generates multiple high quality plans), it is possible to produce a more reliable result more efficiently than can be achieved by other means. In this paper, the authors also point out that, although a plan in classical task-planning is given as a sequence of *actions*, in practice, the actions themselves are rarely visible. They suggest, therefore, that observations may be better understood as the *effects* of actions. This approach, whereby observations are treated as 'observable fluents' (in their case, mapped into states in a plan's execution trace rather than to actions in the plan itself) is a good fit with discrete path-planning domains where it is convenient to define a path as a sequence of locations (e.g., the nodes in a graph rather than the edges that connect them).

Freedman and Zilberstein (2017) extend Ramirez and Geffner's framework into the realm of human machine interaction (HMI). Their work uses the solution from a plan recognition problem ("what is that agent trying to achieve?") as the goal (or partial goal) input into a planning problem. Effectively this provides an agent with seeming 'foresight' as to the intentions of the observed agent, which can then be used either to help or hinder. Whether assisting a colleague or blocking an adversary, it is useful to identify an

optimum point for intervention. In a subsequent paper, Freedman, Fung, Ganchin, and Zilberstein (2018) save time with a solution that ranks multiple goals at once, rather than carrying out a computation for each goal separately. As the authors point out, the many extensions to Ramirez and Geffner's framework share a common objective in attempting to achieve goal recognition more quickly.

2.2.3 Goal Recognition in a Navigational Context

We conclude this section by noting that goal recognition is already used with great effect in explicitly navigational scenarios, particularly in the emerging area of ubiquitous computing. Typically, such scenarios involve additional features, beyond the core path-planning domain which is the focus of our work.

For example, in their work on elastic pathing, Gao et al. (2014) show that speed alone (monitored telematically) can be used to identify an agent's path, and thereby her goal. The only constraints are that the agent should be operating in a domain (such as the road network) where additional available information includes her likely start location (e.g., her home address), maximum legal speeds along each path segment and the location of intersections (where it may be necessary to stop or slow down).

Tastan and Sukthankar (2011) tackle the problem of predicting human movement through an indoor environment, where the presence of obstacles and a human tendency to take different routes to and from a goal make behaviour difficult to assess on the basis of cost. Their model assumes instead a human trajectory based on heading, mediated by a preference for moving in straight lines, attraction towards goal and object avoidance. In tests (both online and in a crowded real-world office environment) the subtleties achieved using this kind of vision-based steering were found to be far superior to those obtained using traditional 'shortcut' (i.e., cost-based) prediction.

In the realm of ubiquitous computing, the GPS function in mobile phones has been used to predict an agent's future location by identifying changes in her mode of transport (Patterson, Liao, Fox, & Kautz, 2003). Meanwhile, recent innovations include dissemination of the so-called "Sentry" app for mobile phones, whereby civilians in Syria share observations about aircraft travelling overhead (specifically time, location and direction of travel) to assist in the prediction of impending bomb targets (Dadouch, 2018). More mundane but useful nevertheless, an agent's plans can be predicted, even before she acts, from browsing habits (i.e., without monitoring any of the usual physical considerations, such as speed and direction). Indeed, researchers have demonstrated that they can map an agent's information needs in a virtual network to her purchasing needs in a real-world shopping centre (Ren et al., 2017).

Goal recognition for the core path-planning domain does not need to be seen as an alternative to these approaches but as orthogonal to them; together they can provide a richer model.

2.3 Adversarial Recognition

In her comprehensive survey of the field, Carberry (2001) categorised plan recognition into three types, based on the attitude of the agent being observed: *keyhole* recognition, which involves fly-on-the-wall observations to which the agent is indifferent or unaware; *intended* recognition, where the agent is supportive and may interactively give clues or answer questions; and *adversarial* recognition, which entertains the possibility that observations may somehow be fabricated by the agent deliberately in order to deceive.

Of these, adversarial recognition has, in the past, been something of the poor relation. As Carberry (2001, p.32) states, "...little published research has examined situations where deception must be taken into account."[4]

Geib and Goldman (2001) represent an early contribution to this field. Previous work on goal recognition (they say) assumed full observability. This work, however, allows for the possibility that, in recording observations, some actions may have gone undetected and it provides a means of inferring them. This is achieved by assuming actions that *must* have occurred based on those that are subsequently seen. The Geib and Goldman model assumes that there is a known upper bound of unobserved actions and also that observations that are made occurred in the order that they were recorded. One key limitation, however, is a dependence on deceptive plans being explicitly added to the plan library. In effect, this places deception-detection on a par with traditional goal recognition: if 'adversarial' plans are loaded into the plan library, observations can be matched against them in the usual way. Reliance on plan libraries is thus a problem for adversarial recognition, just as it is for keyhole recognition. Typically, plans are domain-specific and, even within a given domain, it is impossible to cater for every eventuality. So, plans must be hand-coded (or labelled) based on the input of domain-experts or extracted somehow from other sources.

With an increased focus on HMI and anxieties about potentially hostile AI,[5] the landscape is slowly changing. Nevertheless, recent approaches to deception-detection continue to be domain-specific, relying either on deceptive plans (Geib, 2006) or anomaly detection, for example, as a 'decision aid' referring cases on to a trained operative (Elsaesser & Stech, 2007). Avrahami-Zilberbrand and Kaminka (2014) provides a recent survey of the field.

The framework presented by Avrahami-Zilberbrand and Kaminka (2014), following their survey, references an adversarial plan library but also identifies as anomalous any plan that *fails* to make a match. The authors combine this approach with a 'worst case assumption'. This tactic, whereby the agent is assumed to be pursuing whichever

[4] There may be some justification for the lack of attention deception has received from the PAIR community. Timothy Levine, editor of the Encylopedia of Deception (Levine, 2014a), asserts that people are right to believe one another because usually they do tell the truth (Levine, 2014b).

[5] An open letter calling for a ban on autonomous weapons, presented at the International Joint Conference on Artificial Intelligence (IJCAI) 2015, garnered over 22,000 signatures (The Future of Life Institute, 2015).

ambiguous plan is least good for the observer, provides a mechanism to protect against suspicious behaviour even if it is (ostensibly) unlikely to occur. Avrahami-Zilberbrand and Kaminka's work is of particular interest because the domain is path-planning-related and, although dependent on a plan library, accumulation of plans is built into the model. The context is one of surveillance. Agents are observed by cameras and the trajectories included in the plan library are assembled using supervised machine learning over observed paths.

Ambiguity, tackled by Avrahami-Zilberbrand and Kaminka (2014) using the worst case assumption, is a major problem for goal recognition, whether matching against plans or against optimal plan costs, and one standard approach to deception involves deliberately engineering observations to be as ambiguous as possible (e.g., Keren et al., 2014; Kulkarni, Klenk, Rane, & Soroush, 2018). When this occurs, a goal recognition system is effectively placed on an adversarial footing. It can either make a random selection, 'wait and see' or attempt some other means of disambiguation. Cost can be used for this purpose. Sukthankar and Sycara (2005) disambiguate in favour of whichever goal can be accessed at least expense to the observed agent while Mao and Gratch (2004) calculate the estimated expected utility of competing hypotheses and, assuming rationality, if two goals are equally probable, assess that the most likely goal is the one that maximises the observed agent's expected value. An alternative approach is to make a 'worst case assumption' (as discussed above) whereby, rather than taking the *observed* agent's preferences into account, selection is made based on whichever possibility risks maximising the *observer's* expected costs (Avrahami-Zilberbrand & Kaminka, 2007; Tambe & Rosenbloom, 1996). If the situation is dynamic and interactive, Tambe and Rosenbloom (1996) suggest a further strategy: that of 'active ambiguity resolution', whereby the observer tries to provoke the observed agent into an action capable of 'flushing out' its true intent.

2.3.1 Goal Recognition Design

Ambiguity is one of the key problems tackled in the emerging field of goal recognition design, introduced by Keren et al. (2014). This innovative contribution to offline goal recognition also builds on the Ramirez and Geffner model. Goal recognition design involves modifying a domain's layout in order to achieve goal recognition more easily. The first step in the process is analysis of the problem domain to determine 'worst case distinctiveness', that is, the maximum number of steps in (or length of) an optimal plan before its goal can be uniquely identified. Briefly, a shared optimal path prefix is identified using a compilation whereby two agents, operating within the same model, aim at different goals but receive a discount for acting together.

The method of calculation is ingenious but dependent on classical planning technology and potentially cumbersome in a path-planning context. It is also based on an assumption that each agent's behaviour is fully rational. In a subsequent paper, how-

ever, Keren, Gal, and Karpas (2015) extend their model to consider *sub*optimal plans. To handle this situation, they employ two concepts: 'bounded non-optimality', whereby they assume the agent's behaviour is naively suboptimal; and 'bounded deception', whereby the agent's behaviour is deliberately suboptimal with the intention to deceive. In either case, the idea is to cap the number of permissible steps (or permissible distance travelled) within a fixed budget. The underlying problem here is not explicitly addressed: namely that, without introducing a budget, a suboptimal path's 'worst case distinctiveness' is potentially infinite. However, alternative methods of identifying distance travelled along a suboptimal path are not considered.

Keren et al. (2016) extend the notion of worst case distinctiveness into partially observable environments based on the idea that a deceptive agent may be able to control which actions 'emit' observations (e.g., because the agent's location is betrayed by a mobile phone signal and the mobile phone can be turned on or off at will). Under this assumption, it is possible to maintain an ambiguous path for a greater distance. Again, the distance is calculated from the initial state using optimal (or 'bounded optimal') paths. Meanwhile, we note that this strategy, which is essentially one of deliberate deception, is characterised as 'privacy protection'.

2.4 Deception

Deception is more difficult to define than goal recognition and more challenging to quantify. Its nature has been debated by philosophers since Plato's Phaedrus (Murray, 1988). The Stanford Encyclopedia of Philosophy (Mahon, 2008) suggests five different definitions then finds at least two objections to each of them. Shim and Arkin (2013), seeking a system of classification suited to robotics, cite taxonomies from disciplines as diverse as psychology and cyberspace and, although not every aspect of the topic is central to this thesis, many have useful and interesting implications for future work.

With regard to ethics, (Carson, 2010) has said that the very word "deception" implies wrong-doing. Bowyer (1982), on the other hand, (whose theory we consider more extensively in Section 2.4.2, below) discusses many benign applications, such as sport and magic. As Bowyer points out, even when applied to military strategy—for example, in the retreat from Gallipoli—deception is the technique that *saves* lives; it is the failure to deceive that leads to carnage. Also, of course, when we rename deception as "surprise" or "privacy", the ethical dilemma tends to disappear. Indeed, as noted above, when *advocating* deception, the literature prefers to characterise it as 'privacy protection' (Keren et al., 2016).

Interestingly, deception is closely linked to theory of mind, which we have already identified as a key aspect of goal recognition (Baker et al., 2011).

The lie is... a normal phenomenon of what Heidegger calls the 'Mit-sein'. It

> *presupposes my existence, the existence of the Other, my existence for the Other, and the existence of the Other for me.*
>
> (Sartre, 1956, cited by Chisholm & Feehan, 1977, p.151)

Sartre's intuition seems confirmed by Short, Hart, Vu, and Scassellati (2010). In their experiment, a humanoid robot cheated at a game of rock-paper-scissors, either by saying it had won when it had not or by changing its response after noticing that it had lost. The authors found that the act of cheating was in itself sufficient for children to attribute the presence of a mental state to a machine. However, an interesting 'circular logic' was also in play: when the robot cheated by lying—that is, by making a false statement about the game's outcome—the children were inclined to regard it as a malfunction rather than a deception. Why? Because a 'malfunction' is something that happens to a machine; and a machine has no mental capacity so it cannot cheat! One implication for a path-planner engaged in the deception of humans is this: the more robotic a path appears to be, the less likely it is that an observer will anticipate that they are about to be deceived.

Deception obviously involves false belief. Therefore, we briefly note the problems presented by false belief in relation to knowledge representation. Mathematical logic, upon which computer systems are based, is monotonic: the addition of new information can only add knowledge; it cannot decrease the set of propositions previously known to be true (Russell & Norvig, 2013). That is, if $A \vdash q$ and $A \subset B$, then $B \vdash q$ (McCarthy, 1980, pg.28). Deception presents a problem: by definition, it induces false belief; an intelligent system that maintains a model of the deceived must be capable of modelling false belief; and a model capable of admitting false belief is inherently non-monotonic. Such a system requires an alternative logic capable of belief revision, such as the dynamic epistemic logic of van Ditmarsch, van der Hoek, and Kooi (2007). The impact for us is in terms of motivation. A machine based on standard logic is incapable of revising its beliefs; but the richer the logic, the more processing it requires. If we can trick a computer system into accepting a falsehood, its correction process may be anything from time-consuming to impossible.

Chisholm and Feehan (1977) present a catalogue of further philosophical considerations: degrees of intentionality, the distinction between believing a proposition and not disbelieving it, the difference between deceptions achieved by commission or omission, those that result in the target adding to their stock of beliefs or ceasing to believe something they previously held true, and those where a target is deceived into continuing an existing false belief versus those where the deception has forced a change. Although, for path-planning, we can largely ignore these aspects, this work is interesting in that, by grading and classifying on each of these different dimensions, it is one of few that attempts to rank deceptions in order. These rankings, however, do not represent how likely it is that a deception will succeed but rather how relatively immoral it is.

2.4.1 Deception and Artificial Intelligence

Arguably, AI began with deception. The thinking machine in Turing's imitation game (1950) was essentially a deceiving machine. Notwithstanding subsequent reworkings, the original Turing Test required the machine to be substituted for a man, who was engaged in deceiving his interrogator into believing that he was actually a woman (and that the woman, also under interrogation, was actually a man). If the machine succeeded in deceiving its interrogator as often as the *real* man had managed to do then it would be reasonable to conclude, according to Turing, that machines can think.

In his paper, "Artificial liars: Why computers will (necessarily) deceive us and each other", Castelfranchi (2000) argues that the agent-oriented paradigm—within which autonomous agents interact with the environment and each other to further their own self-interest—must inevitably lead to deceptive practices, whether the agent is strategic (calculating each situation's utility) or purely reactive. Anyone who thinks Castelfranchi may be exaggerating has not witnessed the demonstration of Google Duplex, recorded at Google's I/O conference, in which a robotic personal assistant masquerades as a human for no more important purpose than to book a hair-dressing appointment (Google, 2018).

In computer science, much of the work on active deception (as distinct from its detection) has involved game theory. Unlike Google's personal assistant, the focus here is on deceptive acts that occur spontaneously (rather than being crafted) as the inevitable consequence of agents pursuing self-interested goals, unchecked by social or moral constraints. This is particularly true of agents operating within a game theoretic paradigm with its emphasis on strategic and, typically, selfish motivations (Castelfranchi, 2000).

In a partial information game, for example, Player A may be able to gain an advantage by taking an action which suggests the game state is other than it actually is. Player B is then deceived into responding with the action that, given the implied (but false) game state, appears to offer the best pay-off when, in fact, it offers the worst (Hespanha et al., 2000). Even in full information games, deceptions can spontaneously occur if one player has more computational power than the other (Hespanha, 2007). In such cases, the more powerful player can act to prompt a response that *appears* advantageous to a less powerful opponent looking ahead to the extent of its preview horizon at, say, n steps, knowing that this response will lead it to disaster at step $n + 1$. Limited computational power can also be exploited by using diversionary tactics whereby a player may be tempted into exploring the *wrong* locations, leaving it with insufficient resources to explore the *right* ones: a strategy applicable not only to machines exploring the branches of a search tree but also to humans, unable to look in two places at once.

In "A Theory of Deception" Ettinger and Jehiel (2010) offer a more sophisticated approach with their game theoretic account of 'fundamental attribution error'[6]. The sophis-

[6]Fundamental attribution error is a concept from social science which describes the tendency to attribute behaviours to an actor's personality rather than taking external factors into account.

tication, however, is primarily in connection with the target or 'mark' (i.e., the deceived player), not the deceiver. Their definition of deception (p.2) sums up their approach: "the process by which actions are chosen to manipulate beliefs so as to take advantage of the erroneous inferences." Their framework is a two-player multi-stage game with partial information in which players have different stereotypes and also different cognitive types. Cognitive types vary on two dimensions: 'analogy' (how far the player has stereotyped its opponent) and 'sophistication' (how far the player recognises qualitative differences between behaviours). Players assess each move based on their analogy assumptions and belief about the other player's type, updating their beliefs during the course of the game. By endowing the mark with the capacity to update its beliefs, it is made deceivable. The paper goes on to show that it may be useful for the deceiver to 'prime' the mark (e.g., with a demonstration of trustworthiness) before performing the deception; and that the ideal mark is neither fully rational (because it must be capable of false belief) nor fully irrational (because it must be able to make inferences based on observations).

Social roboticists Arkin et al. (2012) set out a 13-step algorithm which attempts to position deception within a more ethical framework. They adopt a definition from biology which, being inclusive of plants and animals, is broad enough also to encompass robots, software agents, and computer systems: "a false communication that tends to benefit the communicator" (Bond & Robinson, 1988, p.295). In this framework, the target's mental state is modelled as a feature set, an action model and a utility function. Game play is represented in a standard grid but, in addition to representing the payoffs available to each player based on their actions, it also tracks 'interdependence' (the extent to which one individual's action is influenced by the other) and 'correspondence' (the extent to which the players' interests coincide). By reference to these extra dimensions the system is able to determine (a) whether, in the current situation, deception is a useful strategy and (b) whether or not it is likely to be believed.

Like Ettinger and Jehiel (2010), Arkin et al. are concerned not with *how* one might craft a deception but with *how likely* it is that a pre-conceived deceptive strategy will achieve the desired outcome, given the nature of the target.

Though less overtly concerned with deception than the above, Tambe (2015) discusses a stable of projects—which include ARMOR, a patrol scheduling system developed for Los Angeles International Airport (Pita et al., 2008) and the PROTECT patrolling system, successfully deployed by the US Coast Guard (Shieh et al., 2012)—all of which deal with a similar problem: how to deploy limited resources so as to maximise the protection they afford. The general approach is to model each security situation as a Stackelberg game (i.e., leader/follower) in which the leader (protector) adopts a mixed strategy and the follower (attacker) is assumed to have performed surveillance and responds with a pure strategy. The implemented systems output patrolling schedules that are 'optimally randomized' and maximise expected utility for the defender based on the number of people

Table 2.1: Six strategies.

Dissimulation	Simulation
Masking: hide the real by making it invisible (e.g., camouflage, palming a card).	**Mimicking**: show the false by having one thing imitate another (e.g., use a double, make the sound of coins clinking when the real coins are elsewhere).
Repackaging: hide the real by disguising (e.g., disguise a warship as a freighter, exchange costumes with an assistant).	**Inventing**: show the false by displaying another reality (e.g., create rubber tanks or wooden guns, make forgeries for mind-reading acts).
Dazzling: hide the real by creating confusion (e.g., use a cipher, use equivoque to force a 'choice').	**Decoying**: show the false by diverting attention (e.g., feint left but turn right, sleight of hand).

(i.e., potential lives saved) at each target location.

PROTECT originally modelled its patrol targets as nodes on a graph with paths (edges) between them but this path-planning solution was ultimately rejected on three grounds: the travel times between nodes could not be known with certainty; it failed to exploit the local knowledge of boat crews; and it was perceived as micro-managing by users (Shieh et al., 2012). Nevertheless, these security games are worthy of attention for several reasons: (a) they are closely related to deception; (b) they are closely related to path-planning; (c) they reduce the mind of the attacker to a probability; and (d) although a path-planning solution may seem prescriptive to human crews now it could become a welcome resource for autonomous robotic patrols of the future.

2.4.2 A General Theory of Deception

We now consider the work of J. Bowyer Bell and Barton Whaley. Writing in the 1980s, they presented what they declared to be the first (and to our knowledge still the only) *general* theory of deception (Bell, 2003; Whaley, 1982)[7]. Drawing examples from magic, the military and elsewhere, they define deception as the deliberate distortion of a target's perception in pursuit of advantage or, more succinctly, the 'distortion of perceived reality'.

Bell and Whaley found that all deception involves some combination of 'hiding the real' (dissimulation) or 'showing the false' (simulation). They articulated three distinct strategies in each case (see Table 2.1 and Figure 2.3) though the strategies may be used in combination with one another. They note too that *every* deception (including each of the three simulation strategies) involves dissimulation to some degree (Whaley, 1982).

This thesis accepts Bell and Whaley's definition. It adopts their terminology and their perspective on deception: that it is usually beneficial (it is the course of action that, in warfare, saves lives where an 'honest' show of force cannot), is often enjoyable

[7]The theory was first published jointly under a pseudonym (Bowyer, 1982).

Figure 2.3: Hiding the real, showing the false.

Figure 2.3 shows a slide of Bell & Whaley's six strategies from 'The Art of Deception', presented by the Human Science Operations Cell at MI5's Joint Threat Research Intelligence Group, leaked to the public via edwardsnowden.com (GCHQ, 2014).

(our sporting heroes thrill us when they duck and weave) and is anyway an undeniably pervasive human trait. Although they could not know it at time of writing, as we move towards an expectation of collaborative human-machine relations, this final point is an important consideration in the development of AI. In terms of applicability, their theory has the great benefit that it is unencumbered by the ethical considerations which—as already discussed—have dogged this topic. They unashamedly discuss their craft in terms of deception (and 'cheating').

According to Bell and Whaley, there are six kinds of deception, three that involve dissimulation (masking, repackaging and dazzling) and three that involve simulation (mimicking, inventing and decoying) but there is only one *way* to cheat.

Operationally, in order to deceive someone (or something), we must change the characteristics—or 'charcs'—detected by the target's senses. A 'charc' may be any feature that is available for manipulation and can be employed to achieve the desired effect. For example, a hat on a stick is a charc that could be used to achieve the effect (from behind a rock, say) of a soldier. In the context of path-planning, our charcs are limited to location and cost but it is easy to see how the opportunities for deception (and the potential complexity of the task) increase exponentially with every additional charc, such as speed, heading or acceleration.

Having chosen the category and identified the available charcs, the deceiver must decide on a particular ruse. Bell and Whaley enumerate five types of ruse, named according

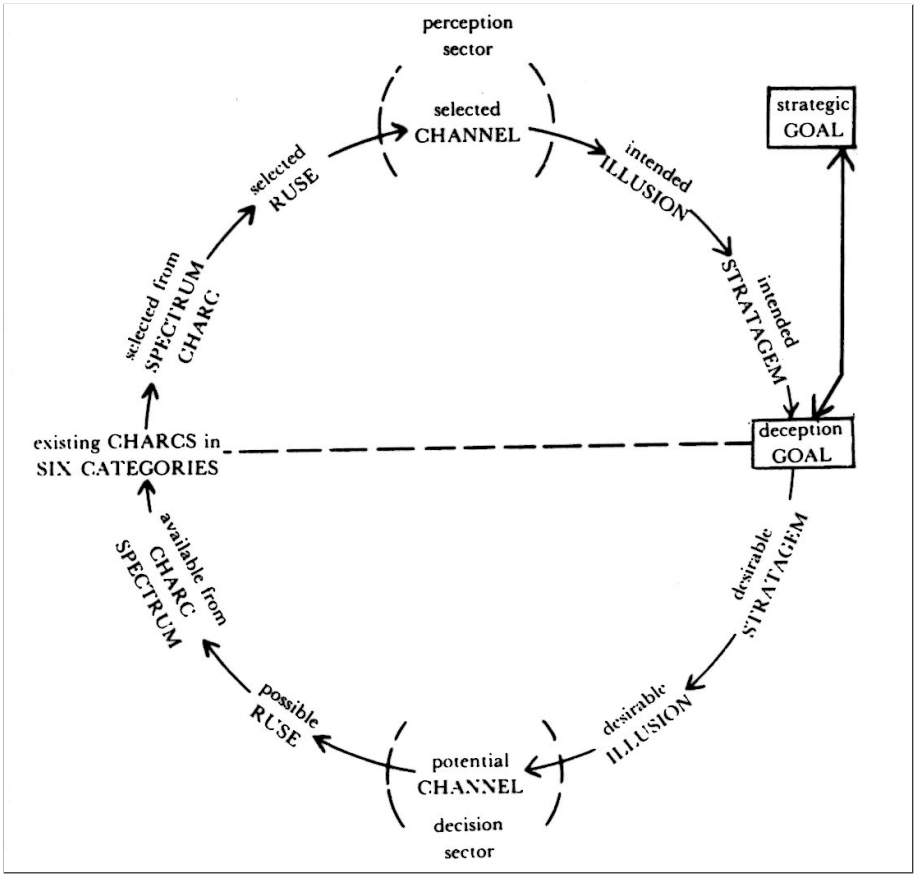


Figure 2.4: The deception planning loop.

The deception itself consists of crafting and presenting an illusion. It begins with a strategic goal. From this, the deception planner constructs a *ruse* from a range of possible *charcs* and projects it across some *channel* of communication. The illusion is now in play. (Bowyer, 1982, p.71)

to the effect or impact they aim to achieve: unnoticed, benign, desirable, unappealing or dangerous. The purpose of the ruse is to create some kind of 'cover' or effect, which operates over a 'channel' or medium (such as the air, radio waves or, potentially perhaps, a mobile signal). If the effect is achieved, the illusion is complete.

Of course, completing the illusion does not guarantee deception. To achieve deception, the target (or observer) must *accept* the illusion. Thus, if developing a deceptive path, it can only be assessed on the basis of its *potential* to deceive, for example as a probability. How unlikely does the observer believe the agent's real goal to be?

Note. An illusion does not need to last forever. A temporary deception may be enough to achieve the desired result. In path-planning with full observability, the illusion is always temporary: it can only last until the agent is seen to arrive at a (hopefully) unexpected destination. As Bell (2003) warns, however, the target's response may not always be as the planner hopes or expects.

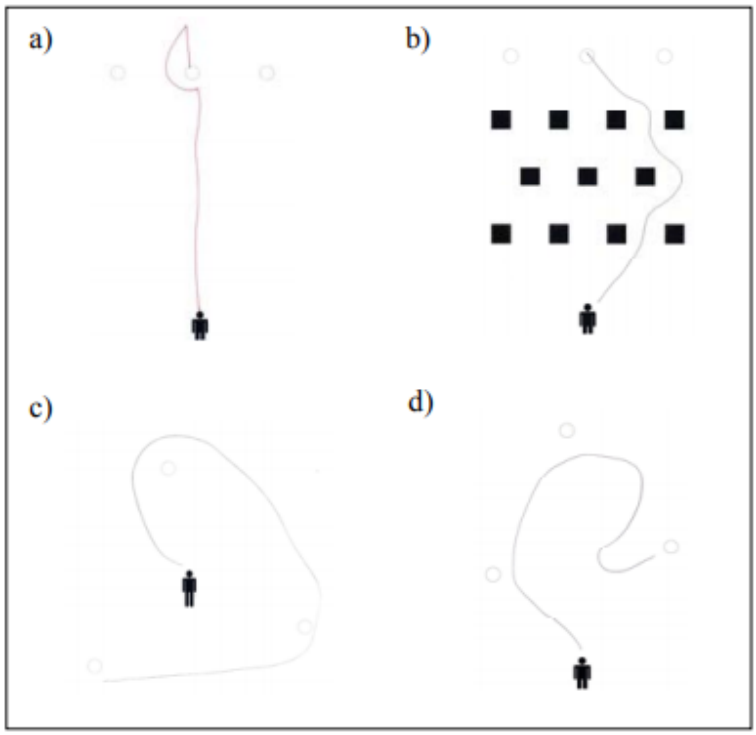


Figure 2.5: Hand-drawn deceptive trajectories (Jian et al., 2006, p.1564).

Bell and Whaley present a comprehensive theory, from which this thesis borrows only part. We adopt the notions of simulation and dissimulation and make particular use of 'dissimulation by dazzling'. This is a strategy typically brought into play when other attempts at hiding have failed or are unavailable. It is useful in path-planning because it applies in situations where the observer knows you are there: that is, it enables a path-planner—who does not know when or where she may be observed—to hide in plain sight. In Section 4.4.1, we consider how other of Bell and Whaley's strategies might be implemented in the context of path-planning.

Their complete 'deception loop', which illustrates the process from conceiving the strategic goal to its operational conclusion, is reproduced at Figure 2.4.

2.4.3 Deception in a Navigational Context

Jian et al. (2006) conducted a pencil and paper study to find out if deceptivity could be detected from a path-plan. The study is of special interest, although related to cognitive science rather than computer science, because: (a) it concerns both deception and goal recognition; (b) it places both topics squarely in the context of path-planning; (c) it demonstrates that truthful paths tend to be optimal paths (in line with the intuition of cost-based goal recognition).

Using four different layouts, each showing a start point and three possible goals

(i.e., one 'true' goal and two decoys), subjects were asked to assume that they were under surveillance whilst drawing a path, from start to goal, without giving away their destination. While the control group, who were drawing a 'normal' path, under no such constraints, predictably drew the shortest, most direct path, the experimental group produced a convoluted assortment which adopted 38 recognisably deceptive strategies such as "overshoot passing target", "straight towards decoy", "concentric circles around starting point". Participants tended to overcompensate in order to obscure their intent and in over 75% of cases made their initial move away from the real goal and towards a bogus goal. Thus, the study found that it might indeed be possible to detect deceptive behaviour simply by observing the presence of one of the recognised strategies.

Knowing that a path is deceptive does not necessarily help an observer to determine the true target, however. In his game-theoretical account, Hespanha (2007) suggests that, if an agent is able to control all the information available to the observer, the use of deception can render *all* observations meaningless: when the observer does not know what to believe, she must make her decision as if she had made no observations at all. (Hespanha, 2007). This is precisely the objective of a dissimulation strategy and one successfully exploited in an experiment by Root, De Mot, and Feron (2005) in which drones conduct reconnaissance while under surveillance. The experiment is presented as a non-cooperative zero sum game which runs as follows. The invader, in control of the drones, selects a ground path from a set of all feasible paths and deploys a team of drones to reconnoitre it—which involves flying over every edge and pausing over every node of the selected path. The defender observes the drones and decides whether to set an ambush and at what location. If the defender sets an ambush anywhere on the selected path, the defender wins; if not, the invader wins.

In a domain modelled as a graph, the system (acting as invader) first selects its ground path, then constructs a set of flight plans that involve overflying not only that path but *every* edge capable of supporting military traffic. To take advantage of Hespanha's insight, ideally the invader traverses every such edge in the graph. It is a strategy which emphatically *hides the real*. The defender (observer) is left to select between multiple possible routes, all of which—assuming equal prior probabilities—are equally probable.

A different approach, inspired by the food hoarding behaviour of squirrels, demonstrates a clear application of *showing the false*. Squirrels, according to Shim and Arkin (2012), have two hoarding behaviours: 'caching', which involves stashing their food in multiple dispersed locations, and 'protection', which involves patrolling the caches. The usual patrolling behaviour is simply to go from cache to cache and check on the food but if a squirrel becomes aware of a competitor nearby (i.e., an observer), it also visits empty locations, apparently with the intention of confusing the competitor about the location of the food.

The experiment tested the strategy using computerised robotic squirrels, each mod-

elled as a finite state automaton (FSA) with six states: 'caching', 'true patrolling', 'false patrolling', 'enough food cached', 'select true place', and 'select false place'. The robotic squirrel randomly selects which cache to visit by calculating transition probabilities based on the number of food items in each cache. The competitor, meanwhile, is also modelled as an FSA. It wanders the map and, when it finds a squirrel, decides whether it is at a cache based on the length of time that the squirrel remains stationary. Once it identifies what it thinks is a cache, the competitor steals whatever food it finds.

Notwithstanding the simplicity of the implementation, the experiment clearly demonstrated the effectiveness of the strategy: deceptive squirrel robots retained their food significantly longer than non-deceptive squirrel robots.

The general problem of deception (or privacy) and location tracking arises in many other settings which we mention here only briefly: amongst the robotics community, for example, in consideration of the panda tracker problem (O'Kane, 2009) in numerous ambush, pursuit-evasion and patrolling games (Shieh et al., 2012). Furthermore, in a real-world setting (as mentioned in Section 2.3.1 p.23), Keren et al. (2016) suggest the possibility of an agent concealing her location by deliberately shutting down electronic means of surveillance (e.g., by turning off her smartphone): a very literal application of dissimulation (hiding the real).

2.4.4 Emerging Trends

As AI applications spill out into the public domain, there has been an intensification of interest in human-machine interaction, bringing with it an increased emphasis on approaches (to AI in general) that are guided by an awareness of human intentionality.

Notably, Kulkarni, Srivastava, and Kambhampati (2018) propose a planning framework that supports both adversarial and cooperative environments. This work is one of few, within planning, that tackles deception as an active, intentional endeavour. The contribution revolves around the idea that an agent operating in the real world (e.g., working in a team with people or other AI) should be capable of concealing its intentions from adversaries while revealing them to associates. The key insight is that, whether the desired plan is adversarial or cooperative, it can be built in relation to the same observational model, aiming either to maximise or minimise ambiguity from that observer's point of view. The problem is therefore presented as one of 'controlled observability'. The authors avoid the issue of whether observations should be regarded as actions or states (considered by Sohrabi et al., 2016) by regarding them instead as a combination of the two: the output of some sort of 'sensor', whereby each observation corresponds to an action-state (i.e., an action *and* the state that results from that action). Each observation induces a belief-state in the observer and the objective of the agent is to ensure that the last belief state in the sequence is one consistent with some particular number of goals. An adversarial agent aims for a final belief state that is "k-ambiguous" (i.e., consistent with *at least* k goals);

a cooperative agent aims for a final belief state that is "j-legible" (i.e., consistent with *at most* j goals).

In a subsequent paper (Kulkarni, Klenk, et al., 2018), the authors apply insights from the world of cryptography and security over the same framework to achieve ambiguous plans that are also *secure*, in that the plan is the same no matter which goal is being targeted. This clever application of a fundamental cryptographic principle means that even if the observer knows which algorithm the planner is using, they cannot determine her real goal. It is unclear, however, how the planner should proceed when fully ambiguous solutions are unavailable (e.g., in a navigational scenario where obstacles prevent the agent from tracking equidistantly from multiple goals). Kulkarni, Srivastava, and Kambhampati's model can be extended to cover the case where an observer is aware of the goal (but not the plan that will be used to achieve it). The authors observe that, in this emerging area of human-aware (or 'human-in-the-loop') research, goal recognition in general (and intended recognition in particular) begin to overlap with another emerging field: that of explainable AI.

Sreedharan, Chakraborti, and Kambhampati explore this topic explicitly. Their approach combines intended recognition or 'explicable' AI (i.e., the generation of 'legible' plans that a human can readily understand) with 'explainable' AI (i.e., the generation of textual explanations for agent or robotic behaviour that might otherwise be difficult for a human to understand). Their algorithm depends on multi-model planning—similar to that usually required for deceptive planning—whereby the agent maintains not only its own model of the task at hand but an additional model, which represents the (human) observer's view of that task. The objective is to find the 'sweet-spot' between executing an optimal plan, which (typically) requires explanation, or a suboptimal plan, which is in line with human expectation and therefore requires no (or minimal) explanation. Their algorithm handles the trade-off between optimality and explanation generation using a control variable, set by a system designer. The authors suggest, however, that in future work the setting could instead be learned from any particular human's preferences over time.

We mention also the work of Pozanco, Yolanda, Fernández, and Borrajo (2018) on domain-independent counterplanning. Although this differs from deception, it involves goal recognition in the context of adversarial reasoning and clearly offers scope to incorporate deceptive planning in future work. The counterplanning problem involves two agents, one attempting to reach a goal (the seeking agent) and another (the preventing agent) whose objective is to prevent that goal from being reached. The setting is not dynamic. The seeking agent generates a plan from an initial state to the real goal and part of that plan (up to some time-step before its completion) is provided to the preventing agent for counterplanning to begin. The authors' approach involves the use of landmarks (Hoffmann et al., 2004). The preventing agent first uses goal recognition to obtain a proba-

bility distribution over the set of candidate goals, then extracts a set comprising whichever landmarks are most common amongst the most probable goals. One of the landmarks is now selected (the first, by time-step, that is reachable given the seeking agent's 'current' location) and it becomes the preventing agent's goal to negate that landmark.

The above frameworks suggest a trend towards 'multi-model' planning, which acknowledges goal recognition as a core feature of human-machine interaction and confirms the impression that deception and goal recognition are complementary aspects of a single problem. Thus, we have seen work that combines goal recognition with planning (Freedman & Zilberstein, 2017) and, as above, deception with cooperation (Kulkarni, Srivastava, & Kambhampati, 2018) and goal recognition with counterplanning (Pozanco et al., 2018). As a final note, the emerging field of rebel agents (Coman & Aha, 2018) is tangentially related. These are agents, autonomous to the degree that they may *refuse* to perform certain actions, even though they seem to be indicated by the plan that they are supposedly following or the one expected by humans (or other agents) in their team. Combined with a capacity to 'nudge' associates towards adopting preferred courses of action (Borenstein & Arkin, 2016), it remains to be seen whether *ethical* rebel agents represent a promise or a threat; whether they will be able to protect us from the dystopian future that Castelfranchi's "artificial liars" (2000) seemed to predict or whether they will be instrumental in bringing it about.

2.5 Foundation to Our Approach

We conclude this literature review by now offering a more detailed account of the work that we rely on most closely: two seminal papers from Ramirez and Geffner on 'plan recognition as planning'. As discussed, these papers have been extended and adapted by many previous authors. They form the foundation of our approach to goal recognition in Part I of this thesis. Furthermore, the probability distribution formula used in their 2010 paper is one (the other being from Vered and Kaminka's work on goal mirroring, discussed p.18) that we analyse and build upon in Part II, Section 4.1.

Ramirez and Geffner (2009) define the goal recognition problem for task-planning as one of "planning in reverse" (p.1778). Their framework takes a classical STRIPS-style domain $\langle F, A \rangle$, where F is a set of fluents and A is a set of actions a, each of which has a precondition, add and delete list $Pre(a)$, $Add(a)$ and $Del(a)$, all subsets of F. An action a can occur in state s if $Pre(a) \subseteq s$. The initial state is assumed to be fully observable and the domain is deterministic; that is, if a occurs in s, a new state s' results such that $s' = (s \cup Add(a)) \setminus Del(a)$. A plan is a sequence of actions $\pi = a_1, \ldots, a_k$ that maps a specified initial state $I \subseteq F$ to a goal $G \subseteq F$. Typically, each action has a cost $c(a)$ and the cost of a plan is the cost of all the actions in the plan, $cost(\pi) = \sum_{i=1}^{|\pi|} c(a_i)$.

In this context, Ramirez and Geffner (2009) articulate the goal recognition problem for task-planning as a tuple $\langle\langle F, A\rangle, \mathcal{G}, I, O\rangle$ where: $\mathcal{G} \subseteq 2^F$ is a set of possible goal states; $I \subseteq F$ is the initial state; and $O = o_1, \ldots, o_k$ is a sequence of observed actions, that is, $o_i \in A$ for all $i \in \{1, \ldots, k\}$. The solution to the problem is a set of goals, the optimal plans for which satisfy the observation sequence. This is achieved, the authors say, if a plan $\pi = a_1, \ldots, a_n$ *embeds* the observations $o_1, \ldots, o_m$ in such a way that the order of actions is preserved; that is, there must be a monotonic function f that maps the observation indices into the action indices such that $a_{f(j)} = o_j$ for all $j \in \{1, \ldots, m\}$. Any goal for which there is an optimal plan that meets this criterion is part of the solution set.

The framework performs well online or offline. Ostensibly, it is necessary to make two calls to the planner for every goal—first to obtain an optimal plan, then to obtain the optimal plan *that complies with observations*—and to repeat this, in an online environment, with every new observation. In practice, however, the optimal path costs need only be calculated once for each goal and can then be reused (Vered et al., 2016). Moreover, since the model is only interested in *optimal* solutions, optimal costs can also be used as upper bounds, meaning that all subsequent calls to the planner to plot "optimal paths that comply with observations" can be heavily pruned. A major drawback with this framework, of course, is that it only identifies a goal if observations conform to an optimal plan whereas, realistically, agents behave *sub*optimally. Arguably, this is especially true in navigational domains, which tend to be less structured than those encountered in general task-planning; as Pattison and Long (2013) point out, the number of possible routes a person might take through a typical city, even with a fixed starting point, is intractable. Thus, rational behaviour should be assumed as a guiding principle only.

In their 2010 paper, Ramirez and Geffner present an alternative *probabilistic* framework which uses classical planners off-the-shelf to identify, not a set of goals, but a posterior probability distribution which prefers those goals whose plans 'best' satisfy observations. The probabilistic problem definition $\langle\langle F, A\rangle, \mathcal{G}, I, O, Prob\rangle$ adds a prior probability distribution *Prob* and its solution is a *posterior probability distribution* which prefers those goals whose plans best satisfy the observations, as determined by the principle of rationality.

The authors derive their solution from Bayes' Rule making two assumptions: that the probability of a plan is inversely proportional to its cost; and that probabilities of multiple plans for the same goal are dominated by the highest of those probabilities. The first assumption is essential to the model and is encapsulated in the notion of 'cost difference' between the cheapest plan for a goal, given the observed actions already taken, and the cheapest plan that could have reached the goal *had the observed actions not occurred* (i.e., had the agent's actions differed from those observed by even the smallest degree). Denoting the costs of these optimal plans as $cost(G, O)$ and $cost(\overline{G}, O)$, respectively, the

cost difference amounts to a function, $costdif : 2^F \times A^* \mapsto \mathbb{R}$, defined as follows:[8]

$$costdif(G, O) = cost(G, O) - cost(\overline{G}, O).$$

By comparing cost differences for all $G \in \mathcal{G}$, the authors propose generation of a probability distribution across $\mathcal{G}$ with the following important property: *the lower the cost difference for a particular goal, the higher its probability.* Concretely, they propose the application of a Boltzmann distribution, which yields:

$$P(G \mid O) = \alpha \frac{e^{-\beta\ costdif(G,O)}}{1 + e^{-\beta\ costdif(G,O)}},$$

where α is a normalising constant and β a positive constant which can be used as a rate parameter to 'modulate' the assumption that the observed agent is pursuing plans sensitive to the same cost function used by the observer: as β approaches zero, the distribution flattens out (Ramirez, 2012, p.63).[9]

These two seminal papers moved the focus in goal recognition from plan libraries to declarative goals (and a model of the environment or 'domain theory'). The key is that optimal path costs, including both terms in their cost difference formula (i.e., optimal cost via the observations and optimal cost *not* via the observations), can be computed using classical planning technology, despite the fact that planners do not natively handle requirements about observations. Ramirez and Geffner proved that such requirements could be encoded back into the planning task.

Notwithstanding the scholarship of Ramirez and Geffner's probabilistic model (2010), unlike their 2009 framework, it is computationally expensive. To calculate cost difference, an agent reasoning about another agent's intent must *always* perform (and complete) two planning tasks for each potential goal. Although plans are still generated on-the-fly, they cannot be reused and there is no obvious basis on which they can be pruned. Furthermore, the planning tasks themselves are arguably more complex than merely planning for each goal as they not only embed the behaviour observed so far but also reason negatively about it (to obtain the cost of an optimal path that does *not* comply with observations).

Our approach takes advantage of the economies that can be achieved under the 2009 model in the context of the 2010 framework, then extrapolates an even more economical solution. The concept of 'cost difference' is central to our work. As you will see (in Section 3.1), we simplify both terms in the equation to arrive at an alternative formulation (which can be used in navigational domains interchangeably with the original) in which all references to the observation sequence are eliminated. This is important because, as Vered et al. (2016) point out, generation of an optimal plan under Ramirez and Geffner's

[8] This and the following formula are presented again later in this thesis in the context of our goal recognition framework for path-planning. See Equations (RG1) and (RG2), pp.47 - 48.

[9] We use several variations of Ramirez and Geffner's probability distribution formula in this thesis. This is the formulation that appears in the codebase referenced from Ramirez and Geffner's 2010 paper. For further details, see Appendix A.

definition of *not* complying with observations (i.e., such that it avoids at least one observation but may go through none, one, some or all of the others) is computationally demanding. As they explain, although it is achievable using Ramirez and Geffner's approach in a STRIPS-like environment, it cannot be done natively by motion planners in a continuous domain. In fact, Kaminka et al. (2018, p.6203) suggest that "the requirement is meaningless in continuous domains" since it is almost always possible to create a plan that does *not* comply with observations at an arbitrary (and immeasurably small) distance from an optimal plan that *does* comply with them. Meanwhile, in the context of discrete path-planning, it is straightforward to modify a path-planning domain to accommodate 'avoidpoints' (locations that must not be traversed) by marking them impassable, as if they were obstacles and it is common to include 'waypoints' (locations that *must* be traversed) by subgoaling; but to find an optimal path that manages to avoid at least one location, even though it may go through one or more of the rest, is not a native function for standard path-planning algorithms.

The simplest way to avoid this negative reasoning is to substitute the more straightforward 2009 formulation based on the cost of an optimal path. We note that Ramirez and Geffner explicitly reject this simpler formulation (2010, p.1123) because it fails to recognise the possibility of *negative* cost difference (which can arise if the optimal plan that does *not* comply with observations involves a detour, making it more costly than the optimal plan itself). Notwithstanding this objection, it is the approach taken by Escudero-Martin et al. (2015), discussed previously (p.20).

Escudero-Martin et al. (2015) justify substitution of the simpler cost difference formula by arguing, though without proof, that both terms return the same result in the majority of situations. Furthermore, although they identify it as "somewhat counterintuitive" (p.762), they do not formally define the special case where results returned are different, nor do they discuss the implications when that occurs. Their experimentation, however (which replicates Ramirez and Geffner's experiments in the discrete domain), confirms that "for most problems there are multiple distinct optimal plans for each goal". This implies that "for most problems" the special case cannot arise.

In the next chapter—as a first step towards our examination of goal recognition as path-planning—we prove the precise cases where the simpler cost difference that we have just discussed, used previously but without formal justification by authors such as Escudero-Martin et al. (2015) and Vered et al. (2016), is equivalent to Ramirez and Geffner's more complex cost difference formula. We demonstrate the special case where the formulas return different results and fully explore the implications of that difference (see Sections 3.1.2, p.48 and 3.4.1, p.80).

Part I

CHAPTER 3

Goal Recognition as Path-Planning[†]

> *"Jumping to conclusions is efficient if the conclusions are likely to be correct and the costs of an occasional mistake acceptable, and if the jump saves much time and effort."*
>
> *–Daniel Kahneman*

Goal recognition (GR) is not an exact science. It is a problem of jumping to correct conclusions: *jumping* because we hope to do it quickly; and *correct*, hopefully much of the time. It is the problem that underlies this entire thesis. "How, from my limited observations, can I correctly conclude your destination? How can I prevent you from correctly concluding mine?"

Consider an airport surveillance system. An agent of interest enters the terminal. The problem is to assess whether she is making for some particular boarding gate; if she is, the system will raise a flag to trigger her interception. Now, using state-of-the-art GR, the accepted solution would be to track the agent's movements throughout the airport accumulating as many observations as possible (which might include watching her criss-cross the terminal many times as she buys a paper, uses the bathroom, gets coffee and so on) repeatedly calculating and recalculating the probabilities of each gate she might be making for until the target gate becomes the most probable (or exceeds the probability of all other gates by some given margin).

In this chapter, however, we develop solutions such that, provided we know an agent's starting point, we need only discover her current location to generate a probability distribution which ranks goals in the same order as if we had tracked her all over the airport. Our model, single-observation recognition, can be used by a human operator to make spot-checks (e.g., on an agent acting suspiciously) or to pre-calculate a probability distribution across goals (e.g., boarding gates) at any point in the domain *even before the agent*

[†]Some of the work in this chapter has been published previously (Masters & Sardina, 2017a, 2019a).

has entered the terminal. Alternatively—without generating any actual probabilities—we show how to calculate a radius within which the target gate is guaranteed to have become the most probable. Using this radius of maximum probability (RMP), instead of tracking an agent all over the airport, surveillance operatives can focus resources strictly on those locations where, should the agent appear, the target gate is already known to be the most likely gate: if she is spotted there, the flag should be raised.

Our approach to GR is to take a state-of-the-art model developed for task-planning and reduce it to the special case of path-planning. As discussed in Chapter 2, the innovative and principled model on which we build was presented in two seminal papers (Ramirez & Geffner, 2009, 2010), which introduced the 'plan recognition as planning' approach to GR (reviewed at Section 2.5, p.35). Recall that this is a cost-based model, which uses classical planning technology to generate plans as-needed over a model of the domain. Their 2009 solution identifies a subset of 'optimal' goals (that is, the observed behaviour is consistent with an optimal plan for each goal in the subset). Ramirez and Geffner's 2010 model, on the other hand, is more flexible: it generates a probability distribution across the set of possible goals, based on the cost difference between the cheapest plan that can be achieved, given the behaviour already observed, and the cheapest *alternative* plan (that is, the best plan compatible with any slight deviation from the actions already seen). One drawback of the probabilistic account is that it is computationally expensive, in that it requires two plans to be generated per goal, the second of which (to find the 'alternative' plan) is a complex task. In this chapter, however, we show that, in the context of *path*-planning, the 2009 and 2010 approaches can be combined to derive solutions that are much more economical.

The rest of this chapter is organised as follows. In Section 3.1, we show how Ramirez and Geffner's cost difference formula can be deconstructed to arrive at single-observation recognition, so-called because unlike competing models, given the initial configuration (which we take to include the agent's starting point and possible goals), it requires only one observation to generate a probability distribution across goals. Section 3.2 shows how our insight with respect to single-observation recognition (i.e., the relationship between an agent's current location and the probability of each goal) can be exploited to determine a goal's RMP, that is, the distance from a goal within which it is guaranteed to be more probable than any other goal in the domain. Section 3.3 demonstrates that experimental evaluation confirms our theoretical results; and in Section 3.4, we discuss key issues raised in the course of the chapter.

3.1 Single-Observation Recognition

Ramirez and Geffner's probabilistic GR for task-planning is highly principled but computationally expensive. We now show, however, that in the context of path-planning, the

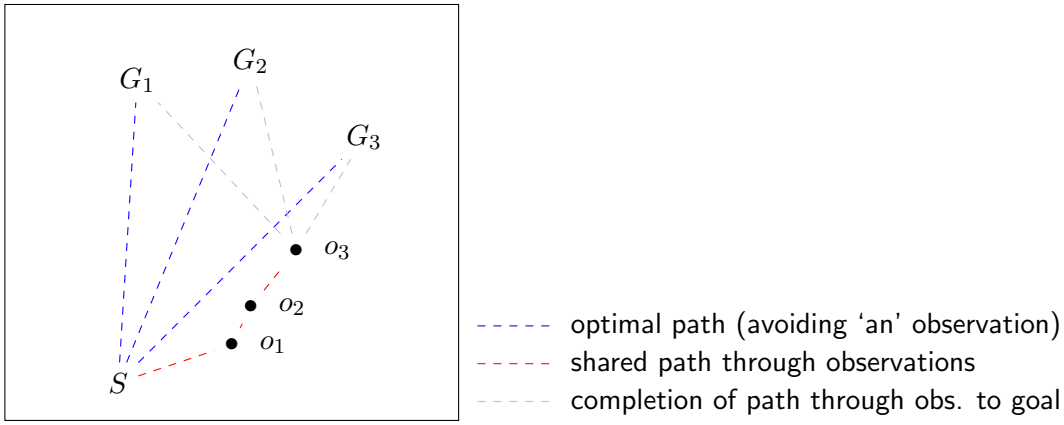


Figure 3.1: The intuition behind single-observation recognition.

Ramirez and Geffner's cost difference formula (2010) takes the cost of an optimal path through the observations and deducts the cost of an optimal path that avoids at least one of them. In a path-planning domain, the formula can be reworked to remove reference to the observation history from both terms, making it much simpler and faster to calculate.

cost difference formula that underlies their framework can be reformulated to make the solution very much simpler and faster to calculate. Single-observation recognition derives directly from Ramirez and Geffner's model. It generates a probability distribution that ranks goals in the same order as their model in all cases bar one but without negative reasoning, without even referencing the observation history of the agent whose goal is of interest and in less than half the time.

Single-observation recognition depends on two intuitions, illustrated in Figure 3.1. The first is as true for task-planning as it is for path-planning. The second is always true for path-planning but applies to task-planning only in certain circumstances (discussed at Section 3.4.2).

1. As others have observed (e.g., Escudero-Martin et al., 2015), since most observed paths are suboptimal, the best (i.e., optimal) path that *differs* from the observed path is usually an optimal path.

2. When comparing optimal paths that run from a starting point, through a sequence of observations, to each of multiple goals, the segment of path from the starting point to the most recent observation is shared and (in a fully observable domain) can be safely ignored.

Used online, single-observation recognition is quick and convenient: it is a one-off operation that returns a probability distribution as needed, so requires no incremental processing of observations. These properties are advantageous in numerous settings. They can benefit the gaming community working on real-time strategy games, for example, where every speed gain is welcome. Moreover, in any real-world application, the fewer the

observations, the less infrastructure required to retrieve and process them. Additionally, being independent of an agent's observation history—and provided possible entry points (e.g., doors to the building) and destinations of interest (e.g., locations to monitor and protect) are known in advance—single-observation recognition makes it possible for probability distributions to be pre-computed offline to generate a sort of 'goal probabilities heatmap', from which they can be retrieved, as needed, in constant time.

Our overarching objective is to illuminate the core problem of GR by placing it in the well-understood context of path-planning: the problem of finding a path from an initial location to a final destination in some map or model of the world. Since Ramirez and Geffner's model was designed for task-plan recognition, we begin by importing it into a path-planning context.

3.1.1 Technical Framework: GR as Path-Planning (discrete)

Ramirez and Geffner's 2010 probabilistic framework for GR (for convenience, hereafter referred to as R&G) operates in a STRIPS domain of fluents and actions. For path-planning in the discrete domain, we express the underlying model (or 'domain theory') as a graph or, in the special case, a grid. Instead of states, comprising multiple fluents, we have atomic nodes or cells; instead of costed actions by which one state may be transformed into another, we have costed edges that enable traversal from node to node.[1]

Definition 1. *A **discrete path-planning domain** is a triple $\mathcal{D}_d = \langle N, E, c \rangle$ where:*

- *N is a non-empty countable set of nodes (or locations);*
- *$E \subseteq N \times N$ is a set of edges between location nodes; and*
- *$c : E \mapsto \mathbb{R}_0^+$ is a function that returns the non-negative cost of traversing each edge.*

A ***path*** π in a domain $\mathcal{D}_d$ is a sequence of node locations (not actions, and not edges) $\pi = n_0, n_1, \ldots, n_k$ such that $(n_i, n_{i+1}) \in E$, for each $i \in \{0, 1, \ldots, k-1\}$. We use π^i to denote the i-th node n_i in π, and $|\pi|$ to denote the length of π, being the total number of edges (k) in π. So, the last location in a path can be referred to as $\pi^{|\pi|}$. Furthermore, we use $\pi(i, j) = \pi^i, \pi^{i+1}, \ldots \pi^j$ to denote the ***subpath*** of π from π^i to π^j (inclusive). The ***cost*** of a path is the cost of traversing all edges in π, that is, $cost(\pi) = \sum_{i=0}^{k-1} c(\pi^i, \pi^{i+1})$. The ***set of all paths*** in $\mathcal{D}_d$ is denoted by $\Pi(\mathcal{D}_d)$ or simply Π, and the set of all paths π starting at $\pi^0 = n_1$ and ending at $\pi^{|\pi|} = n_2$ is denoted by $\Pi(n_1, n_2)$.

Note. Our model of single-observation GR is geared to path-planning and depends on some fundamental distinctions between this and the classical planning model that we are importing. A task-plan is obtained by concatenating actions and an action may

[1] In grids, the contiguity of traversible cells implies an edge between them.

be applicable from many different states: any state where the necessary preconditions pertain. This means that, unless the task-planning domain is fully observable, in order to determine an agent's *current* state, one needs to have tracked each successive action from a known starting point. In a path-planning situation, on the other hand, given a path obtained by concatenating nodes, the final node *is* the agent's current state. No further information or investigation is required.

A path-planning problem adds a start location and a goal to the underlying domain.

Definition 2. *A **discrete path-planning problem** is a tuple $\mathcal{P}_d = \langle \mathcal{D}_d, n_s, n_g \rangle$ where:*

- *$\mathcal{D}_d = \langle N, E, c \rangle$ is the path-planning domain;*
- *$n_s \in N$ is the start location; and*
- *$n_g \in N$ is the goal location.*

As one would expect, the solution to a path-planning problem is a path in its domain $\mathcal{D}_d$ from start location n_s to goal n_g. Technically, a ***solution path*** π is any path π such that $\pi^0 = n_s$ and $\pi^{|\pi|} = n_g$. A path-planning problem is often framed as a search for the shortest path or an ***optimal path***, that is, a solution path with the lowest cost among all solution paths; and we use $\Pi^*(n_s, n_g)$ to denote the ***set of all optimal solution paths***.

Note. In classical task-planning, the starting point is a state (some combination of fluents) quite different from the actions that make up a plan. Here, the start location is a node, just like the nodes at each step in a path. We discuss these distinctions further, and their implications, later in this chapter (see Section 3.4.2, p.84).

In what follows, it will be convenient to specify ***waypoints***: nodes that must be visited. Waypoints are commonly encountered in path-planning and precisely correspond to our proposed treatment of observations. They represent a sequence of observed locations that must be visited in the order given, just as a plan in R&G embeds observations (see p.35).[2] Thus, given a path π and a sequence of waypoints $\vec{w} = (w_0, w_1, \ldots, w_k)$, where $w_i \in N$, we say that π proceeds ***via waypoints*** $\vec{w}$ if there exists a monotonic function $f : \{0, \ldots, k\} \mapsto \{0, \ldots, |\pi|\}$ mapping waypoint indices into path indices in such a way that $\pi^{f(i)} = w_i$. The ***optimal cost via waypoints*** of a path from n_i to n_j via $\vec{w}$—that is, the lowest cost of any path from n_i to n_j via $\vec{w}$—is denoted by $optc(n_i, \vec{w}, n_j)$. When $\vec{w} = \emptyset$ (i.e., no waypoints),[3] we just write $optc(n_i, n_j)$, and when $\pi^0 = w_0$ and $\pi^{|\pi|} = w_k$,

[2] Waypoints are also the mechanism used for interpolation, whereby paths in continuous or grid-based domains may be discretised (e.g., Ferguson & Stentz, 2006).

[3] Though we recognise the abuse of notation, we use $\emptyset$ to represent an empty sequence.

we write $optc(\vec{w})$, that is, the optimal cost through the waypoints themselves. We generalise sets $\Pi(n_s, n_g)$ and $\Pi^*(n_s, n_g)$ to those embedding waypoints $\vec{w}$ as $\Pi(n_s, \vec{w}, n_g)$ and $\Pi^*(n_s, \vec{w}, n_g)$, respectively, since $\Pi(n_s, n_g) = \Pi(n_s, \emptyset, n_g)$ and $\Pi^*(n_s, n_g) = \Pi^*(n_s, \emptyset, n_g)$.

With the basic framework in place, we now formulate the GR problem itself. Whereas in task-planning, the problem is to determine an agent's goal or intent by observing one or more of her actions, in path-planning, we seek to determine the agent's destination by reference to one or more of the locations she has already visited.

Definition 3. *A **GR problem for path-planning in the discrete domain** is a tuple $\mathcal{R}_d = \langle \mathcal{D}_d, G, n_s, \vec{o}, Prob \rangle$, where:*

- *$\mathcal{D}_d = \langle N, E, c \rangle$ is a discrete path-planning domain;*
- *$G \subseteq N$ is the set of possible goal locations;*
- *$n_s \in N$ is the start location;*
- *$\vec{o} = o_1, \ldots, o_k$, $k \geq 1$ and $o_i \in N$ for all $i \in \{1, \ldots, k\}$, is a sequence of observations that is feasible, that is, $optc((n_s, o_1, \ldots, o_k)) \neq \infty$, and such that $o_1 \neq n_s$; and*
- *Prob represents a prior probability distribution across the goals G.*

Effectively, the observation sequence $\vec{o}$ is a partial, discontinuous path from which we generate (or at least cost) a connected path to determine how closely it matches an optimal path towards one of the goals $n_g \in G$. In common with R&G, we assume that the GR problem environment is static and deterministic, that the observation sequence $\vec{o}$ is partial (i.e., that it never extends all the way from start to goal and that it may or may not represent a continuous, connected path) and that it may be noisy (i.e., may conform to an optimal or suboptimal path towards the actual goal).

Note. Since observations are nodes (just like a path or the start location in D_d), we could omit specification of n_s and, instead, use the first observation o_1. Indeed, other authors have taken this approach (e.g., Vered et al., 2016). Our design decision, however, has been to treat n_s as part of the problem domain. While this is consistent with usage in R&G (where the initial state *must* be specified because it is a state, qualitatively different from each observed action) our primary objective is to give the start location a similar status to the possible goals in that, like goals, it is a location likely to be known in advance. For example, it might be a door or a gate or perhaps the location of some CCTV device under which every entrant to the domain must pass.

Our inclusion of n_s as part of the problem definition means, of course, that where

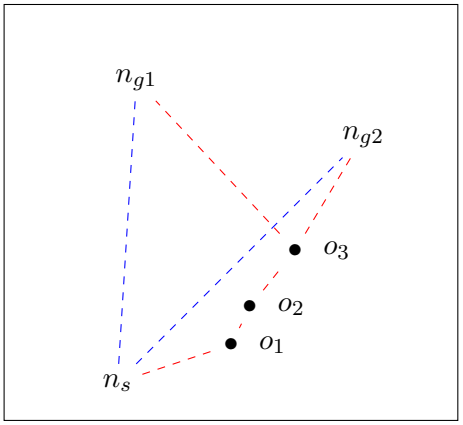


Figure 3.2: Probabilities under the Ramirez and Geffner framework.

Ramirez and Geffner's cost difference formula (2010) takes the cost of an optimal path through the observations and deducts the cost of an optimal path that avoids at least one of them. Their probability distribution formula prefers the goal with the lowest cost difference.

a domain has multiple possible start locations (door1, door2, etc.), use of a different door implies construction of a different problem.

Now, the solution to a GR problem R_d is a posterior probability distribution $P(G \mid \vec{o})$ which prefers those goals the optimal paths to which best satisfy the observations, as determined by the principle of rationality. As a baseline, we obtain the probability distribution using the R&G framework by comparing, for each goal, the difference between the optimal cost of a solution path that embeds the observations $\vec{o}$ (i.e., by treating them as waypoints that must be visited) with the optimal cost of a solution path that does *not* embed them (i.e., because it avoids one or more of them or attains them out of order). The concept is illustrated at Figure 3.2. Formally, the ***baseline cost difference*** is the formula $costdif_{RG} : N \times N \times N^* \mapsto \mathbb{R}$:[4]

$$costdif_{RG}(n_s, n_g, \vec{o}) = optc(n_s, \vec{o}, n_g) - optc^{\neg}(n_s, \vec{o}, n_g), \tag{RG1}$$

where $optc^{\neg}(n_s, \vec{o}, n_g)$ denotes the optimal cost of navigating from location n_s to n_g without embedding all the observations $\vec{o}$, that is:

$$optc^{\neg}(n_s, \vec{o}, n_g) = \min_{\pi \in \Pi(n_s, n_g) \setminus \Pi(n_s, \vec{o}, n_g)} cost(\pi).$$[5]

[4] Formulas (RG1) and (RG2) were informally introduced in the Literature Review (p.37). We have modified the notation to improve legibility. Instead of $cost(G, O)$ to denote the optimal cost of a plan that embeds O, we use $optc(G, O)$ and instead of $cost(G, \overline{O})$ to denote the optimal cost *without* observing O, we use $optc^{\neg}(G, O)$ as, technically, it is not the same function applied to different arguments, but rather a different function to the same arguments.

[5] An unintentional consequence of this construction is that, if it should happen that every path from n_s to n_g embeds the observations $\vec{o}$ (as might happen if every observation were made in a narrow corridor or—as we discuss later—if the only observation occurred at the unavoidable start location), then $\Pi(n_s, n_g) \setminus \Pi(n_s, \vec{o}, n_g) = \emptyset$ (i.e., excluding all such paths leaves an empty set). In this case, $optc^{\neg}(n_s, \vec{o}, n_g) = \infty$ (infinity being the cost of a path that does not exist), which leaves $costdif_{RG}(n_s, n_g, \vec{o})$ undefined.

We generate the probability distribution itself—the solution to an R_d problem—by plugging the cost difference formula into the following template.

$$P_f(n_g \mid \vec{o}) = \alpha \frac{1}{1 + e^{\beta \, costdif_f}}, \tag{RG2}$$

where f identifies the particular cost difference formula in use, α is a normalising constant so probabilities sum to 1 and β is a positive constant.[6] So, in words, substituting RG for f in (RG2), $P_{RG}(n_g \mid \vec{o})$ uses $costdif_{RG}(n_s, n_g, \vec{o})$ to determine the probability that the observed agent is travelling to goal $n_g \in G$, relative to an R_d problem, when the start location is n_s and the observation sequence is $\vec{o}$.

> The lower a goal's cost difference, the higher its probability.

We discuss formula (RG2) in detail in Part II, where we consider its effectiveness in evaluating irrational (and potentially deceptive) paths. For now, simply note that it is provably equivalent to the formulas used for R&G but reformulated in such a way as to make its core property more apparent, that is: the lower the cost difference for a goal, the higher its probability.[7]

For the remainder of this section, we focus on the cost difference formula, assuming an R_d problem of the form $\langle \mathcal{D}_d, G, n_s, \vec{o}, Prob \rangle$, as given by Definition 3. To this end, consider an observational agent using probability distribution $P_{RG}(\cdot)$ to reason about another agent's travel. Using baseline cost difference (RG1), the observational agent must perform two planning tasks for each goal: one to extract $optc(n_s, \vec{o}, n_g)$ and another to extract $optc^{\neg}(n_s, \vec{o}, n_g)$. Both terms demand a full history of observed locations and the second term requires negative reasoning. Furthermore, in a typical application, such as during a real-time strategy game or while conducting surveillance, the computational expense is not incurred once only: the calculations must be repeated (and the time-hit sustained) for every potential goal, every time a new observation is obtained.

To demonstrate our re-working of the solution to improve its efficiency in the path-planning domain, we first eliminate the need for negative reasoning.

3.1.2 GR without Negative Reasoning

As we have seen, path-planners can readily accommodate observation requirements, in a way that task-planners cannot, because observed locations can simply be treated as waypoints: nodes that must be visited. Having a less expressive representation, however, it is not possible to encode negative requirements back into the input of the problem (as done by Ramirez and Geffner (2010) for a STRIPS-like task-planning domain) and still

[6] Though omitted for legibility, values may be multiplied by priors before normalisation. For technical convenience, we abuse notation and take $1/\infty = 0$, whenever $costdif_f = \infty$.

[7] For proof of equivalence and further discussion, see Appendix A.

resolve the problem using a standard path-planner.[8] It is possible to achieve the desired result by calling a path-planner multiple times (making 'avoidpoints' out of first this, then that observation) or by modifying the path-planner (the approach we have taken for experimentation, p.76). However, either method makes the negative reasoning required to calculate optimal cost—whilst simultaneously excluding one or more observations—cumbersome and computationally expensive.

To address this, we adopt an alternative formulation (essentially that used—though not explicitly—by Ramirez and Geffner (2009) to arrive at a *non*-probabilistic solution) whereby, instead of calculating and deducting optimal cost from the start location to each goal 'avoiding at least one of the observations', we simply deduct the more readily available optimal path cost from the start location to each goal. As discussed by Escudero-Martin et al. (2015), this coincides with the intuition that, in the great majority of cases, an optimal path which does not pass through all observed locations *is* an optimal path per se (see Figure 3.3, p.50). Formally, this ***simpler cost difference*** is the formula $costdif_1 : N \times N \times N^* \mapsto \mathbb{R}_0^+$:

$$costdif_1(n_s, n_g, \vec{o}) = optc(n_s, \vec{o}, n_g) - optc(n_s, n_g). \tag{3.1}$$

Note. The second term in this formula deducts optimal cost from n_s to n_g rather than than the cost of avoiding one or more observed waypoints and therefore never returns a value below zero (hence $\mathbb{R}_0^+$). This is not the case for the baseline cost difference formula (RG1), as we discuss shortly.

Formula (3.1) is not only conceptually simpler than (RG1), it is also computationally less demanding, in that there is no need to reason negatively about the observations for the second term. Furthermore, since the cost of an optimal path to each potential goal $n_g \in G$ is not dependent on observations (which typically accumulate), the second term in the formula need only be computed once for each goal. Better still, if the potential start node and all candidate goal locations are known for the path-planning domain itself, as they are in the case of an airport terminal, for example, which has a fixed, finite number of entrances and boarding gates, then all $optc(n_s, n_g)$ terms can be pre-computed and stored for retrieval as needed in constant time.

Interestingly, Ramirez and Geffner explicitly reject this simpler formulation (Ramirez & Geffner, 2010, p.1123) for general task-plan recognition by reference to a particular example (which we review in Section 3.4). For now though, we just want to understand the differences between Equations (RG1) and (3.1) and the likely impact of those differences.

[8] It would be possible (though inefficient) to encode the entire path-planning problem as STRIPS but our objective is to use one of the numerous algorithms (such as Dijkstra's algorithm (Dijkstra, 1959), A* (Hart et al., 1968) and their derivatives) that are optimised for this specialised task.

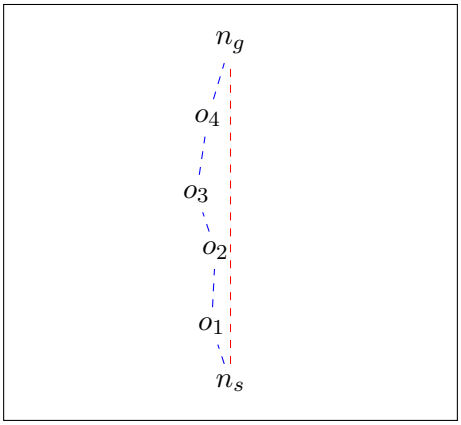


Figure 3.3: Suboptimality. The best path via observations (blue) is suboptimal, so the best path that *avoids* the observations is an optimal (red) path.

In doing so, we demonstrate not only that the simpler formula produces an *identical* result to the baseline formula in all cases bar one; but that, even then, the difference has *minimal impact* on the overall probability distribution across potential goals. Furthermore, in one corner-case, (3.1) actually enables calculation of the posterior probability distribution where the more complex, baseline cost difference (RG1) may not.

Note that all of the cases (1-4) set out below in the context of path-planning are equally applicable in the STRIPS-style task-planning domain described by Ramirez and Geffner (2010).

Case 1: Suboptimal paths. We first consider the situation illustrated in Figure 3.3, where observations conform to a *suboptimal* path, that is, to the observation of an agent whose behaviour is not completely rational (i.e., optimal). In this case, the cheapest available path from n_s to a potential goal $n_g \in G$, given the steps already observed, will inevitably be suboptimal.

Note. The need to accommodate suboptimality in observations was the primary motivation behind the development of the probabilistic R&G framework, as compared to Ramirez and Geffner's previous non-probabilistic model (2009). We argue, in fact, that accommodating observations from agents whose behaviour is not completely optimal is fundamental: in most real-world settings, intelligent agents (including humans) are indeed rational but only to some degree.

Our first result relates to this case. It states that, when the best path possible given the observed behaviour is suboptimal, the simpler formula (3.1) yields *exactly the same* value as the baseline formula (RG1). Recall that the only difference between the two formulas is that, in (3.1), we substitute $optc(n_s, n_g)$ for the second term, $optc^{\neg}(n_s, \vec{o}, n_g)$.

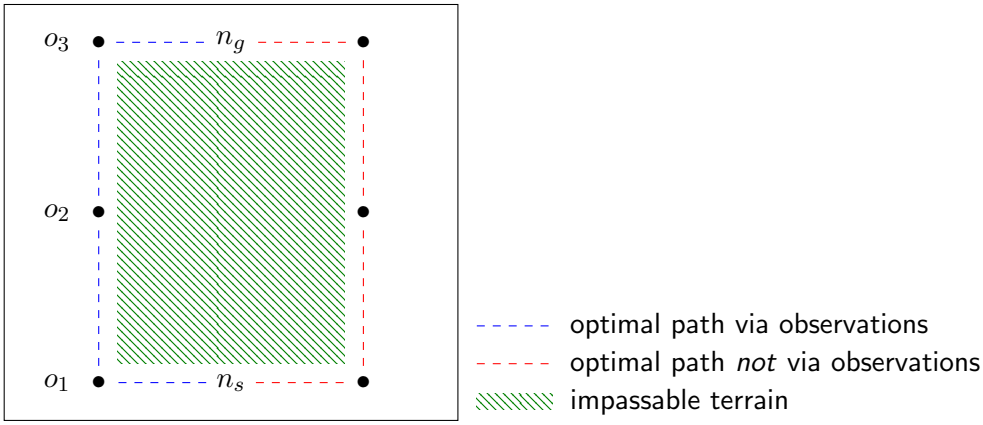


Figure 3.4: Non-exclusive optimality. The path via observations (blue) is optimal; and the best path that avoids the observations (red) is also optimal.

Theorem 1. *Let $\vec{o}$ be an observation sequence such that $optc(n_s, \vec{o}, n_g) > optc(n_s, n_g)$ (i.e., the observed path is suboptimal). Then, $costdif_{RG}(n_s, n_g, \vec{o}) = costdif_1(n_s, n_g, \vec{o})$.*

Proof. Let π^* be an optimal solution path, that is, $\pi^* \in \Pi^*(n_s, n_g)$. Then, by definition, $cost(\pi^*) = optc(n_s, n_g)$. We conclude that path π^* does *not* embed $\vec{o}$, otherwise, we would have $optc(n_s, \vec{o}, n_g) = optc(n_s, n_g)$. Hence, since π^* does not embed $\vec{o}$ and is optimal among all solution paths, we get $optc^{\neg}(n_s, \vec{o}, n_g) = cost(\pi^*) = optc(n_s, n_g)$. Thus, since $optc^{\neg}(n_s, \vec{o}, n_g) = optc(n_s, n_g)$, $costdif_{RG}(n_s, n_g, \vec{o}) = costdif_1(n_s, n_g, \vec{o})$ follows. □

Case 2: Optimal paths (non-exclusive). Let us now consider the case depicted in Figure 3.4 in which observations do conform to an optimal path, but they are *not the only way to behave optimally.*

> **Note.** In path-planning, it is unusual to encounter a solution path, optimal or sub-optimal, whose cost is unique. This is particularly true in a gridworld environment, where there may be thousands of optimal solution paths due to symmetries (Harabor & Grastien, 2012).

When the observed behaviour is optimal but there are multiple optimal paths, not all of which pass through the observations, we have the following result.

Theorem 2. *Let $\vec{o}$ be an observation sequence such that $optc(n_s, \vec{o}, n_g) = optc(n_s, n_g)$ (i.e., the observed path is optimal). If it is the case that $\Pi^*(n_s, n_g) \setminus \Pi^*(n_s, \vec{o}, n_g) \neq \emptyset$, then $costdif_{RG}(n_s, n_g, \vec{o}) = costdif_1(n_s, n_g, \vec{o})$.*

Proof. Take $\pi' \in \Pi^*(n_s, n_g) \setminus \Pi^*(n_s, \vec{o}, n_g)$, that is, path π' is an optimal solution path that does *not* embed $\vec{o}$. Because π' is optimal, $cost(\pi') = optc(n_s, n_g)$, and since it does not embed $\vec{o}$ we can conclude that $optc^{\neg}(n_s, \vec{o}, n_g) = cost(\pi')$. Thus, $optc^{\neg}(n_s, \vec{o}, n_g) = optc(n_s, n_g)$, and $costdif_{RG}(n_s, n_g, \vec{o}) = costdif_1(n_s, n_g, \vec{o})$ follows. □

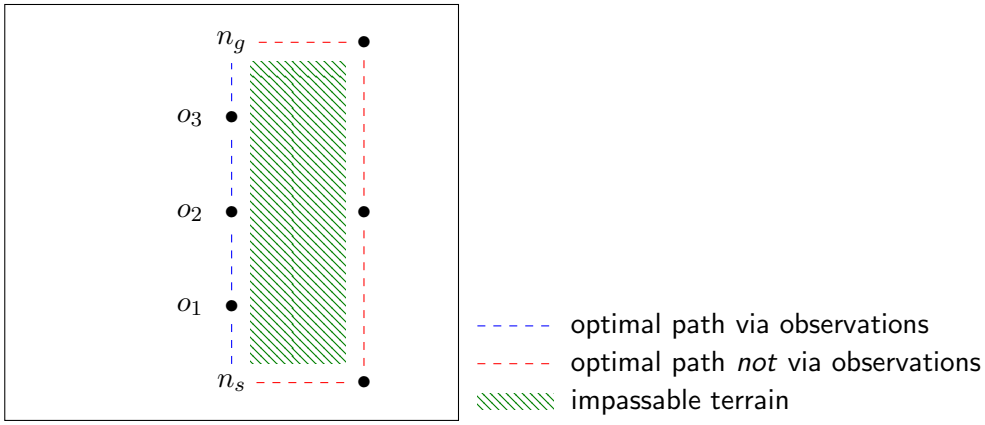


Figure 3.5: Exclusive optimality. The path via observations (blue) is the only optimal path. The best path that avoids the observations (red) is suboptimal.

So, even if the observed behaviour is fully rational, provided that there are other ways of behaving rationally, then the simpler formula (3.1) is, once again, *exactly equivalent* to the baseline formula (RG1).

Case 3a: Optimal paths (exclusive). We now turn to the situation shown in Figure 3.5, which is the only situation in which cost difference Equations (RG1) and (3.1) return different results, that is, when observations are not only sufficient for optimal behaviour, but necessary. In this case, we say the observations are ***exclusively optimal***: the only way of behaving fully rationally involves taking a path that embeds the observations.

Theorem 3. *Let $\vec{o}$ be an observation sequence and $n_g \in G$. Now, $costdif_{RG}(n_s, n_g, \vec{o}) \neq costdif_1(n_s, n_g, \vec{o})$ iff $\Pi^*(n_s, \vec{o}, n_g) = \Pi^*(n_s, n_g)$ (i.e., iff all optimal solution paths embed the observations).*

Proof. The (ONLY-IF) follows directly from Theorem 2 and the fact that $\Pi^*(n_s, \vec{o}, n_g) \subseteq \Pi^*(n_s, n_g)$. For the (IF) direction, suppose that $\Pi^*(n_s, \vec{o}, n_g) = \Pi^*(n_s, n_g)$, that is, all optimal solution paths embed the observations. Take any path $\pi \in \Pi(n_s, n_g)$ that does *not* embed $\vec{o}$, that is, π is a solution path and $\pi \notin \Pi(n_s, \vec{o}, n_g)$. Then, $\pi \notin \Pi^*(n_s, \vec{o}, n_g)$ and since $\Pi^*(n_s, \vec{o}, n_g) = \Pi^*(n_s, n_g)$, $\pi \notin \Pi^*(n_s, n_g)$ follows. Given that $\pi \in \Pi(n_s, n_g)$, we get that $cost(\pi) > optc(n_s, n_g)$. As path π was arbitrarily chosen, $optc^{\neg}(n_s, \vec{o}, n_g) > optc(n_s, n_g)$, and $costdif_{RG}(n_s, n_g, \vec{o}) \neq costdif_1(n_s, n_g, \vec{o})$ follows. □

Now, exclusive optimality—the only case where formulas (RG1) and (3.1) return different results—is a corner case and, arguably, less relevant than the expected suboptimal behaviour that the probabilistic GR framework was designed to handle; but regardless of how relevant or interesting it may be, let us further investigate its implications.

Recall that we are not interested in the result of the cost difference calculation for its own sake, but in order to generate a probability distribution across the set of possible

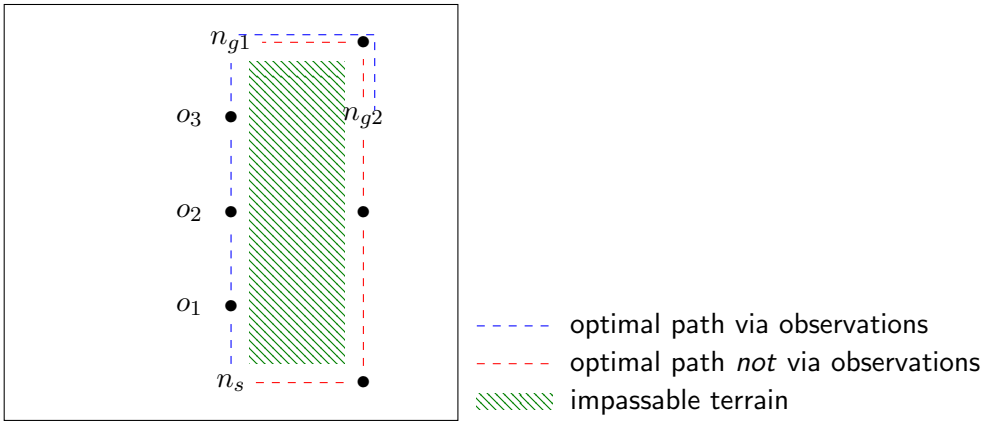


Figure 3.6: Exclusive optimality: rankings unchanged.

The path via observations (blue) is optimal for n_{g1} but suboptimal for n_{g2}. The best path that avoids the observations (red) is suboptimal for n_{g1} but optimal for n_{g2}. Cost difference for n_{g1} is negative using the baseline formula (optimal subtracts suboptimal) but zero using the simple formula (optimal subtracts optimal). Cost difference for goal n_{g2} (for which the best path via the observations is suboptimal) has the same positive value using either formula. Either a negative or zero cost difference for n_{g1} is less than any positive value, so, using either formula, n_{g1} is the most likely goal. That is, rankings are preserved.

goals. Often, we do not need to know exactly how probable each goal is, only their relative order or, more particularly, which goal is most probable.

Case 3b: Optimal paths (exclusive) with rankings unchanged. With this in mind, consider the example depicted in Figure 3.6 which shows (as we prove in Theorem 4) that, in practice, even if an agent is observed taking an exclusively optimal path to goal (i.e., all optimal paths to that goal embed the observations), *unless observations conform to an optimal path for some other goal*, the relative ranking of goals by probability is unaffected by use of the simpler cost difference formula, which still results in successful identification of the most probable goal.

To begin, we make the following auxiliary observation, which formally restates the intuition that the lower the cost difference, the more probable the goal. This follows from the fact that (RG2) shows the relationship between cost difference and probability and is provably equivalent to the account given by Ramirez (2012) (as set out in Appendix A).

Observation 1. *Let* $costdif_f$ *be some (cost difference) function and let* P_f *be the template probability distribution defined in* (RG2)*. If* $costdif_f(n_s, n_{g1}, \vec{o}) < costdif_f(n_s, n_{g2}, \vec{o})$ *then* $P_f(n_{g1} \mid \vec{o}) > P_f(n_{g2} \mid \vec{o})$.

Theorem 4. *Let* $\vec{o}$ *be an observation sequence such that* $\Pi^*(n_s, \vec{o}, n_g) = \Pi^*(n_s, n_g)$ *for some* $n_g \in G$*, that is, the observations are exclusively optimal for potential goal* n_g *(i.e., the case of Theorem 3). Suppose further that* $optc(n_s, \vec{o}, n_{g'}) > optc(n_s, n_{g'})$*, for every* $n_{g'} \in G \setminus \{n_g\}$*, that is, observations would result in suboptimal paths to all the other possible goals. Then, for all distinct goals* $n_{g1}, n_{g2} \in G$*, it is the case that* $P_1(n_{g1} \mid \vec{o}) > P_1(n_{g2} \mid \vec{o})$ *if and only if* $P_{RG}(n_{g1} \mid \vec{o}) > P_{RG}(n_{g2} \mid \vec{o})$.

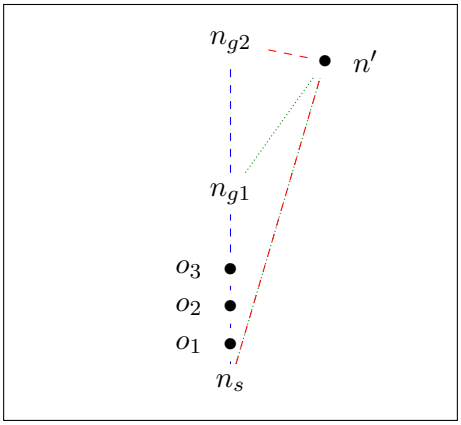


Figure 3.7: Exclusive optimality: rankings changed.

The path via observations (blue) is optimal for both goals; and the best path that avoids the observations (green for n_{g1}, red for n_{g2}) is suboptimal for both goals: it has to go via n'. Complex cost difference is negative for both goals but results in a lower cost difference (higher probability) for n_{g1}.

Proof. Take any $n_{g'} \in G \setminus \{n_g\}$ (i.e., the suboptimal case). From Theorem 1, we know that $costdif_{RG}(n_s, n_{g'}, \vec{o}) = costdif_1(n_s, n_{g'}, \vec{o})$ and $optc^{\neg}(n_s, \vec{o}, n_{g'}) = optc(n_s, n_{g'})$. Since $optc(n_s, \vec{o}, n_{g'}) > optc(n_s, n_{g'})$, using Equation (3.1) we conclude that:

$$costdif_{RG}(n_s, n_{g'}, \vec{o}) = costdif_1(n_s, n_{g'}, \vec{o}) > 0.$$

So, for all goals different from n_g, cost difference values are the same and greater than zero, whether using the baseline or the simpler formula.

It remains to verify the ranking of goal n_g. Since $\Pi^*(n_s, \vec{o}, n_g) = \Pi^*(n_s, n_g)$, it must be that $\vec{o}$ is necessary to travel from n_s to n_g in any optimal way. Therefore, any route that does not embed $\vec{o}$ is suboptimal. Formally, $optc^{\neg}(n_s, \vec{o}, n_g) > optc(n_s, n_g) = optc(n_s, \vec{o}, n_g)$. Using this in Equation (RG1) we get that $costdif_{RG}(n_s, n_g, \vec{o}) < 0$. In turn, from Equation (3.1), we get that $costdif_1(n_s, n_g, \vec{o}) = 0$. Thus, for all $n_{g'} \in G \setminus \{n_g\}$, we conclude that: (a) $costdif_1(n_s, n_g, \vec{o}) = 0 < costdif_1(n_s, n_{g'}, \vec{o})$; and (b) $costdif_{RG}(n_s, n_g, \vec{o}) < 0 < costdif_{RG}(n_s, n_{g'}, \vec{o})$.

Putting it all together, both cost difference accounts rank all goals in G equivalently. For all $n_{g1} \neq n_{g2} \in G$, $costdif_{RG}(n_s, \vec{o}, n_{g1}) < costdif_{RG}(n_s, n_{g2}, \vec{o})$ iff $costdif_1(n_s, n_{g1}, \vec{o}) < costdif_1(n_s, n_{g2}, \vec{o})$. Thus, using Observation 1, the theorem follows. □

Theorem 4 shows that, even in this corner case of exclusive optimality, the simpler cost difference formula (3.1) lets us determine the same ranking among potential goals as (RG1) and is therefore sufficient to identify the most probable goal. There is, however, one variation of exclusive optimality where rankings do differ. We address this next.

Case 3c: Optimal paths (exclusive) where rankings change. The only case of exclusive optimality for which Theorem 4 does not apply occurs when observations coincide with the optimal path to *multiple* goals (rather than one) and is the *only* optimal path to

at least one of them. This can arise, for example, if two goals are aligned sequentially, as shown in Figure 3.7. In this case, the baseline formula (RG1) returns multiple (negative) cost differences to yield a ranking across the goals, whereas the simpler formula (3.1) ranks all goals for which observations match their optimal paths equally (the result obtained in Ramirez and Geffner's 2009 account and, arguably, the outcome that one would expect).

That said, there are realistic route-planning scenarios where this situation does arise, which we discuss at Section 3.4. For now, we point out that this is, nevertheless, a corner case and one that concerns the very set of goals in which the probabilistic account (Ramirez & Geffner, 2010) is, arguably, least interested (since the 2009 framework already accommodated cases where observations matched with optimal paths).

Case 4: A single solution path. Finally, in the extreme case, where the only path in the domain from n_s to n_g is one that passes through all the observations, the cost of a path that does not conform to observations is infinite (because no such path exists). In this case, as Ramirez and Geffner (2010, p.1123) themselves point out—and as flagged as an unintended consequence of the baseline formula at Footnote 3.1.1 (p.47)—Equation (RG1) ought to return $-\infty$ giving n_g the highest possible probability within the distribution. However, since $-\infty$ is not a number, the result may be undefined with the flow-on effect that normalised scores for the rest of the distribution may also be undefined. In any practical implementation, of course, the problem can easily be rectified by allocating some minimum value instead of $-\infty$ or treating this case separately. In an identical situation, however, the simpler cost difference equation (3.1), based on optimal cost from start to goal (rather than 'optimal cost given not the observations') returns zero and the issue does not arise.

Having eliminated the need for negative reasoning from the cost difference formula, we next eliminate the need to track a succession of multiple observations.

3.1.3 GR without the Observation Sequence

In this subsection, we prove that, given a known starting point, the ranking among potential goals, as judged by probability distribution formula $P_1(\cdot)$—already shown to return the same result as the baseline formula $P_{RG}(\cdot)$ in all cases bar one—can be achieved without needing to know anything about the agent other than where she is 'now'.

The claim seems surprising; it implies that we can perform GR without knowing how the agent behaves over time. Nevertheless—as we show in Theorem 5—under the R&G model in the context of path-planning, probability rankings at any point in the domain can be predicted and remain unchanged, regardless of the path taken to get there.

The following ***single-observation cost difference*** formula dispenses with both negative reasoning *and* the observation history. Formally, $costdif_2 : N \times N \times N \mapsto \mathbb{R}$ is

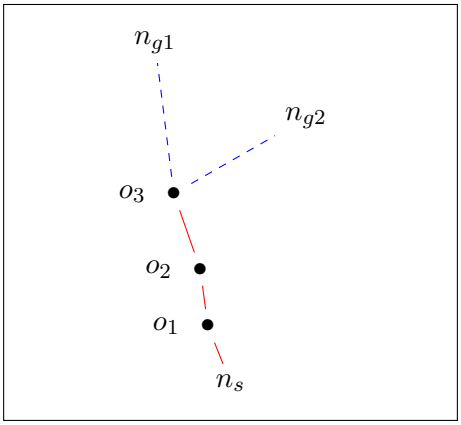


Figure 3.8: single-observation: the optimal cost of a path from n_s through $\vec{o}$ is the same for all goals.

defined as follows:

$$costdif_2(n_s, n_g, n) = optc(n, n_g) - optc(n_s, n_g). \tag{3.2}$$

Note. In the single-observation formula, n typically stands for the current or most recently observed location of the agent whose destination we are trying to determine (i.e., $n = O^{|O|}$). So, plugging this formula into the probability function (RG2), it can be used to answer the question: if an agent were observed at this node (which might be any traversible node in the domain) what is the likelihood of each goal being her destination?

Recall that $P_2(\cdot)$ denotes the templated probability function (RG2) when used with the single-observation formula $costdif_2$ above (3.2), while $P_1(\cdot)$ uses the simpler cost difference formula $costdif_1$ (3.1). We now show that, using either formula, the goal rankings are the same. The intuition behind the theorem is straightforward and is depicted in Figure 3.8: the optimal cost of a path from n_s through $\vec{o}$ is the same for all goals and can be ignored.

Theorem 5. *Let $\vec{o}$ be an observation sequence where $\vec{o}^{|\vec{o}|} = n$ (i.e., n is the last observation in $\vec{o}$). Then, for all $n_{g1}, n_{g2} \in G$, it is the case that $P_1(n_{g1} \mid \vec{o}) > P_1(n_{g2} \mid \vec{o})$ iff $P_2(n_{g1} \mid n) > P_2(n_{g2} \mid n)$.*

Proof. From Observation 1, recall that $P_1(n_{g1} \mid \vec{o}) > P_1(n_{g2} \mid \vec{o})$ iff $costdif_1(n_s, n_{g1}, \vec{o}) < costdif_1(n_s, n_{g2}, \vec{o})$, that is, the relative ranking between n_{g1} and n_{g2} with respect to their posterior probabilities can be deduced directly from the relative values of their cost difference formulas. Recall also, from formula (3.1), that for each $i \in \{1, 2\}$:

$$costdif_1(n_s, n_{gi}, \vec{o}) = optc(n_s, \vec{o}, n_{gi}) - optc(n_s, n_{gi}),$$

where the first term (the optimal cost from the starting node, through the observations, to the goal) can be written as:

$$optc(n_s, \vec{o}, n_{gi}) = optc(n_s, \vec{o}^{\,0}) + optc(\vec{o}) + optc(\vec{o}^{\,|\vec{o}|}, n_{gi}).$$

Comparing cost differences, we get:

$$\begin{aligned}
&costdif_1(n_s, n_{g1}, \vec{o}) - costdif_1(n_s, n_{g2}, \vec{o}) \\
&\quad = [optc(n_s, \vec{o}^{\,0}) + optc(\vec{o}) + optc(\vec{o}^{\,|\vec{o}|}, n_{g1}) - optc(n_s, n_{g1})] - \\
&\qquad [optc(n_s, \vec{o}^{\,0}) + optc(\vec{o}) + optc(\vec{o}^{\,|\vec{o}|}, n_{g2}) - optc(n_s, n_{g2})] \\
&\quad = [optc(n_s, \vec{o}^{\,0}) + optc(\vec{o}) + optc(n, n_{g1}) - optc(n_s, n_{g1})] - \\
&\qquad [optc(n_s, \vec{o}^{\,0}) + optc(\vec{o}) + optc(n, n_{g2}) - optc(n_s, n_{g2})] \\
&\quad = optc(n_s, \vec{o}^{\,0} + optc(\vec{o})) + optc(n, n_{g1}) - optc(n_s, n_{g1}) - \\
&\qquad optc(n_s, \vec{o}^{\,0}) - optc(\vec{o}) - optc(n, n_{g2}) + optc(n_s, n_{g2}) \\
&\quad = optc(n, n_{g1} - optc(n_s, n_{g1})) - optc(n, n_{g2}) + optc(n_s, n_{g2}) \\
&\quad = costdif_2(n_s, n_{g1}, n) - costdif_2(n_s, n_{g2}, n).
\end{aligned}$$

Thus, $costdif_1(n_s, n_{g1}, \vec{o}) > costdif_1(n_s, n_{g2}, \vec{o})$ iff $costdif_2(n_s, n_{g1}, n) > costdif_2(n_s, n_{g2}, n)$. So, applying Observation 1, $P_1(n_{g1} \mid \vec{o}) > P_1(n_{g2} \mid \vec{o})$ iff $P_2(n_{g1} \mid n) > P_2(n_{g2} \mid n)$. □

The finding is useful and unexpected. It tells us that we can achieve the same recognition, at the qualitative level, by considering only the 'last' observation (rather than a complete observation sequence). This is important because it allows us to judge every location node *as if* an agent were observed there and calculate the likelihood of each goal being their destination, regardless of how they arrived at that node. Furthermore, since we are now dealing with individual nodes, recognition can be achieved by calls to any standard path-planner: no specialised path-finding system is needed to reason negatively or, indeed, to reason about observations at all. Finally, if start and all candidate goal locations are known—as would typically be the case in most path-planning domains—formula $costdif_2(n_s, n_g, n)$ can be fully *pre-computed offline* for any node $n \in N$ in the domain.

There are significant implications. Not only can we perform *online* goal recognition without having to track the agent's movements (and therefore without incremental reasoning) but, as an *offline* strategy, we can create a probabilistic heatmap of the domain (see Figure 3.9), showing the probability of each goal at each (or any) location, according to where the agent entered. Armed with such a heatmap, there are two obvious uses.

1. If we have a particular goal of interest (e.g., a valuable location to monitor and protect), we can focus our attention fully on locations where that goal is the most probable. Rather than tracking an agent's movements all over the terrain, we can just monitor the high-probability 'hot-spots' revealed by the heatmap and only start tracking if the agent arrives at one of them.

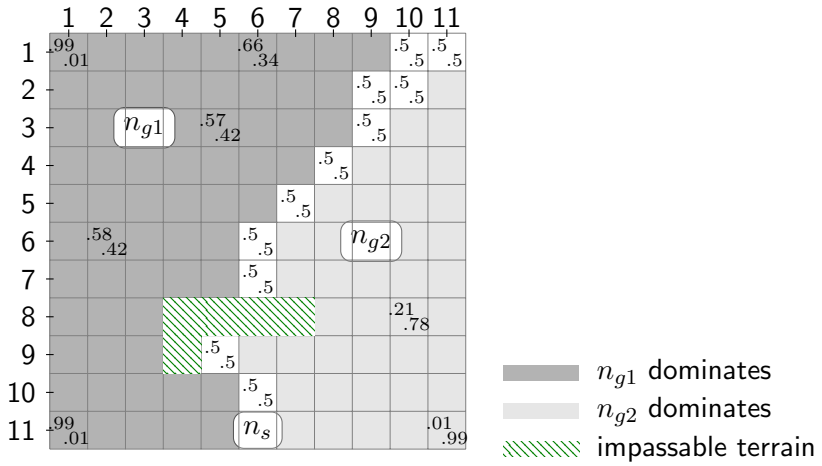


Figure 3.9: Heatmap. Probabilities depend on location and can be pre-calculated if the start location (n_s) is known. Probability values are shown at selected locations, given first for n_{g1}, then for n_{g2}.†

†For legibility, only two goals are shown and probabilities are detailed only at selected locations, however $costdif_2(\cdot)$ can be substituted for $costdif_{RG}(\cdot)$ in Equation (RG2) to generate a complete probability distribution for any number of goals at any location.

2. Having developed the heatmap offline, we can put it to use as a tool online. If we identify an agent of interest 'on-the-fly', we can use the heatmap as a lookup table to retrieve her most likely destination in constant time.

Note. The above result is not dependent on formula (RG2) itself. Rather, it is applicable whatever manipulation is used to derive the probability distribution, provided that the posterior probability function satisfies the property that the lower the cost difference, the higher the probability and relative cost differences are preserved (see Observation 1).

Single-observation recognition is a method of GR that avoids negative reasoning and has no need to reference incremental observations. We conclude this section by considering the rationale behind the single-observation formula and examining its properties.

3.1.4 The Rationale behind Single-Observation Recognition

We have shown (by Theorems 1 through 5) that, under the R&G model for probabilistic GR as specialised to path-planning, unless an agent must pass through every observed location to remain on an optimal path (i.e., the special case of exclusive optimality excepted by Theorem 3), provided that we know where she started from, the path she takes has no effect whatsoever on the probability rankings of her possible destinations.

Probabilities depend on an agent's location relative to the starting point and the possible goals. The observed path is, effectively, redundant. Even on reflection, the finding is counter-intuitive: how is it that we can predict where an agent is going without needing to know where she has been? On careful consideration, however, the Markovian nature of the result is not quite so unlikely as it seems. It has three sound bases.

Probabilities depend on the agent's location relative to the starting point and the goals. The observed path is redundant.

1. Observations, in this domain, are not action sequences (as they are in Ramirez and Geffner's account of GR); they are fully observable states encompassing everything we need to know about the agent's condition, that being her location (effectively a contraction of the action $go(x)$).

2. The agent's starting location is similarly fully observable and known, supplied as input to the GR problem. Thus, given an agent's current location (and given that, in this environment, location implies full observability), we are implicitly aware of everything relevant that has *changed* since the agent entered the domain. Hence, the optimal path to her current location is also (implicitly) given.

3. Path-segments cannot be arbitrarily reversed or 'executed simultaneously' as may occur in a task-planning environment. In path-planning, observations must have been traversed in the seen order and this, taken with the above, allows the 'optimal cost through the observations' to be separated into terms that cancel out when taking relative differences.

The effect of breaking these conditions is discussed further in Section 3.4. Provided they are met, however (as they always are in the core path-planning domains with which we are concerned), probability distribution $P_2(\cdot)$, which uses the single-observation formula (3.2), yields exactly the same probability rankings across goals as $P_1(\cdot)$, which uses the simpler cost difference (3.1). We can, therefore, make certain specific claims for single-observation recognition by comparison with the more complex baseline formulation; and these are set out in Tables 3.1 and 3.2.

Goal types are defined according to the cases presented in Section 3.1.2. So, referring to Table 3.1, goals such that observations conform to suboptimal paths have the same probability values whether probability distribution formula (RG2) is used with the simple or baseline cost difference formula and the same probability rankings whichever of the three cost difference formulas is used.

Table 3.2 compares the complexity of the formulas based on the reasoning required to calculate a probability distribution. We see that $P_2(\cdot)$ is the most computationally advantageous. It does not need to reason negatively about observations; neither does

Table 3.1: Comparison of probability distributions by goal type.

Goal Type†	P_1 **v** P_{RG}	P_2 **v** P_{RG}
Suboptimal	same	same rankings
Non-exclusively optimal	same	same rankings
Exclusively optimal (1 optimal only)	same rankings	same rankings
Exclusively optimal (multiple optimal)	optimals unranked	optimals unranked

†Goal types by case, as defined in Section 3.1.2.

Table 3.2: Comparison of probability distributions by computational complexity.

Computational Expense	P_{RG}	P_1	P_2
Reasons about observations	Yes	Yes	No
Reasons negatively	Yes	No	No
Planning calls per goal	2	1††	1

††Plus subgoaling between waypoints (see commentary, inline).

it need to reason about observations. Assuming optimal path costs from the starting location to each goal are known or pre-computed, it requires just one optimal path from point to point per goal compared with $P_{RG}(\cdot)$ which requires two or $P_1(\cdot)$ which, again only requires one (once $optc(n_s, G)$ is known) but which must also take observations into account.

To elaborate more formally, calculation of $P_{RG}(\cdot)$ requires $2|G|$ calls to the path-planner, whereas $P_1(\cdot)$—and by extension $P_2(\cdot)$—require only $|G|$ calls. In practice, the time-saving is more emphatic. Time taken is more than halved using either $P_1(\cdot)$ or $P_2(\cdot)$ because the call to the planner that is *not* now needed is the one that required negative reasoning (i.e., the more expensive call of the two). Furthermore, in practice, $P_2(\cdot)$ is faster than $P_1(\cdot)$ because it does not necessitate finding a path through waypoints. This means that it can be solved by an even more rudimentary path-planning algorithm than can be used for $P_1(\cdot)$. An algorithm for $P_1(\cdot)$ must plan an optimal path that involves subgoaling from waypoint to waypoint; and that, depending on their distance apart and the intricacy of the underlying terrain, can take considerably longer than finding a single optimal path from point to point.[9]

Note. The above discussion highlights points of *comparison* between GR using the single-observation formula (3.2) and the original, more complex formula (RG1). However, our account also inherits some limitations. In particular, we assume that the observed agent is rational. Thus, although it could be argued that an agent who directly advances towards a gate should be assigned a higher probability with respect

[9]In the worst case, path-planning through $|\vec{o}|$ waypoints involves $|\vec{o}|$ path-planning tasks, using subgoaling.

to that gate than one that has, for example, zigzagged between many other gates on the way, $P_2(\cdot)$, in common with $P_{RG}(\cdot)$, is unable to achieve this.

We seek to resolve this major limitation arising from the rationality assumption when we tackle deceptive path-planning in Part II (see p.96).

In this section, we demonstrated a method of GR for path-planning domains that is substantially less demanding than the previous state-of-the-art. Using this method, we can generate a probability distribution for any locaton in the domain, based on an agent's starting point. In the next section, we take advantage of this capability to develop a formula for finding the distance from a goal of interest within which that goal becomes the most probable; and we do this, not only without negative reasoning and independently of observations, but without even calculating any probabilities.

3.2 The Radius of Maximum Probability

The radius of maximum probability (RMP) is a distance from goal within which that goal can be *guaranteed* to be an agent's most probable destination. In this section, we demonstrate that it can be calculated from costs typically available in a known domain, namely the optimal costs between a set of possible goals and the start location.

Observe that this notion complements single-observation goal recognition rather than being a substitute for it. Whereas the single-observation formula calculates a full probability distribution for a set of goals at any one location, the RMP (based on a similar set of parameters) is a single measurement, a radius within which any one particular goal is the most probable; and that radius might encompass a significant number of nodes or (in a continuous environment) infinitely many points. So, there is a trade-off between probability values for many goals at any one point or a determination that one goal is the most probable at (perhaps) many points. Also, it is important to state right away that, although the RMP guarantees a goal has maximum probability within the calculated radius it says *nothing* about probabilities outside that radius.

Nevertheless, the RMP is a potentially powerful tool. In the real world, it could be used offline in various ways, for example, as an adjunct to goal recognition design applications (Keren et al., 2014), to help decide where a security detail should be positioned, the area it should patrol or ideal locations for surveillance cameras. Alternatively, since it is quickly calculable, it could be used online to find RMPs in dynamic domains where calculation of probabilities point by point would take too long, such as games that automatically generate new (previously unknown) terrains (e.g., Raffe et al., 2012).

The RMP is a real number, best demonstrated in terms of continuous measurement from point to point. Therefore, we now extend our technical framework to a continuous setting.

3.2.1 Technical Framework: GR as Path-Planning (continuous)

Transposing the GR problem from a graph-based domain to a continuous environment resembling the real world is, on the one hand, more familiar and relateable but, on the other, it is theoretically more demanding, seeming to grapple with Zeno's paradox at every turn.[10]

We are interested in continuous two-dimensional planes (consistent with traditional maps or groundplans) and three-dimensional settings (consistent with our real-world experience of navigation). In both cases, movement from one location to another necessarily involves traversal through an infinitely divisible sequence of points in between. In place of a finite number of nodes and edges, there are now an infinite number of points capable of becoming path-connected whenever reachable from one another through a connected region of traversable space.

Our continuous account occurs in the context of *metric* spaces, which are (standard) topological spaces that define connectedness but also distance (LaValle, 2006).[11] Technically, a ***metric space*** (X, d) amounts to a set X (typically points or locations) coupled with a *distance* metric d, which conforms to the following axioms: (i) $d(x, y) \geq 0$ for all $x, y \in X$ (non-negativity); (ii) $d(x, y) = 0$ iff $x = y$ (identity of indiscernibles); (iii) $d(x, y) = d(y, x)$ (symmetry); and (iv) $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality).

That in hand, we can restate Definition 1 (the discrete path-planning domain) for the continuous case.

Definition 4. *A **continuous path-planning domain** is a tuple $\mathcal{D}_c = \langle X, d, Obst \rangle$ such that:*

- *(X, d) is a metric space, where $X = \mathbb{R}^n$ (for some $n \in \{2, 3\}$) and $d : X \times X \mapsto \mathbb{R}^+$ is a Euclidean metric, that is, the (non-negative) straight-line distance from point to point (i.e., $d(x, y) = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2}$; and*
- *$Obst \subset X$ is a set of obstacles in the space (e.g., walls and other barriers).*

In order to define a path-planning problem in $\mathcal{D}_c$, we need to know exactly where in the domain our agent can go. Following LaValle (2006), we adopt the representation of a moveable, embodied ***agent*** $A(q) \subset X$, where $q = (x, y)$ or $q = (x, y, z)$ and $A(q) \subset X$ denotes all those points occupied when A is at q.[12] Without loss of generality, we assume

[10] Zeno is the Greek philosopher (c.490–430 BC) famous for the paradox whereby one is never able to arrive at one's destination because, in order to get there, one must always travel half of the way, which means there is always half way still to go.

[11] The ***standard topology on*** $\mathbb{R}^n$ is the topology induced by a Euclidean metric (i.e., Pythagorean distance) on the set of real numbers. It is referenced as 'standard' because it conforms to our understanding of connectedness and continuity in the real world.

[12] If the agent were not embodied, either cost difference formula (3.3) loses meaning (as Kaminka et al., 2018, remark) or the formula collapses to its limit and fails to capture its intended meaning—because the optimal cost of complying with the observations and *not* complying could be the same! In any

that A has only one configuration; therefore, the space it occupies can be identified from its location $q \in X$ (as in the case of a solid vehicle or the circular overhead view of a pedestrian).[13] We define ***obstacles to movement*** X_{obst} as all those points where the presence of the agent would intersect with an obstacle, that is, $X_{obst} = \{q \in X | A(q) \cap Obst \neq \emptyset\}$. Intuitively, the remaining ***traversable space*** X_{free} is the complement of X_{obst} but, as we want to calculate shortest paths (and potentially make contact with the edges of obstacles), we define X_{free} as the *closure* of $\{X \setminus X_{obst}\}$ (i.e., $\{X \setminus X_{obst}\}$ plus its boundary points).[14]

Definition 5. *A **continuous path-planning problem** is a tuple $\mathcal{P}_c = \langle \mathcal{D}_c, A, x_s, x_g \rangle$, where $\mathcal{D}_c$ is a continuous path-planning domain, $A(\cdot) \subset X$ is the agent, $x_s \in X_{free}$ is the starting point and $x_g \in X_{free}$ is the destination.*

Given that the number of points in a continuous path is infinite, it is not possible to define a path as just an enumerated sequence of points. Whereas a sequence of nodes in a discrete domain may be regarded as a function such that $\pi(i)$ or π^i denotes the i-th node in the path, a ***connected path*** in the problem domain $\mathcal{P}_c$ is a continuous function $\pi : [0, 1] \mapsto X_{free}$. Intuitively, the domain $[0, 1]$ (of a path function) represents normalised distance along the path as if it had been straightened out like a piece of string; its image is the path in space. So $\pi(0)$ is the path's starting point and $\pi(1)$ its endpoint; and every point in the path can be referenced by $\pi(i)$ for some $i \in [0, 1]$.

Note. The notion of connected space is built into a path's definition: if $\pi(0)$ and $\pi(1)$ were points in disjoint open subsets of X_{free}, then the function π could not be continuous.

To reference a ***subpath***, given an interval $[i, j]$ in $[0, 1]$, the segment of path π from $\pi(i)$ to $\pi(j)$, denoted $\pi_{[i,j]}$, is the normalised restriction of π to the interval in question:

$$\pi_{[i,j]}(k) = \pi(i + k \cdot (j - i)), \text{ for all } k \in [0, 1].$$

Notice that $\pi_{[i,j]}$ is in itself a path (i.e., its domain is $[0, 1]$). As before, we use $\Pi(\mathcal{D}_c)$ or (where the domain is understood) Π, to denote the set of *all* paths $\mathcal{D}_c$.

To define the ***length of a path***, we see the path as partitioned into equal segments and sum their lengths as the number of segments approaches infinity. Formally, the length of a path π with respect to a Euclidean metric d is defined as:

practical application, an agent is embodied and capable of reaching its destination. Therefore, we follow LaValle (2006) and give her substance.

[13] The notion of multiple configurations (e.g., to allow for arm movement) has no impact in our domain.

[14] Closed and open sets are complementary notions: sets, respectively, that do or do not include their boundary points. 'Closure' of an open set is that open set *plus* its boundary points. Since *Obst* and $A(q)$ are closed sets in X, $X \setminus X_{obst}$ is open. To include the boundary points, we take its closure (LaValle, 2006, p.128-9).

$$L(\pi) = \lim_{N\to\infty} \sum_{i=1}^{N} d(\pi\Big(\frac{i}{N}\Big), \pi\Big(\frac{i+1}{N}\Big)).$$

A path π is a ***solution path*** for $\mathcal{P}_c$ if $\pi(0) = x_s$ and $\pi(1) = x_g$ and, similar to previous notation, we use $\Pi(x_s, x_g)$ to denote the set of *all* solution paths for $\mathcal{P}_c$.

The notion of the *path cost* and, as a consequence, that of *optimal* paths, requires some additional explanation when it comes to continuous domains. In a discrete graph-based domain, a cost is associated with every edge and path cost simply involves summing the costs of all traversed edges. In a continuous environment, however, there is no finite set of edges. In this thesis, we adopt the simple and often applicable model of cost as distance, that is, we let $cost(\pi) = L(\pi)$.[15] Thus, given a continuous path-planning problem $\mathcal{P}_c$, an ***optimal*** solution $\pi^*(x_s, x_g)$ is a solution path of minimum *length*, that is:

$$\pi^*(x_s, x_g) \in \underset{\{\pi|\pi\in\Pi(x_s,x_g)\}}{\operatorname{argmin}} L(\pi).$$

Note. Since the agent is embodied and X_{free} is closed, the agent can come into contact with obstacles without intersecting with them. Otherwise, a minimum length path would not be calculable (LaValle, 2006, p.156).

The final major adjustment with respect to our previous discrete account is in the treatment of observations. Here, we follow Vered et al. (2016)[16] in making each observed point (or trajectory) a function of the time interval during which it was observed. Formally, given a total time interval $[0, T_m]$ during which all observations were made, we define an ***observation model*** for a continuous domain $\mathcal{D}_c$ as a pair $\mathcal{O} = \langle T_o, o\rangle$ where:

1. $T_o \subseteq \mathbb{R}_0^+ \times \mathbb{R}_0^+$ is a finite set of non-intersecting (closed) time intervals, that is, (i) if $[t_1, t_2] \in T_o$, then $t_2 \geq t_1$ and $t_2 < T_m$; and (ii) if $[t_1, t_2], [t_3, t_4] \in T_o$ and $[t_1, t_2] \neq [t_3, t_4]$, then either $t_3 > t_2$ or $t_1 > t_4$; and

2. $o : T_o \mapsto \Pi(\mathcal{D}_c)$ is an observation function which denotes the path observed during each observation interval.

In words, T_o represents a set of all the intervals during which the agent has been observed and $o([t_1, t_2])$ yields a path that represents continuous observation during each of those

[15] Alternative cost models tend to be either: (i) linear to distance anyway (e.g., the sum of distances multiplied by the cost of traversal through each particular terrain type such as ground, water, swamp, etc.); (ii) necessitate discretising the domain back to a graph or grid (e.g., road networks for SatNavs or sampled spaces for Rapidly-exploring Random Trees). In more sophisticated accounts, such as motion-planning for articulated robots, optimality is often more to do with optimisation than cost (i.e., to reconcile distance, angle of movement, number of moving parts, etc.); or the complexity of the problem may be such that *any* realisable solution suffices and optimality is not even considered (LaValle, 2006).

[16] We have modified the notation to more completely express the intervals represented in T_o.

intervals, that is, each time period $[t_1, t_2] \in T_o$. Since the domain of a continuous path function is $[0, 1]$, $o([t_1, t_2])(0)$ denotes the first observed location (observed at time t_1), whereas $o([t_1, t_2])(1)$ stands for the last observation (observed at time t_2).

Using this account, we identify ***all paths that embed the observations*** as the set $\Pi(\mathcal{O})$ such that $\pi \in \Pi(\mathcal{O})$ iff there exists a mapping m from the observations into the path, $m : T_o \mapsto [0, 1] \times [0, 1]$ where: (i) $o([t_1, t_2]) = \pi_{m([t_1,t_2])}$ (i.e., every observed point or trajectory occurs somewhere in π); and (ii) for all $\vec{t}, \vec{t'} \in T_o$, $\vec{t} < \vec{t'}$ iff $m(\vec{t}) < m(\vec{t'})$. In words, every path in $\Pi(\mathcal{O})$ includes all subpaths observed during the time interval $[0, T_m]$ in the same order that they were observed.

With the set $\Pi(\mathcal{O})$ in hand, we can precisely state the optimal cost from point x_1 to point x_2 that embeds the observations $\mathcal{O}$ as follows:[17]

$$optc(x_1, \mathcal{O}, x_2) = \min_{\{\pi|\ \pi \in \Pi(\mathcal{O}), \pi(0)=x_1, \pi(1)=x_2\}} cost(\pi).$$

Similarly, the optimal cost *not* embedding the observations (i.e., because the path avoids at least one of them or they occur out of order) can be defined as follows:

$$optc^{\neg}(x_1, \mathcal{O}, x_2) = \min_{\{\pi|\ \pi \in \Pi(\mathcal{D}_c) \setminus \Pi(\mathcal{O}), \pi(0)=x_1, \pi(1)=x_2\}} cost(\pi).$$

With all the technical machinery in place, we now turn our attention to the GR problem itself and, more importantly, to its solution concept.

Definition 6. *A **GR problem for path-planning in the continuous domain** is a tuple $\mathcal{R}_c = \langle \mathcal{D}_c, A, X_g, x_s, T_m, \mathcal{O}, Prob \rangle$, where:*

- $\mathcal{D}_c = \langle X, Obs, d \rangle$ *is a continuous path-planning domain;*
- $A(\cdot) \subset X$ *is a mobile, embodied agent;*
- $X_g \subset X_{free}$ *is a finite set of points denoting all candidate goal locations;*
- $x_s \in X_{free}$ *is the starting location;*
- $T_m \in \mathbb{R}_0^+$ *is the limit of the total time interval* $[0, T_m]$ *during which observations were made;*
- $\mathcal{O} = \langle T_o, o \rangle$ *is the observation model such that* $t_2 \leq T_m$*, for every* $[t_1, t_2] \in T_o$*; and*
- *Prob is a (prior) probability distribution over* X_g*.*

The solution to a continuous GR problem $\mathcal{R}_c$ is a probability distribution across X_g and—given that Ramirez and Geffner's insight is as relevant in this setting as in the other—we achieve this probability distribution exactly as before, by evaluating and

[17] As standard, if there is no path (here, because there can be no connected path between the points), the minimum path cost is ∞.

comparing, for each goal, the cost difference between two optimal paths: one that embeds the observations and one that does not. The baseline cost difference formula (RG1), our simpler formula (3.1) and the single-observation formula (3.2) can be restated in a continuous setting as follows (here $x_g \in X_g$ is a possible goal, and $x_n \in X_{free}$):

$$costdif_{RG}(x_s, x_g, \mathcal{O}) = optc(x_s, \mathcal{O}, x_g) - optc^{\neg}(x_s, \mathcal{O}, x_g); \tag{3.3}$$

$$costdif_1(x_s, x_g, \mathcal{O}) = optc(x_s, \mathcal{O}, x_g) - optc(x_s, x_g); \tag{3.4}$$

$$costdif_2(x_s, x_g, x_n) = optc(x_n, x_g) - optc(x_s, x_g). \tag{3.5}$$

Finally, we can plug these cost differences into probability distribution (RG2) as before, without any loss of meaning.

Note. Reformulation of our path-planning framework for the continuous domain has necessitated significant changes, particularly in the notions of space, paths, cost model and observations. It turns out, however, that the results presented in sections 3.1.2 and 3.1.3 in relation to the discrete domain, also hold in the continuous setting.

The behaviour of our simpler formula (3.4) is unchanged, as follows.

Theorem 6. *Let $\mathcal{O}$ be an observation model in the scope of a continuous GR problem $\mathcal{R}_c$, then Theorems 1, 2, 3 and 4 hold.*

Proof. Referring to Theorems 1, 2, 3 and 4, we replace $n_s, n_g \in N$ with $x_s, x_g \in X_{free}$ and the observation sequence $\vec{o}$ with our observation model $\mathcal{O}$. Now $optc(n_s, \vec{o}, n_g)$ becomes $optc(x_s, \mathcal{O}, x_g)$, $optc^{\neg}(n_s, \vec{o}, n_g)$ becomes $optc^{\neg}(x_s, \mathcal{O}, x_g)$, $\Pi^*(n_s, n_g)$ becomes $\Pi^*(x_s, x_g)$ and $\Pi^*(n_s, \vec{o}, n_g)$ becomes $\Pi^*(x_s, \mathcal{O}, x_g)$. These substitutions made, the meaning of all four formulas is entirely preserved. □

In words, as in the discrete domain, $costdif_{RG}(x_s, x_g, \mathcal{O})$ yields a different result from $costdif_1(x_s, x_g, \mathcal{O})$ in the continuous setting *only* in the case of exclusive optimality (i.e., when the only way to achieve an optimal path is via the observations) and results in different rankings *only* if there is a second goal for which the observed path is also (exclusively or non-exclusively) optimal.

In addition, the analogue of Theorem 5 (which supports the single-observation formula) also holds in the continuous domain.

Theorem 7. *Let $\mathcal{O}$ be an observation model in the scope of a GR problem $\mathcal{R}_c$. Let x_n be the last observation in $\mathcal{O}$, that is, $x_n = \pi^*(1)$, where $\pi^* = \text{argmin}_{\pi \in \Pi(T_o)} cost(\pi)$. Then, for all $x_{g_1}, x_{g_2} \in \mathcal{G}, P_1(x_{g_1} \mid \mathcal{O}) > P_1(x_{g_2} \mid \mathcal{O})$ iff $P_2(x_{g_1} \mid x_n) > P_2(x_{g_2} \mid x_n)$.*

Proof. Observe that π^* is an *optimal* path that *embeds* the observations; it extends from the first observation to the last x_n and no further, that is, $\pi^*(0) = o(\vec{t_1})(0)$ and $\pi^*(1) = x_n = o(\vec{t_2})(1)$ where $\vec{t_1}, \vec{t_2}$ are the first and last observation, respectively, in T_o. As in the discrete case (Theorem 5), the optimal cost of a path that embeds observations can be considered as the sum of three parts, that is, $optc(x_s, \mathcal{O}, x_{g_i}) = optc(x_s, \pi^*(0)) + cost(\pi^*) + optc(\pi^*(1), x_{g_i})$. Similarly, therefore:

$$\begin{aligned}
&costdif_1(x_s, x_{g_1}, \mathcal{O}) - costdif_1(x_s, x_{g_2}, \mathcal{O}) \\
&\quad = [optc(x_s, \pi^*(0)) + cost(\pi^*) + optc(\pi^*(1), x_{g_1}) - optc(x_s, x_{g_1})] - \\
&\qquad [optc(x_s, \pi^*(0)) + cost(\pi^*) + optc(\pi^*(1), x_{g_2}) - optc(x_s, x_{g_2})] \\
&\quad = [optc(\pi^*(1), x_{g_1}) - optc(x_s, x_{g_1})] - [optc(\pi^*(1), x_{g_2}) - optc(x_s, x_{g_2})] \\
&\quad = costdif_2(x_s, x_{g_1}, x_n) - costdif_2(x_s, x_{g_2}, x_n).
\end{aligned}$$

Since $costdif_1(x_s, x_{g_1}, \mathcal{O}) > costdif_1(x_s, x_{g_2}, \mathcal{O})$ iff $costdif_2(x_s, x_{g_1}, x_n) > costdif_2(x_s, x_{g_2}, x_n)$, we get $P_1(x_{g_1} \mid \mathcal{O}) > P_1(x_{g_2} \mid \mathcal{O})$ iff $P_2(x_{g_1} \mid x_n) > P_2(x_{g_2} \mid x_n)$. □

This concludes reformulation of our GR account from the discrete to the continuous setting. Recall that our motivation is not merely to demonstrate equivalence of results, but to set up the ground for calculation of the RMP, which follows next.

3.2.2 Calculation of the Radius of Maximum Probability

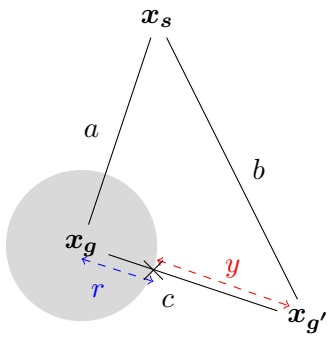
Calculation of a distance from goal, within which that goal is guaranteed to be the most probable, depends on the relationship—implicit in the continuous single-observation formula (3.5)—between a goal's probability and an agent's precise location. As we have seen, a major implication of this relationship is that—provided we use a probability distribution formula which satisfies the property that the lower the cost difference, the higher the probability, such as the R&G formula (RG2)—goal rankings (with respect to their probabilities) at every point in the domain remain constant, regardless of the path taken to get there.

One way of exploiting this finding, as already discussed in terms of the discrete domain, is to pre-calculate probabilities node by node to create a probabilistic heatmap (as at Figure 3.9, p.58). This is an effective way of revealing the complete perimeter within which a goal becomes most probable. It is, however, computationally demanding, even in a discrete domain; while in a continuous domain, it would clearly be impossible to pre-calculate probabilities for infinitely many points!

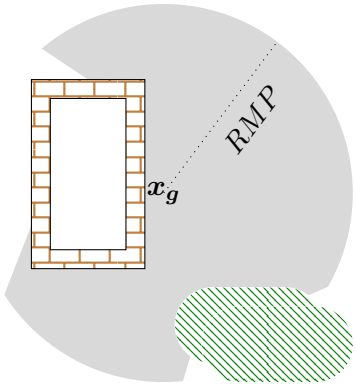
Building on the single-observation formula (3.5), however—and provided that we know the relative locations of the starting point and candidate goals (information typically available in a known domain)—an alternative approach is available, as we now show.

We first precisely define the cost-distance[18] that we propose to measure.

[18] Recall that we have made the simplifying assumption that the cost of a path is equal to its length.



(a) Labels a, b, and c stand for optimal cost-distances. We show only two goals; if we find tipping points (marked here by a cross) between x_g and *multiple* goals, then RMP_g is the *minimum* r.



(b) Only the grey area is *within* RMP_g. (Shading is indicative only; dimensions are not precise.)

Figure 3.10: The Radius of Maximum Probability.

Definition 7. *Given a probabilistic GR problem for continuous path-planning $\mathcal{R}_c$, the* ***radius of maximum probability (RMP)*** *for a possible goal $x_g \in X_g$, denoted* RMP_g*, is a distance $r \in \mathbb{R}$ such that:*

1. *for all $x \in X_{free}$ such that $optc(x, x_g) < r$, it is the case that $P(x_g \mid x) > P(x'_g \mid x)$, for all $x'_g \in X_g \setminus \{x_g\}$; and*
2. *there exists a point $x' \in X_{free}$ such that $optc(x', x_g) = r$ and $P(x_g \mid x') = P(x'_g \mid x')$, for some $x'_g \in X_g \setminus \{x_g\}$.*[19]

Intuitively, RMP_g signals a ***tipping point***: a distance at which the probability of x_g becomes equal to the probability of some other goal, that is, a distance from x_g where probabilities 'flip' from favouring an alternative goal to the goal in question. At all points *within* this distance, the probability of x_g dominates; at *some* point just beyond this distance, some other goal $x_{g'}$ becomes more likely.

Since probabilities are based on cost difference, a point $x \in X_{free}$ at the tipping point has the property (by Observation 1) that $costdif_2(x_s, x_g, x) = costdif_2(x_s, x'_g, x)$. It is this property that enables us to develop a closed formula for the RMP, as follows.

Consider the simple two-goal domain depicted in Figure 3.10a. We wish to construct a formula for the distance denoted r (i.e., RMP_g). In what follows, take $a = optc(x_s, x_g)$, $b = optc(x_s, x_{g'})$, $c = optc(x_g, x_{g'})$, and $y = c - r$. The tipping point (t) must occur at some (unknown) point on an optimal path from x_g to $x_{g'}$, where $optc(t, x_g) = r$ and $optc(t, x_{g'}) = y$. Using the continuous single-observation cost difference formula (3.5) and

[19] If point 2 in this definition were omitted, every cost-distance greater than r would qualify as an RMP.

retaining the abbreviated terms for legibility:

$$\begin{aligned} \mathit{costdif}_2(x_s, x_g, t) &= \mathit{costdif}_2(x_s, x_{g'}, t) \\ r - a &= y - b \\ y &= r + b - a \\ c &= r + (r + b - a) && \text{from } y = c - r \\ &= 2r + b - a \\ r &= \frac{c + a - b}{2}. \end{aligned}$$

Thus, taking cost-distances between the starting point, the goal of interest and one other goal, we obtain r, the optimal cost from goal location x_g to the point at which the tipping point t must occur.

Of course, there may be many potential goals. Therefore, to construct a general formula for RMP_g, we take the minimum (recall x_s is the starting point and X_g the set of all candidate goals):

$$\text{RMP}_g(x_s, X_g) = \min_{x_{g'} \in X_g \setminus \{x_g\}} \frac{\mathit{optc}(x_g, x_{g'}) + \mathit{optc}(x_s, x_g) - \mathit{optc}(x_s, x_{g'})}{2}. \tag{3.6}$$

> The RMP formula identifies the radius within which a goal is guaranteed most probable, without having to calculate any probability distributions.

With respect to complexity, calculation of a single RMP (e.g., RMP_g as above) requires $2|X_g| - 1$ calls to a path-planner. If we wanted to calculate RMPs for every goal in the domain, it would take $\frac{|X_g|^2+|X_g|}{2}$ calls. So, for example, in a domain with five candidate goals, calculation of one RMP requires nine calls to the planner whereas calculation of RMPs for all five goals would require 15. Compare this with the computational cost of calculating a probability distribution at a single point. This would require $|X_g|$ calls using $P_2(\cdot)$ or $2|X_g|$ calls using the baseline R&G formula $P_{RG}(\cdot)$: one more call than is necessary to calculate the RMP for one goal, yet the area that an RMP describes may contain many points of interest. To determine the most probable goal at all those points *without* using the RMP, probability distributions would have to be calculated individually at every one.

The significance of formula (3.6) rests on three factors: it is relatively inexpensive computationally (as just discussed); it depends on distances between locations that, in a static domain, are typically known and available; and it frees us from the computational cost of calculating any probability distributions whatsoever. Moreover, to restate the formula's impact, once the calculation has been performed (and it need only be performed once per goal), we are able to identify a clear boundary—a target area—within which each goal is guaranteed to be the most probable.

Before proceeding, we clarify two important points:

1. The measurement represented by the RMP is a *cost*-distance, not distance per se. Consider, for example, the continuous terrain depicted in Figure 3.10b. The green area represents an obstacle (e.g., impenetrable forest): it is not part of X_{free} and, therefore, cannot be occupied by an agent. The cost of reaching x_g from a point inside an obstacle is infinite (i.e., impossible); thus, even though based on distance the forest's boundary appears to be within RMP_g, based on *cost* it is not.

   Note that Figure 3.10b also depicts a bricked compound. The white area inside the compound *is* in X_{free} (and could, theoretically, be occupied by an agent deposited there somehow). The region is inaccessible however and so, despite being in X_{free}, again its *cost*-distance (from *any* goal) is infinite. In short, whatever the value of RMP_g, it is always exceeded by the (infinite) cost-distance to x_g from any inaccessible point. (See too that the impossibility of traversing those inaccessible areas means that other points (unshaded) become more distant from x_g and are forced outside RMP_g.)

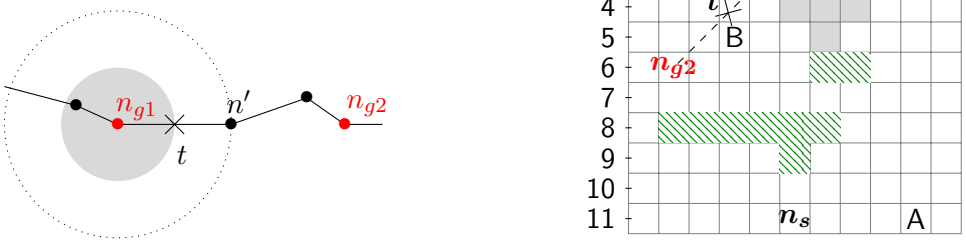
2. The fact that a point x lies beyond the cost-distance represented by RMP_g, does not imply that the probability of x_g at x is less than the probability of some other goal. Recall that the guarantee concerns only those points within RMP_g; it says *nothing* about points outside that radius. Indeed, there may be many points beyond RMP_g where x_g is still the most probable goal; but probabilities at those points cannot be guaranteed: if we need to know them, we must calculate them individually, using some other method (such as single-observation recognition).

   Although it depicts a discrete domain, Figure 3.11b conveniently illustrates the case. Let us say that straight moves cost 1, diagonals $\sqrt{2}$. Now, if we calculate probabilities for an agent at cell (9,11) marked "A", her most probable goal is n_{g1}, approximately 10 units away. An agent at (4,5) marked "B", on the other hand—less than 4 units from n_{g1}—is most probably heading for goal n_{g2} (not n_{g1}). The RMP tells us the *minimum* distance within which we can guarantee the probability of one goal is greater than the probability of any other: in this case, $\text{RMP}_{g1} < 4$ even though n_{g1} is the most probable goal at other cells 8, 9 and 10 units away.

Although our theorems hold in both path-planning domains, the RMP is a real number calculated over continuous distance, which presents difficulties when we try to apply it in a discrete domain, as we consider next.

3.2.3 RMP in the Discrete Domain

The calculation (and definition) of an RMP in the discrete domain is complicated by the possibility that there may be no node at the precise cost-distance from a goal where the tipping point between goals ought to occur.



(a) In a graph, the tipping point t (marked here by a cross) may fall between nodes: there is no node where $P(n_{g1}) = P(n_{g2})$.

(b) In a grid, the tipping point t is always *within* some cell but probabilities *at* that cell cannot be guaranteed.

Figure 3.11: The RMP in discrete domains. The grey shaded areas are *within* the RMP.

Graph-based representation. Referring to Figure 3.11a, we see that when a discrete domain is represented as a graph (e.g., a train network or the locations of mobile towers along a route), there may literally be no node (i.e., no station or no mobile tower) at the precise tipping point between two candidate goals, even in the presence of no obstacles (i.e., $X_{obst} = \emptyset$). Thus, in the discrete domain, although the RMP can be calculated in exactly the same way as before, here, it represents a theoretical minimum distance; in practice, a no less useful lower bound.

Referring again to Figure 3.11, we see that, if there were a node at t we could say that $P(n_{g1} \mid t) = P(n_{g2} \mid t)$ and that at all nodes closer to goal than t (within the shaded area), the probability of n_{g1} dominates. In this example, however, the first existing node where $P(n_{g1}) \leq P(n_{g2})$ is the node marked n', so the shaded area (within which no goal is more probable than n_{g1}) could be extended up to the radius marked by the dotted line. Observe that the cost-distance $optc(n', n_{g1})$ is in fact *greater* than the value returned by Equation (3.6). Thus, in a domain discretised into a graph, the RMP provides a lower bound for the distance of interest.

Grid representation. When a path-planning space is discretised into a grid, the situation is slightly different. In a gridworld environment, nodes are represented as cells and every cell is immediately adjacent to some other cell. Therefore, it can never happen that the tipping point t occurs at a point where there is no cell at all. Furthermore, since t is identified as a special point on the optimal path from one goal to another—the path within which t is located is known to be traversable—there must be a cell 'containing' t that can be reached.

Knowing that, theoretically, some cell n_t would contain point t, however, does not guarantee that the probability of two goals would be equal when measured at that cell: a cell n_t might theoretically *contain* a tipping point without *being* a tipping point. Recall

that, in a gridworld environment, distance (and cost) are only calculable in discrete chunks. Referring to Figure 3.11b, where cost is 1 for straight moves and $\sqrt{2}$ for diagonal moves, Equation (3.6) returns a value of 3.53 for RMP_{g1}, but the cost of cell traversal on an optimal path from n_{g1} to n_{g2} jumps from 3.41 at cell $(4, 4)$ to 4.82 at $(3, 5)$ or from 2.4 at $(5, 4)$ to 3.8 at $(4, 5)$. So, we cannot know whether probabilities at n_t favour n_{g1} or n_{g2} or are equal. What we do know is that for all cells n' such that $optc(n', n_{g1}) < optc(n_t, n_{g1})$, it is the case that $P(n_{g1} \mid n') > P(n_{g2} \mid n')$, for all $n_{g2} \in G \setminus \{n_{g1}\}$.

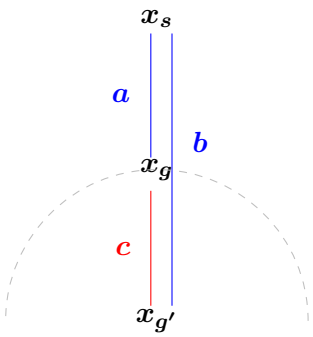
Heatmap example. Before looking in detail at the properties of the RMP, we remind the reader of the heatmap example discussed in Subsection 3.1.3 (p.55). We explained there that one could use Equation (3.2) to pre-calculate a complete heatmap of probabilities (Figure 3.9) from which to identify the perimeter within which goal n_{g1} or n_{g2} was the most probable. Using Equation (3.6), we now have the option of instead instantly calculating the minimum distance of that perimeter from either goal without having to pre-calculate any of those probabilities:

$$\text{RMP}_{g2} = \frac{9 + 8 - 11}{2} = \frac{6}{2} = 3$$
$$\text{RMP}_{g1} = \frac{9 + 11 - 8}{2} = \frac{12}{2} = 6$$

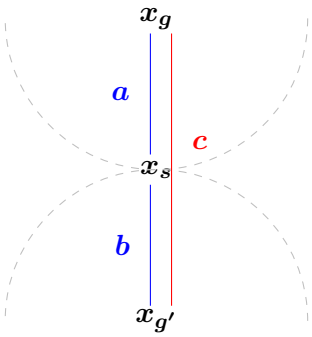
Using this calculation—and nothing else—we can identify a clear target area within which to focus our attention. If we have a particular interest in monitoring or protecting goal n_{g1} or n_{g2}, for example, we can deploy our surveillance effort into access points located 3 (or 6) units of cost-distance away, knowing that any agent who breaches that radius is most likely heading for our goal of interest.

Note. To clarify: the *heatmap* is an artefact—a potential application of the single-observation cost difference formula (3.2)—that can be obtained by repeated use of that formula for every node in the domain (or every location of interest) at a computational cost of $|G| \cdot k$ calls to a path-planner, where k is the total number of locations for which probabilities are required. It can serve as a brute-force method for identifying the perimeter around a 'most probable goal' and, once calculated in an offline process, can also be used to access a complete probability distribution for every point that has been evaluated: a resource that can be accessed in constant time to support, for example, online 'spot-checks' of an agent acting suspiciously.

The *RMP* is a number: a distance from a given goal, which need be calculated once only at a computational cost comparable to calculation of a single probability distribution. Similar to the heatmap, it can be a useful tool on the ground (e.g., for use by a surveillance operative who only needs to know whether an agent is inside or outside the target zone). It might also be used as a complementary tool in a goal recognition design process (Keren et al., 2014), where it has similarities to 'worst case



(a) The optimal path to $x_{g'}$ goes through x_g, so $\text{RMP}_g = 0$. Dotted grey lines show RMPs for x_g and $x_{g'}$.



(b) The optimal path from x_g to $x_{g'}$ goes through x_s. Unequal priors may put x_s *inside* RMP_g or $\text{RMP}_{g'}$.

Figure 3.12: Properties of RMPs.

distinctiveness' (see Chapter 2, p.23), or in place of landmarks for GR or goal pruning (Vered et al., 2018). Furthermore, being quickly calculable (unlike a heatmap), the RMP can be used in online scenarios, for example where the terrain changes dynamically (Raffe et al., 2012), if candidate goals only become known in real-time or are agent-specific; that is, where pre-calculation would be impossible.

3.2.4 Properties of the RMP

The RMP identifies a radius around a goal within which that goal is guaranteed to be the most probable. It is important to realise however that, depending on the particular terrain and location of goals, there may be *no* RMP for a particular goal, or rather, the radius within which its probability dominates that of all other goals may be precisely zero.

Consider the two goals at Figure 3.12a. Here, the goals and starting point are in a direct line, that is, the optimal path to $x_{g'}$ goes directly through x_g. Calculating the RMPs for each goal, we get:

$$\text{RMP}_g = \frac{c + a - b}{2} = 0$$
$$\text{RMP}_{g'} = \frac{c + b - a}{2} = \frac{2c}{2} = c$$

This coincides with the intuition that, if an agent is approaching x_g from x_s (and is therefore on the optimal path to *both* goals) there may *never* be a point where we can discount the possibility that $x_{g'}$ is the agent's goal; therefore $\text{RMP}_g = 0$. On the other hand, the moment the agent proceeds *beyond* x_g, it is the case that $x_{g'}$ becomes most probable; and appropriately $\text{RMP}_{g'} = c$.

Consider now the two goals at Figure 3.12b. Again, the goals and starting point are in a direct line but this time the relationship is different in that the optimal path from x_g to $x_{g'}$ goes directly through the starting point x_s. Calculating the RMPs for each goal,

we obtain:

$$\text{RMP}_g = \frac{c+a-b}{2} = \frac{a+b+a-b}{2} = a$$
$$\text{RMP}_{g'} = \frac{c+b-a}{2} = \frac{a+b+b-a}{2} = b$$

This again coincides with our intuition: if two goals lie in opposite directions from one another relative to the starting point, such that the optimal path to one discounts the optimal path to the other, probabilities 'flip' to favour one or the other as soon as the agent commits to a direction.

Finally, note that with the introduction of priors, the size of an RMP increases (or decreases) relative to the prior probability of the goal in question. We leave investigation of this interesting topic for future work except to note that this could have the effect of placing the starting point x_s *inside* a goal's RMP: that is, one goal may be 'most probable' from the start.

The RMP is almost paradoxical in itself: it reasons about probabilities without calculating probabilities. It represents a novel approach to GR which is fast, entirely domain-based and has many potential applications. The RMP formula (3.6) is calculable as a direct consequence of the relationship between location and probability revealed by our investigation of single-observation recognition. Before moving on to discussion, we present our evaluation of the cost difference formulas on which it depends.

3.3 Experimental Evaluation

We have seen that, in the context of path-planning, the R&G model of GR can be reformulated to arrive at single-observation recognition and that the single-observation cost difference formula at the heart of our single-observation account can then be used to calculate the RMP around a goal in discrete or continuous domains.

In this section, we report on the results of our experimentation. We tested the performance of GR in path-planning using R&G's complex cost difference (RG1), the simpler version (3.1) (that does not reason negatively about observations), and the single-observation formula (3.2), which we have proposed for online and offline recognition (to generate a probabilistic heatmap) and as the basis of the RMP formula (3.6). Tests were conducted using problems adapted from the well-known Moving-AI[20] path-planning benchmarks (Sturtevant, 2012), which discretise the underlying maps and groundplans to a 512×512 grid. We experimented in a discrete environment and have relied on

[20] http://movingai.com/

our theoretical conclusions to support the applicability of our formulas in the continuous domain.[21]

Our aim was to develop an experimental framework for GR in path-planning to empirically confirm: (i) that the case of exclusive optimality (as in Theorem 3) is rare and, otherwise, the simpler formula (3.1) yields identical posterior probability distributions to (RG1); (ii) that all three accounts return posterior probability distributions that rank goals the same; and (iii) that use of either formula (3.1) or (3.2) cuts processing time by more than half.

3.3.1 Experimental setup

We generated over 2500 individual probability distributions from a problem set of 774, built on 43 scenarios selected at random from two sets of Moving-AI benchmarks (Sturtevant, 2012):[22] game landscapes from StarCraft; and connected room layouts (chosen for their similarity to internal locations, such as airport terminals or shopping centres). Since the scenarios are intended for path-planning, we adapted them for GR as follows. First, we added two to five additional (reachable) candidate goals at random locations. Second, to generate the observations, we used Weighted-A* (Pohl, 1970) to build three full continuous paths from the start location to the real goal, differing in quality: one optimal, one suboptimal, one greedy. We then extracted observation sequences varying two further dimensions: 'observation density', that is, the proportion of the continuous path used in the extracted observation sequence (sparse 20%, medium 50%, dense 80%) and 'observation strategy', that is, the method of extracting the observations, which was either random (taking the required density of observations from random locations anywhere along the path) or prefix (taking the required density as a consecutive sequence of nodes from the start location on).

In preliminary tests, we had found that the probability distribution formula (RG2) was particularly sensitive to any small variation in cost difference. This was especially noticeable with the large negative values often returned by our single-observation formula (3.2) for the most probable goal in a distribution (where the optimal cost of a complete path is subtracted from the much smaller cost of reaching that goal from the more recent observation).

Note. We initially attributed the sensitivity of R&G's probability distribution formula (RG2) to the use of exponential values in generating posterior probabilities. Sub-

[21] Arguably, a grid can be regarded as a middle way between continuous and discrete domains in that while, algorithmically, it behaves like a graph (i.e., we can use variations of Dijkstra's algorithm or A* to find shortest paths), it can equally be used to discretise almost any continuous space, adjusting the size of the grid to whatever granularity of solution is required.

[22] Experiments were conducted on an i7 3.4GHz dual core with 10GB RAM in a virtual Linux environment; preliminary and manual tests were conducted on a similar 1.8GHz machine.

sequent examination, however—conducted while investigating its performance when dealing with excessively suboptimal paths, which we report on in Part II—reveals a 'quirk' in the formula so that, although it reliably enforces the intuition that lower cost difference results in higher probability *across* goals, it does not enforce that intuition with respect to the probability values for any *one* goal.

In fact, as we will show (p.105), the higher the cost difference for any one goal, the higher its individual probability value. Conversely, with our unusually low (negative) cost differences using the single-observation formula, low cost difference resulted in low probability values, which exagerated the delta we recorded between the value returned by single-observation and the value returned using complex cost difference.

To compensate for the unexpected behaviour noted above, in addition to the three probability distributions derived using formula (RG2) with the original (baseline) formula (P_{RG}), the simpler formula (P_1) and the single-observation formula (P_2), we also included a variation on (P_2) obtained by adding a large constant value (which we set at 800) to the cost difference returned by the single-observation formula ($P_2{}^*$). Recall that the β constant is a rate parameter, which modulates the shape of the distribution (as further discussed in Part 2, p.111). For the automated tests, we adopted a β value of 0.1 throughout. (With $\beta = 1$, P_{RG} and P_1 tended to return 1 for the most probable goal and otherwise 0, while with $\beta = 0.01$, the distribution tended to even out and, with loss of precision on test equipment, returned for example, 0.33 for each of three goals, 0.25 for each of four, and so on). We used the usual uniform-cost approach for grids, with horizontal and vertical moves costed at 1, and diagonal moves at $\sqrt{2}$; and we made the simplifying assumption that priors were equal.

In addition to the auto-generated problem set, we manually set up individual experiments to trial the various cost difference formulas against completely open landscapes and 'single-pixel' mazes (through which there is typically only one path from any given starting point to goal). For simplicity and, given that planners are meant to be used off-the-shelf, optimal costs for paths with, and without, waypoints were calculated using a standard A* algorithm (Hart et al., 1968)[23]. To obtain the cost of an optimal path that did *not* embed the observations, inspired by the technique used for R&G, we modified A* so that each search node, in addition to a location indicator, also included an observation counter. When the counter reached the total number of observations (meaning all observations had been encountered) the search node representing that last observation—and so the associated path that embedded all the observations—was pruned.

[23]We used our own Python-based infrastructure, originally designed as a simulator and testbed for path-planning algorithms (`https://tinyurl.com/p4sim`).

3.3.2 Results

Our theoretical results were confirmed. In hand-crafted problems using maps with open landscapes, the formulas performed exactly as predicted: formulas (RG1) and (3.1) returned identical results, all four formulas ranked goals identically by probability and (after the first iteration in a domain with the same start location and goals) formulas (3.1) and (3.2) returned in half the time of formula (RG1).

In the single-pixel maze, again as predicted, the implementation based on formula (RG1) was unable to return a probability distribution (because the most probable goal gave a cost difference of $-\infty$), whereas formulas (3.1) and (3.2) returned in 0.005 and 0.002 seconds, respectively, and successfully identified the real goal.

The corner case of exclusive optimality, in which observations conform to the only optimal path to goal, did not arise in any of the randomly generated scenarios. This is perhaps unsurprising, given the symmetries found in a two-dimensional grid. Nevertheless, we were able to reproduce the condition by exactly replicating the example scenario. That is, we set up an environment in which diagonal moves were prohibited, there were three goals, observations were on the optimal path to all of them but one goal lay in a straight line from the start location. In the resulting probability distributions, formula (RG1) returned 0.329, 0.342, 0.329 (for L, M and N in Figure 3.13, respectively), whereas formulas (3.1) and (3.2) returned 0.333 for all three goals.

Tables 3.3 and 3.4 summarise the results of our automated tests. Column Obs displays the percentage of nodes from the full path that were included in the observation sequence. P indicates that the observations were extracted using the continuous path prefix strategy, and R that they were extracted using the random strategy, that is, randomly drawn from the length of the path. Column Time displays the average time-taken per GR problem in seconds. Column Match shows the percentage of probability distributions where the probability value for the target goal exactly matched that generated using cost difference formula (RG1). Where a difference was recorded (only in the cases of P_2 and $P_2{}^*$), column Δ displays the average difference. Our main findings were as follows.

- The implementation using cost-difference (RG1) performed even more slowly than we expected.
  - In the room layouts (Table 3.3), it frequently exceeded our three minute time-out; the longer the paths (i.e., the larger the set of observations) and the more optimal the path they were extracted from, the longer the algorithm took. This is explained by the difficulty of identifying an alternative optimal path (i.e., when we randomly selected problems that timed out and let them run to completion, cost difference for the target goal ultimately returned zero). The problem was exacerbated in room layouts, which are significantly more restrictive than landscapes (doorways are only one-pixel wide).

Table 3.3: Rooms.

		P_{RG}	P_1	P_2, $P_2{}^*$	P_2		$P_2{}^*$	
	Obs	Time	Time	Time	Match	Δ	Match	Δ
Optimal	20%P	94.538	7.223	**3.400**	6.7%	0.202	40.0%	**0.031**
	20%R	68.086	3.316	**2.918**	10.0%	0.340	50.0%	**0.041**
	50%P	180+	3.075	**2.723**	0%	0.487	36.7%	**0.030**
	50%R	83.381	3.473	**3.068**	16.7%	0.313	50.0%	**0.040**
	80%P	180+	3.360	**2.967**	3.3%	0.475	50.0%	**0.052**
	80%R	180+	3.716	**2.991**	16.7%	0.332	50.0%	**0.037**
Suboptimal	20%P	94.609	7.210	**3.457**	10.0%	0.190	36.7%	**0.023**
	20%R	61.842	3.456	**2.993**	13.3%	0.344	56.7%	**0.025**
	50%P	180+	3.319	**2.782**	0%	0.417	40.0%	**0.021**
	50%R	74.184	3.593	**3.073**	16.7%	0.290	60.0%	**0.026**
	80%P	180+	3.435	**2.993**	3.3%	0.415	50.0%	**0.030**
	80%R	88.831	3.729	**3.100**	13.3%	0.332	60.0%	**0.022**
Greedy	20%P	92.260	7.193	**3.287**	10.0%	0.202	56.7%	**0.014**
	20%R	58.117	3.346	**2.919**	13.3%	0.382	80.0%	**0.032**
	50%P	58.667	3.231	**2.634**	0%	0.410	66.7%	**0.014**
	50%R	70.057	3.548	**2.996**	13.3%	0.367	80.0%	**0.031**
	80%P	61.732	3.278	**2.655**	3.3%	0.448	70.0%	**0.024**
	80%R	91.231	3.675	**2.983**	10.0%	0.399	83.3%	**0.029**

540 problems. Average goals: 4.9. Average optimal path cost: 372. Probabilities calculated using formula (RG2) with β value of 0.1. We obtained $P2^*$ as P_2 but adding a constant (800) to the corresponding cost difference (see discussion inline). The Match column indicates the percentage of cases where probability values matched exactly. The Δ column indicates average difference in non-matching values.

– Observations presented as a path prefix took, in some cases, twice as long to solve as those presented randomly. This seems to be because the pruning algorithm backtracks so, if observations are consecutive, it repeatedly reaches the final observation via multiple different routes.

– Ultimately, the relative slowness may be a symptom of the calculation's inherent complexity. We note that the Easy IPC Grid experiments reported by Ramirez and Geffner (2010) (which include a significant navigational element, though in the more demanding context of general task-planning)[24] also took, on average, over three minutes to complete problems with observation densities of 50% using an optimal planner comparable to A*, on problems with average optimal path lengths of just 17 steps. Ramirez and Geffner improved performance by using a suboptimal planner. We did try a suboptimal—much faster—algorithm but, although it returned approximately equivalent probability distributions, it failed to preserve the corner cases, which were of interest to us.

[24]Easy IPC Grids have far fewer cells than Moving-AI maps so they are easier to navigate and optimal paths are shorter. Problems are more complex, however, as they include task-planning elements, e.g., that keys may be required to access particular cells.

Table 3.4: Landscapes.

	Obs	P_{RG} Time	P_1 Time	P_2, $P_2{}^*$ Time	P_2 Match	P_2 Δ	$P_2{}^*$ Match	$P_2{}^*$ Δ
Optimal	20%P	34.385	3.444	**1.344**	0%	0.140	7.7%	**0.043**
	20%R	19.135	1.749	**1.646**	7.7%	0.317	69.2%	**0.009**
	50%P	51.433	1.541	**1.379**	0%	0.247	30.8%	**0.034**
	50%R	37.100	1.907	**1.672**	15.4%	0.299	69.2%	**0.009**
	80%P	56.109	1.917	**1.515**	7.7%	0.284	46.2%	**0.027**
	80%R	49.645	2.015	**1.687**	15.4%	0.344	69.2%	**0.009**
Suboptimal	20%P	35.183	3.300	**1.446**	15.4%	0.143	38.5%	**0.041**
	20%R	18.939	1.797	**1.690**	15.4%	0.347	69.2%	**0.010**
	50%P	51.395	1.625	**1.450**	15.4%	0.227	46.2%	**0.028**
	50%R	35.180	1.898	**1.669**	15.4%	0.324	69.2%	**0.011**
	80%P	55.780	1.912	**1.564**	7.7%	0.247	46.2%	**0.017**
	80%R	48.455	1.922	**1.731**	15.4%	0.335	69.2%	**0.011**
Greedy	20%P	35.400	3.342	**1.451**	15.4%	0.146	38.5%	**0.038**
	20%R	16.678	1.781	**1.679**	15.4%	0.351	69.2%	**0.011**
	50%P	50.662	1.725	**1.421**	15.4%	0.250	46.2%	**0.013**
	50%R	33.433	1.827	**1.706**	15.4%	0.337	69.2%	**0.011**
	80%P	54.790	1.952	**1.597**	7.7%	0.268	46.2%	**0.011**
	80%R	48.024	2.020	**1.729**	15.4%	0.345	69.2%	**0.011**

234 problems. Average goals: 4. Average optimal path cost: 233.04. Probabilities calculated with β of 0.1 and P_2^* constant of 800, as at Table 3.3. We note that in room layouts, all traversable locations are accessible from one another whereas in landscapes, automatically generated goal locations were frequently inaccessible from the start location resulting in fewer usable scenarios.

- Use of formula (3.1) cut processing time even from landscapes (Table 3.4) by more than an order of magnitude. We should note that, in our experiments, the 20% density, prefix observations were always the first to be tested in each new problem set. This meant that it was always when running the $20P$ test that optimal costs to each goal were calculated (and stored for future use). This is reflected in the results, which clearly show the simple formula taking approximately twice the time for that problem as subsequent problems.

- Although time-savings were on nothing like the same scale, we note that average timings for the single-observation formula (3.2) were consistently lower than those for formula (3.1).

- Whereas the probabilities based on cost difference formula (3.1) always exactly matched those based on cost difference formula (RG1), probabilities generated using formula (3.2) were usually different.

  – This is because the *actual* values returned by that formula are different; it is the *relative* cost differences that are maintained. This is the anomalous effect

discussed above (see p.75) and in our concluding note below.

- As can be seen, delta values for P_2 were sometimes quite high thanks to the typically large negative cost difference for the most probable goal. For $P_2{}^*$, as discussed, we compensated for this effect by adding a large constant to the function's output, which raised it always above zero. This significantly reduced the delta.
- In any event, observe that, whatever the delta, relative rank is always preserved. In particular, whether or not the constant is added, in all cases, use of the single-observation formula successfully identified the same goal as having the highest, or equal highest, posterior probability as either of the other formulas.

Note. We conclude by noting that the anomaly discussed here with respect to probability distribution formula (RG2) is resolved by our reformulation of the formula in Part II (see Equation (4.5), p.110). We predict that use of our revised formula will return identical probability distributions for each of the three cost differences examined above without requiring any additional manipulation (i.e., there will be no need for the addition of an extra constant to bring P_2 more or less in line with P_1). We leave experimental confirmation of this prediction for future work.

3.4 Discussion

In the previous three sections of this chapter, we have transposed the R&G model of GR from task-planning to path-planning and have demonstrated the considerable efficiencies that can be achieved in this context. This section discusses some of the broader issues that arise in relation to our work: first, the special case where the single-observation formula ranks goals differently from Ramirez and Geffner's original cost difference formula (which did not arise in testing), then the extent to which our results apply in a general task-planning domain and, finally, its relationship with plan (as opposed to goal) recognition.

3.4.1 Corner Case: Exclusive Optimality and Negative Reasoning

There is only one corner case where our simpler and single-observation formulas (3.1) and (3.2) rank goals differently from R&G's original complex cost difference formula (RG1). This case arises only when observations conform to the optimal path for multiple goals and are *exclusively* optimal for at least one of them. Using our formulas (3.1) or (3.2), all such goals are ranked equally; using formula (RG1), which depends on negative reasoning, rankings may differ depending on the length of alternative (non-optimal) paths to the various goals.

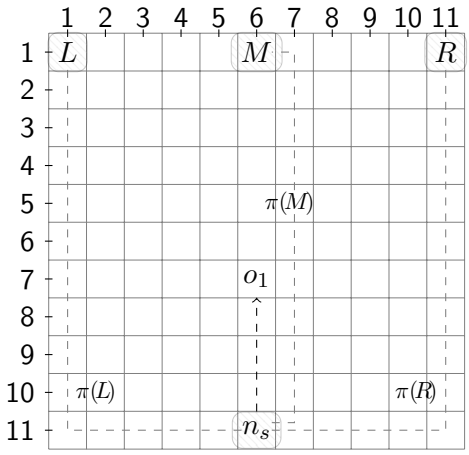


Figure 3.13: Example 1: exclusive optimality.
Observation o_1 is on an optimal path to all three goals but, by formula (RG1), M is the most probable.

Ramirez and Geffner (2010) support the use of negative reasoning by reference to the following example, where observations are optimal for all three goals but exclusively optimal for only one.

Example 1. *Consider the situation depicted in Figure 3.13. An agent operates in a discrete gridworld environment where the only legal moves are horizontal or vertical and all steps cost* 1*. There are three possible goals,* $g_l, g_m, g_r \in G$*, labelled* L*,* M *and* R *respectively. All goals are north of the start location,* n_s*. Observations* $\vec{o}$ *track directly north through the marked observation* o_1 *and satisfy an optimal path to all three goals. In the case of* L *and* R*, there are multiple optimal paths to goal so the optimal path that embeds the observations has the same cost (*15*) as one that does not: there is no cost difference; therefore,* $costdif(n_s, g_l, \vec{o}) = 0$ *and* $costdif(n_s, g_r, \vec{o}) = 0$ *(see Equation RG1). In the case of* M*, however, which lies directly north of* n_s *and* o_1*, there is only one optimal path to goal: the one that embeds the observations. In order to take a path that does not embed them, it is necessary to take a longer route. In the example,* $optc(n_s, \vec{o}, g_m) = 10$*, whereas* $optc^{\neg}(n_s, \vec{o}, g_m) = 12$*. Thus,* $costdif(n_s, g_m, \vec{o}) = -2$*. The lower the cost difference, the higher the probability, making* g_m *(*M*) the most probable goal.*

Although cited as an "illustration" of the distinction between cost difference formulas (RG1) and (3.1) (Ramirez & Geffner, 2010, p.1123), this scenario, in fact, represents the *only* distinction—a case of exclusive optimality—as proved in Theorem 3. Given the considerable additional computational work required to achieve formula (RG1), it is worth noting that this special case is concerned only with that set of goals in which the probabilistic account is least interested, namely goals for which observations are on the *optimal* path; that is, the case already handled in the non-probabilistic account (Ramirez & Geffner, 2009). Nevertheless, let us consider what is lost (and gained) by substituting either the simpler or single-observation formula for (RG1).

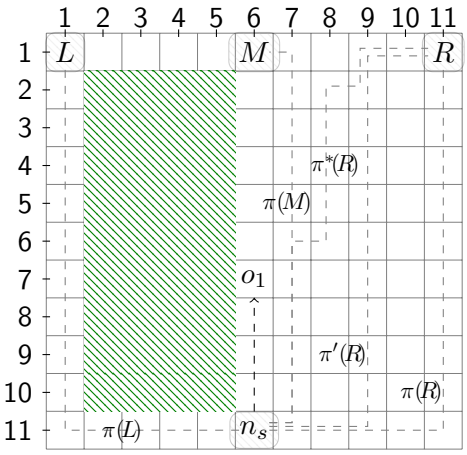


Figure 3.14: Example 2: an anomaly.

To arrive at a probability distribution, R&G appeals to Bayes' Rule, which, using our notation, can be given as $P(G \mid \vec{o}) = \alpha P(\vec{o} \mid G) \cdot Prob$, where α is a normalising constant. Assuming that prior probabilities in $Prob$ are given, the challenge is to account for $P(\vec{o} \mid G)$. The authors assert that it is correct for $P(\vec{o} \mid g_m)$ to exceed the probabilities of either of the other goals because "goal M predicts the observations better than either L or R" (p.1123). The intuition is that, in order to reach g_m optimally, an agent from n_s *must* pass through the observation o_1; whereas, to reach g_l or g_r optimally, the agent *might* (or might not) pass through o_1.

This reasoning seems to link probabilities to the number of available paths to goal. Indeed, Ramirez and Geffner (2010) acknowledge that there are situations where it would be preferable to count the number of paths but their framework does not support it. Thus, one might think that, if there had been four optimal paths and the agent had been seen on one of them, the probability of the goal would be correspondingly lower (because the goal predicts the observations less well than if there had been only one optimal path); and that, if there had been 100 optimal paths, it would be lower still. This is not the case, however. In fact, as soon as there is a second optimal path to goal, the account fails to follow the intuition, as in the following counter-example.

Example 2. *Consider the domain depicted in Figure 3.14. Here we have added a rectangular block—the patterned green cells, (2,2) to (5,10)—which is not traversable. Again there are three goals, labelled L, M and R but the change has now made g_l to be very like g_m and very different from g_r. There are just two optimal paths to g_l, only one more than to g_m. Meanwhile (owing to the notorious symmetry of gridworlds), there are 3003 optimal paths to g_r, as before, and yet Equation (RG1) can no more distinguish between g_l and g_r (which are* non-*exclusively optimal) than can the simpler formula (3.1). In this scenario, the probability of g_l should—based on how well the goal predicts the observations—be very much greater than g_r (only a little less likely than g_m) but, for both goals g_l and g_r, all three*

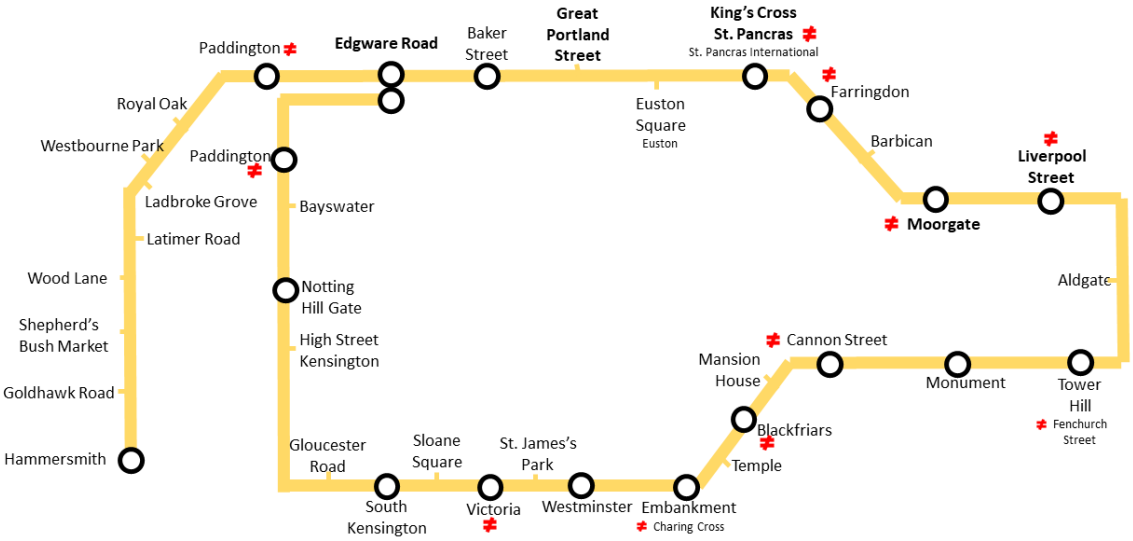


Figure 3.15: The Circle Line.

Using negative reasoning, an agent who boarded a train at Edgware Road and was observed at Great Portland Street is more likely to be travelling to Moorgate than Liverpool Street.

cost difference formulas now return zero so in all three cases, by the posterior probability calculation (RG2)*, both goals g_l and g_r appear to be equally likely (unlikely).*

The authors explain that this apparent anomaly arises because their distribution depends on an approximation whereby "probabilities corresponding to different plans for the same goal are not added up" (Ramirez & Geffner, 2010, p.1124). Our point here is not that it is unreasonable to assume that both goals are equally likely; rather that it would have been just as reasonable to assume that *all three goals* are equally likely.

We offer the following more extreme—and perhaps more commonplace—example of exclusive optimality, which arises in the context of a transport network, such as a road network or the London Underground system.[25] Exclusive optimality occurs routinely in this environment because there are frequently situations where the agent (having decided which line to travel on and in which direction) is frequently on the optimal path towards multiple goals, namely any stop on the line between the station where they boarded and the one where they will alight.

The intuition is challenging: what probabilities ought to apply? In this situation, the single-observation (and simpler) formulas—in common, in fact, with the many other approaches that similarly avoid negative reasoning (e.g., Kaminka et al., 2018; Ramirez & Geffner, 2009; Sohrabi et al., 2016; Vered & Kaminka, 2017; Vered et al., 2016)—rank all potentially optimal goals equally; but the complex formula that involves negative reasoning distinguishes between them.

[25]It is an interesting situation. We have an apparently continuous domain (the train moves continuously through the real world) even though, for practical purposes, the domain is discrete (a passenger can only exit or change direction at a station).

Example 3. *Consider Figure 3.15, which represents the Circle Line on the London Underground. Assume a passenger has boarded a train at Edgware Road with three potential goals: one at King's Cross, one at Moorgate and another at Liverpool Street. The passenger is now observed at Great Portland Street. Goal recognition using formulas* (3.1) *or* (3.2) *is unable to determine which of the potential goals is most likely: the passenger is on an optimal path to all three of them. Goal recognition using the complex formula* (RG1) *(with negative reasoning), however, assesses that King's Cross is the most likely destination, Moorgate the second likeliest, and so on. The distinction is based on lowest cost difference (highest probability) arising from the greater cost of taking an alternative route (the only alternative being to travel in an anti-clockwise direction).*

Though clearly correct, in principle, that the greater the cost of an alternative route, the more likely the observed route and therefore (by Bayes' Rule, priors being equal), the greater the probability of the associated goal, one cannot help but ask: in the case of Example 3, is this a valid conclusion?

Perhaps the question can only be decided on a case-by-case basis, depending on the particular application. Would we rather be presented with a crisp subset of all possible stations a fully rational passenger might be targeting (the result using Ramirez and Geffner's 2009 formulation and our own approach) or a ranking in line with R&G, which strictly applies Bayes' Rule and so suggests that King's Cross is more likely than Moorgate and Moorgate more likely than Liverpool Street?

3.4.2 Application of our Model in a Task-Planning Domain

Our model has three main components: (i) a *simpler formula* (3.1), which eliminates negative reasoning from Ramirez and Geffner's original approach; (ii) a *single-observation formula* (3.2), which further eliminates an agent's observation history (between the initial location and final observation); and (iii) the *RMP measure* (3.6), which states the distance from goal within which, at any point, that goal is guaranteed to be the most probable (as calculated under either of the previous formulas). Of these, only our first result, relating to the "simpler" formula—which is also used by Escudero-Martin et al. (2015) and is similar to that used by Vered et al. (2016)—is fully applicable in a task-planning domain 'off-the-shelf'.

The 'Simpler' Formula

As we saw when reformulating our solution to the continuous domain, our theorems are essentially "plug and play": if we can redefine terms whilst preserving the meaning, then the theorems hold. The simpler formula (Equations 3.1 and 3.4) differs from the original (Ramirez & Geffner, 2010) account (Equation RG1) only in one respect: it substitutes the cost of 'an optimal plan or path' for the cost of 'an optimal plan or path that does

not embed the observations'. We proved for path-planning that (in all cases, bar one) the cost of an optimal plan for a goal g that does not embed all the observed actions *is the same* as the optimal cost of a plan for g, that is, $optc^{\neg}(s, O, g) = optc(s, g)$.

Now, the key differences between Ramirez and Geffner's account of GR for task-planning and our reformulation of that account for path-planning are (a) that plans in their STRIPS-style domain are described as a sequence of actions, not states, and (b) that observations for their account of GR for task-planning are *also* actions, not states. When we generalise our simpler formula back to task-planning, neither of these differences come into play. All the related theorems deal with the costs of *complete paths* that either embed observations or do not; they never involve reasoning about individual observations or the representation of individual states. Thus, the basis on which optimal costs are calculated is not an issue, terms can be substituted and Theorems 1, 2, 3 and 4 will hold.

Arguably, in fact, the special case of exclusive optimality, characterised in Theorem 3 and discussed in Section 3.4.1 above (i.e., where there is no optimal plan for goal that does *not* embed the observations and the two formulas therefore return different results), is *even less* likely to occur in a general task-planning context. This is because, by definition, *embedding* the observed actions, involves performing them in the order they were observed. In path-planning, if an optimal path from a to b passes through points c then d then e, it cannot (optimally) pass through points e then d then c. In task-planning, on the other hand, there might often be situations where the order of actions makes no difference whatsoever to the optimality of the plan, so optimal plans not compatible with observations incidentally exist.

Example 4. *Consider a task-plan to cook pasta. The optimal plan involves (a) filling a saucepan with water; and (b) opening a packet of pasta. Reversing those actions constructs an alternative optimal plan (and so, by Theorem 2, Equations 3.1 and RG1 return the same result). That is, although there is no optimal plan for the "cooking pasta" goal that does not include (a) and (b), an alternative optimal plan can be constructed by performing them in reverse order and the special case of exclusive optimality, which seemed to pertain, does not arise.*

So, the first plank in construction of our model applies directly in a task-planning domain. Turning now to the single-observation formula and the RMP, however, the situation is somewhat different.

Single-Observation Recognition and the RMP

Our complete model, including single-observation recognition and the RMP, does apply to task-planning, *provided the particular task-planning domain conforms to strict—though unfortunately, for the most part, arguably unrealistic—assumptions.* Setting aside inher-

ited requirements (which also apply to Equations 3.1 and RG1) that the domain should be deterministic and that the observed agent should be rational (i.e., cost-sensitive), either:

- each observation must reveal a full state; or
- if observations are partial, those fluents that are observed must include *precisely the fluents necessary* in order to correctly calculate the optimal costs of reaching all goals.[26]

If either one of these conditions can be met, the single-observation formula is also 'plug and play'. The conditions are necessary because, unlike Equation (3.1), this formula does reason, not only about complete plans, but about individual observations. The "single-observation" in the formula's name is taken to be the observed agent's most recent *complete* state and, on that basis, the system is able to infer everything that has changed since the plan began.

In Ramirez and Geffner's model of GR, observations are actions which, by themselves, simply do not provide enough information. Even in a strict path-planning context, a sequence of observed actions such as "turn left; walk 5 metres; turn right," would not reveal the current location (e.g., other actions may have occurred after the agent turned left and before she walked 5 metres). So, a final observation "turn left" would only tell us that the agent ought to be in location that can be reached by turning left from somewhere else! Thus, whereas in the path-planning context of our model, an observation reveals the location/state of the agent fully, in Ramirez and Geffner's model for task-planning, a single observation reveals a set of all possible states the agent may be in.

For the single-observation formula to apply, the optimal path from the initial state to the state reached at the final observation must have the same optimal cost no matter which goal the agent is pursuing. Clearly, if the state at the final observation is one of a *family* of possible states, the optimal cost of reaching it is unknown and *may be different depending on which goal is being targeted.*

Example 5. *Consider a task-planning domain with two candidate goals: shoot the ambassador or go on holiday. There are two observed actions: loads gun; arrives airport. Using the single-observation formula, which considers only the final observation that the agent is at the airport, there is a family of possible current states ("at airport with gun", "at airport with suitcase") but no way to distinguish between them. Using Ramirez and Geffner's (or our simpler) formula, however, since the agent was observed loading the gun, the cost difference between optimal and observed plans is greater for the holiday (no suitcase was observed), making assassination the more likely goal.*

[26] In fact, the first condition is a conservative assumption, which guarantees the second condition. The first condition may be true in special cases (such as path-planning) but unrealistic for general task-planning. The second condition is all that is really required but is much more difficult to specify (see Example 7).

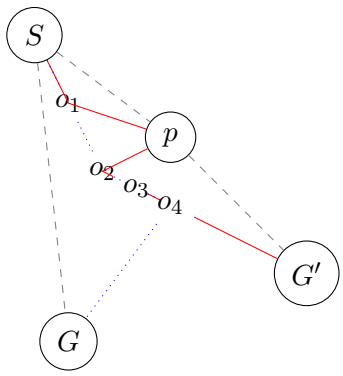


Figure 3.16: An 'enhanced' path-planning scenario. The optimal paths from S to o_4 that embed observations are different for each goal.

A further example shows that, even when path-planning is only slightly 'enhanced' (i.e., much less rich than general task-planning), the single-observation formula may be unreliable.

Example 6. *In the scenario depicted by Figure 3.16, a vehicle sets off from S with* 10 *litres of petrol. It is observed several times, the last time at point o_4. It only takes* 8 *litres of petrol to reach goal G but will need* 12 *litres to reach goal G' (perhaps it is uphill). Thus, assuming cost is measured by time or distance, the optimal route to G' that embeds observations involves driving first to the petrol station at p (the route marked in red); but the optimal route through the observations to G (marked in blue) is direct. Since the optimal path to each goal through the observations is* different *—and has a different cost—the single-observation formula cannot be applied.*

Now, looking at either of the above counter-examples in the light of our two conditions (which would enable application of our model to task-planning) note that, if we were able to *accurately* ascertain the optimal cost from the initial state to the most recently observed action—and so whittle the family of possible states back down to one[27]—the single-observation formula *could* be applied. All that is required is for the final observation to be one that reveals a fully observable state (or rather, reveals all aspects of the state relevant to the candidate goals).

So, in Example 5, the final observation must reveal what the agent is carrying or the result, perhaps, of them having passed through a metal detector. In Example 6, we need the vehicle to be equipped with telematics so that the final observation can reveal, not only $at(o_4)$, but how much petrol is in the tank. If the optimal cost from initial state to final observation can be accurately calculated then it can be applied equally to both goals: the costs cancel out and Theorem 5 applies. The problem of determining which particular fluents need to be observed, however, is a non-trivial task.

[27]Note that, in practice, for our formula, it is not strictly necessary to know the state itself but rather the cost of reaching that state and the optimal cost of getting from that state to each goal.

Example 7. *In a kitchen domain which, for argument's sake, includes only one glass, an agent has two possible goals: to drink a glass of milk or to drink a cup of tea. A full execution trace reveals that the agent takes a glass, drops it (causing it to break), then goes to the fridge and takes out a carton of milk. Now, a GR system in a domain with full observability—or one with partial observability that includes an observation of the glass being dropped—gives zero probability to the goal of drinking a glass of milk but a GR model that bases its prediction on the final observation only is unaware of the breakage and so gives equal (or equal to prior) probability to each goal based on the carton of milk being removed from the fridge.*

There are various methods that might be employed to determine precisely which fluents need to be observed in order to disambiguate between any set of particular goals. However, it is a problem that opens a whole new area of investigation. It is beyond the scope of this thesis but suggests a most interesting avenue for future work.

3.4.3 Implications for 'Plan' Recognition

So far, we have treated GR as if it were synonymous with plan recognition. It is not unreasonable: Sukthankar et al. (2014) point out that the two things are typically taken together; furthermore, our formalism is based on the work of Ramirez and Geffner, who refer to the framework with the expression "plan recognition as planning" when it too focuses on goals.

Strictly, though, GR is just one aspect of plan recognition and provides much less information. Given that goal and plan recognition are typically used to inform some higher purpose, the difference may be significant. Although GR can tell us the end result (or destination) to ultimately prepare for, it cannot tell us, as plan recognition does, which step in the plan is likely to come next. Thus, if we wanted to facilitate (or intercept) the agent under observation, GR alone may be of limited use.

Example 8. *Consider a traveller making a journey from Melbourne, Australia to one of two possible destinations: Sydney to the north-east or Adelaide to the west. A domain expert hand-crafts a plan library of half a dozen routes, which includes: direct routes via the highways; scenic routes via the coast; and inland "heritage" routes via the back roads. The traveller is observed at Lakes Entrance (on the coast, east of Melbourne). Given the observation, a cost-based GR system correctly identifies Sydney as the most likely goal—based on the optimal cost from Lakes Entrance to Sydney. A* plan *recognition system, on the other hand, correctly identifies one particular "scenic route via the coast to Sydney". Either system, when interrogated, gives Sydney as the most likely goal but, if queried for the traveller's most likely next town or* subgoal*, the plan recognition system suggests Eden (on the coast) whereas the GR system suggests Bombala or Jincumbilly, that is, whatever*

town happens to be on the optimal path back towards Sydney, no matter how unlikely it would be for a traveller to deliberately choose such a route.

Our model—and *any* cost-based model that returns a goal but not a plan—would make the same mistake as the GR system in Example 8. Such models can only assume that the agent will now follow an optimal path from the currently observed location to the most likely goal. This is a clear case, however, where consideration of *all* observations could be advantageous. If the full sequence of observations had been processed, instead of assuming the traveller would now take the optimal path to goal, the system might: (a) assume a path to goal with a more or less equivalent degree of suboptimality to that which had so far been observed; or (b) generate a more sophisticated cost function to prioritise the types of locations so far encountered (e.g., 'coastal', 'inland', 'picturesque', etc.).

Either option would be a nice refinement. To our knowledge, however, no contemporary plan recognition system takes all observations into account in this way so, although this is an interesting avenue for future work, the available tactic under our model (i.e., the assumption that the next step will be on the optimal path to the most likely goal) remains a practical and competitive solution.

3.5 Summary

In this chapter, we have taken the 'plan recognition as planning' approach to probabilistic GR first introduced by Ramirez and Geffner (2009; 2010), and applied it in the context of path-planning. In particular, we have focused on how this cost-based approach to GR, which they pioneered, can be exploited in core navigational domains to minimise the computational effort required to determine an agent's most probable destination.

We started by showing that a simpler cost difference formula (3.1)—which does not require negative reasoning about observations and can be achieved by calls to a standard path-planning algorithm—yields an identical result to the original more complex Ramirez and Geffner formula (RG1), which does reason negatively, in all but one specific case, which we characterised and discussed. We found, in fact that, even when the two formulas return different results, that rarely results in different goal rankings; and, where rankings do differ, the usefulness of the R&G ranking (which distinguishes between differently 'optimal' goals) is debatable. We demonstrated an even simpler single-observation cost difference formula (3.2) that does not even depend on the observation sequence but nevertheless generates a posterior probability distribution which exactly preserves the goal rankings from the simplified account and, by extension, results in an identical ordering to formula (RG1) in all cases bar one.

Because the single-observation formula is independent of the observation sequence, it has the benefit that, not only can it be used *online* for on-the-spot checks of agents who have been observed entering but whose movement history is otherwise unknown; it can

also be pre-computed *offline* in any domain for which the start location and candidate goals are known in advance (e.g., doors to a building and rooms that require protection) to create a probabilistic heatmap against which *online* checks can be made in constant time.

We generalised our solution to the continuous domain and used that framework to demonstrate calculation of the RMP (3.6), significant because it formalises the relationship between position and probability (implicit in cost difference formula 3.2) by calculating the cost-distance radius within which a given goal is guaranteed to be the most probable. Moreover, it does this without having to calculate any probabilities and therefore in a fraction of the time that it would take to generate a probabilistic heatmap using the single-observation formula. The RMP formula can also be used offline as a tool complementary to goal recognition design. Alternatively, since it is quickly calculable, it can be used online to find RMPs, for example in games that automatically generate new (previously unknown) terrains.

It seems that we can look at the results from this chapter in one of two ways, depending on whether we are in a glass-half-full or a glass-half-empty frame of mind. On the one hand, the revelation that goal rankings from state-of-the-art GR can be determined in an almost entirely domain-dependent (observation-free) process might be regarded as a massive boon: for faster spot-checks, streamlined goal recognition design and so on, just as we have described it up to this point. On the other hand, these undoubtedly useful results raise the unsettling possibility that state-of-the-art GR may be 'missing a trick'. Is it really okay to dispense with all those observations? Surely some useful information could be extracted from an agent's complete movement history. If not—if the agent's most recent location really is the overriding factor in determining the probability of her goal—might that not make the GR system rather easy to manipulate, somewhat vulnerable to deception? These are the possibilities that we explore in Part II.

In summary, this chapter made the following contributions.

- We proved that a simpler cost difference formula plugged into the R&G probability distribution formula (Ramirez & Geffner, 2010) returns the same result as their more complex formula in all but one set of conditions and in less than half the time.
- We presented an alternative single-observation cost difference formula, which can be used to rank goals in exactly the same order as the simpler formula, based on just one observation (i.e., where the agent is 'now').
- We showed that single-observation recognition may be applicable to general task-planning in some restricted scenarios; more work will be needed to precisely specify the constraints.

- We suggested that the single-observation formula could be used to generate a probabilistic heatmap (relative to a known starting point), implying that probabilities under the R&G model are domain- rather than observation- dependent.
- Building on the notion of single-observation recognition, we developed a novel method of GR which we call the 'radius of maximum probability' (RMP). The RMP is a radius from goal within which that goal is guaranteed to be the most probable (under the R&G model). The calculation is based on similar information and takes a comparable number of calls to the planner as calculation of a single probability distribution using other methods.
- We showed that single-observation recognition and the RMP are applicable in discrete graph-based domains and in continuous domains, although application of the RMP in the discrete domain comes with certain caveats.

Part II

CHAPTER 4

Deception as Path-Planning[†]

"Rigorous training tends to lead to straight thinking. Prideful of their abilities in observation and deductive reasoning, [scientists and the like] do not realize that this straight-line approach is totally inappropriate, even dysfunctional, when used to detect deception."
–J. Barton Bowyer

Deception is "the advantageous distortion of perceived reality" (Bowyer, 1982, p.47), a slippery topic, no doubt; but grounded in path-planning it becomes easier to grasp.

Deceptive path-planning (DPP) is the problem of finding a path through a two-dimensional map or three-dimensional navigational space such that an observer, watching an agent make her way along that path, will be *unable to determine*—until the last possible moment—where it is that the agent is going. Implicitly, the observer is performing goal recognition (GR) but, compared with its treatment in Part I, this chapter flips the problem on its head. In Part I, we developed a quick and practical method of GR in the context of path-planning; In Part II, we look for a means to subvert it.

Consider an airport surveillance system (similar to the one previously encountered in a GR context, p.41). An agent of interest is observed entering the domain. If it is determined that she is approaching some particular boarding gate, a flag will be raised to trigger her interception. The agent is not labelled and the system has no superpowers to assess that it is now dealing with a deceptive agent so, as before, GR proceeds in line with keyhole recognition (which, recall, assumes the agent is unaware of the fact that she is being observed). Nevertheless, suppose that, in this case, the agent is deceptive and suppose that she behaves in the way suggested by one of Jian et al.'s subjects who, when challenged to draw a trajectory that would deceive an onlooker, said:

[†]Some of the work in this chapter has been published elsewhere (Masters & Sardina, 2017b, 2019b).

> *[I was] trying to get to a target with the most confusing way possible which was basically doodling lines all over the place.* (Jian et al., 2006, p.1567)

In this situation, state-of-the-art GR tracks the deceptive agent's movements exactly as we have previously seen, accumulating observations and calculating probabilities as she wanders "all over the place" with the effect that—in the absence of meta-reasoning to notice that the observed agent favours now this goal, now that goal—it returns apparently conclusive results (e.g., Goal A with probability 0.75) even though two or three observations later the result is reversed (e.g., Goal B with probability 0.8). As feared (and flagged) at the end of Part I, it appears that, although observations *seem* to be considered, much of the information they ought to impart is being ignored. Furthermore, if this is the agent's best attempt at deception, she has chosen a pitifully uneconomical way of going about it.

In this chapter, we tackle both aspects of the problem. First, we present an alternative model of GR, a parsimonious elaboration of contemporary frameworks, but one which degrades gracefully in the face of increasing suboptimality, seeming to 'know that it does not know' rather than keep changing its mind. Then, acknowledging that a deceiver taking this approach has been successful up to a point—her observer is unable to determine, after all, which goal is her true target—we present a model of deceptive path-planning (DPP), building on Bell and Whaley's general theory of deception (Bowyer, 1982), which reveals more economical ways of achieving arguably better results.

The rest of this chapter is organised as follows. Section 4.1 presents our self-modulating formula for GR. We first analyse the behaviour of contemporary cost-based GR in situations where the observed behaviour seems to be becoming increasingly irrational, then set out the formula itself, demonstrating its ability to deal with the manifestation of irrationality in a principled way. Section 4.2 presents a more disciplined approach to deception: our formal model for DPP, by reference to which, instead of wandering all over the place, an agent has the option of choosing a rational path that may be even more effective in disguising her destination. Section 4.3 proposes and evaluates deceptive strategies that use our own single-observation recognition from Part I to model the observer. Section 4.4 discusses key issues, including consideration of 'truthful' path-planning, that is, the inversion of our model for deception to instead facilitate *intended* recognition.

4.1 A Self-Modulating Formula

> *There are known knowns . . . things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know.* Donald Rumsfeld

Deception can be achieved by 'simulation' (showing the false) or 'dissimulation' (hiding the true); and dissimulation in one of three ways: by masking, repackaging or dazzling (Bowyer, 1982). While all these strategies can conceivably be applied in a path-planning context (as discussed later in this chapter, p.131), in our minimalist setting, 'dazzling'—that is, the wandering behaviour described by Jian et al.'s subject (p.96), which involves confusing the observer with false positives—seems a particularly promising approach, enabling a deceptive agent to hide in plain sight even when her observer knows that she is there.

Dazzling involves deliberate obfuscation but manifests as cost-inefficient, irrational behaviour. Given that the strategy is likely to be attractive to a deceiver, it is important to understand how contemporary GR systems respond in the face of it and to consider whether (and how) that behaviour might be improved.

Note. This sort of apparent irrationality does not necessarily signal deception. It might also occur because the observed agent is *actually* irrational (e.g., mad or drunk) or it may simply be that she is operating under a different cost model from that used by the GR system observing her (i.e., she seems irrational but it is just that she has different priorities from those we were expecting).

Technically, the problem for GR is that contemporary cost-based systems are strongly predicated on the assumption of rationality, which carries with it a subsidiary assumption that the observed agent is honest; and this represents a significant, exploitable limitation.[1] On discovering that she is not (if, for example, the behaviour increasingly deviates from any optimal plan), one would expect an intelligent system to become more 'agnostic', that is, less confident in its predictions until, in the extreme case—confronted by wildly suboptimal or intentionally deceptive behaviour—it judges all goals to be equally probable (or reverts to prior probabilities). Here, however, we analyse two state-of-the-art GR frameworks—that of Ramirez and Geffner (2010), extended not only by us (in Part I) but by many other authors (e.g., Escudero-Martin et al., 2015; Shvo, Sohrabi, & McIlraith, 2018; Sohrabi et al., 2016), and that of Vered et al. (2016), discussed in Chapter 2 (p.18)—and find that they do not achieve this. We find in fact that, when faced with suboptimality in some clear cases, both of these state-of-the-art cost-based GR systems yield unexpectedly anomalous results.

In this section, we present an alternative and, arguably, more intelligent solution: a self-modulating probability distribution formula, which lifts the assumption of rationality that can make otherwise effective state-of-the-art GR systems vulnerable to decep-

[1] The same significant and exploitable limitation is also present in humans, who overwhelmingly assume honesty (Levine, 2014b) and cost-sensitive intentionality (Baker et al., 2009; Bonchek-Dokow & Kaminka, 2014), much to the frustration of behavioural economists who remind us that much of the behaviour we observe results from unconscious impulses unrelated to conscious intent (Kahneman, 2011).

tion. The self-modulating formula registers suboptimal behaviour and adjusts its level of confidence accordingly. The new formula handles observations at both ends of the spectrum—whether corresponding to optimal, suboptimal or totally random paths—in a principled manner, automatically modulating the shape of the probability distribution as it becomes apparent that the usual assumption of cost-sensitivity does not hold. That is, the more irrational (or 'deceptive by dazzing') the observed behaviour becomes, the more the distribution evens out.

Practically, use of a self-modulating formula can protect a system in three ways: (i) by preventing it from naively jumping to unwarranted conclusions; (ii) by avoiding oscillation between incompatible decisions; and (iii) by flagging the possibility of deceptive intent.

At the technical level, our self-modulating formula builds upon and improves the probability distribution formulas used by Ramirez and Geffner (2010) and Vered et al. (2016), which were originally developed for use in disparate domains: one for STRIPS-style task-planning, the other for continuous online motion-planning. In order to analyse these (and other) models on an equal footing, we next abstract our problem definition for GR to facilitate direct comparison between them.

4.1.1 Technical Framework: Cost-Based GR (Generic)

The GR accounts presented in Part I (for the discrete domain at Section 3.1.1, p.44, for the continuous domain at Section 3.2.1, p.62) were specific to path-planning. Of course, GR occurs in multiple other settings. Therefore, for convenience, we introduce the notion of a *generic* cost-based probabilistic GR problem whose components have different meanings and structures depending on the interpretation given to the domain, represented by $\mathcal{D}$.

Definition 8. *A **generic cost-based GR problem** is a tuple $\mathcal{R}_g = \langle \mathcal{D}, O, s_s, G, \vec{o}, Prob \rangle$ where: $\mathcal{D}$ is a model of the GR domain (which defines states, transitions between states and the cost of transitions); O is the set of observable elements in $\mathcal{D}$; s_s is the initial state; G is the set of candidate goals; $\vec{o}$ is the sequence of observations drawn from O; and $Prob$ is the prior probability distribution across G.*

We are particularly concerned with the approaches of Ramirez and Geffner (2010) (abbreviated to R&G, as before) and Vered et al. (2016) (now abbreviated to V&K).

- In R&G, which is geared to task-planning, $\mathcal{D}$ represents a STRIPS-like domain of fluents and actions where each action has an associated precondition, add and delete list and is individually costed (as set out in Section 2.5, p.35). Grounding $\mathcal{R}_g$ to R&G, O is the set of actions, s_s is a state under $\mathcal{D}$, G is a set of the usual planning goals (i.e., each $g \in G$ is a conjunction of literals) and $\vec{o}$ is a sequence of actions.

- In V&K, which is concerned primarily with continuous motion-planning, the domain is conceived as a multi-dimensional Euclidean space, $\mathcal{D} \subseteq \mathbb{R}^n, n \geq 2$, typically used to

represent two- and three- dimensional map or real-world locations but also capable of representing any number of additional continuous dimensions such as pose, velocity or colour (discussed at p.18). Grounding $\mathcal{R}_g$ to V&K, O is a (potentially infinite) set of points and transitions through $\mathcal{D}$, s_s is a state (a subset of $\mathcal{D}$), G is a set of such states, $\vec{o}$ is a sequence of points and trajectories in Euclidean space, obtained as the range of a time function $f : \vec{t} \mapsto O$, where $\vec{t}$ is a sequence of time intervals during which the world has been observed.[2]

Generically, a plan π in $\mathcal{D}$ is a sequence of elements that imply changes to the underlying domain, transforming it from one state to another.[3] For R&G, a plan is as usual a sequence of actions and its cost is the combined cost of all actions in the plan. For V&K, meanwhile, a plan is modelled as a sequence of states and state transformations, where (similar to the mechanism described in relation to observations above) each element in the sequence is obtained by application of a time function. Thus, although for V&K the space within which the problem plays out is itself continuous, a plan in the space can be conceived as the concatenation of a discretised sequence of states and trajectories (or multi-dimensional transformations) through which it must pass. The cost of a plan in V&K is obtained by reference to a distance metric (i.e., represents the plan's total length).

As previously, a plan is said to *embed a sequence of observations* if the observations can be mapped to the plan in such a way that the order of elements (in both) is preserved. That is, given a plan $\pi = e_1, \ldots, e_m$ and observations $\vec{o} = o_1, \cdots, o_n$, there must exist a monotonic function $f : \{1, \ldots, n\} \mapsto \{1, \ldots, m\}$ such that $e_{f(i)} = o_i$ for all $i \in \{1, \ldots, n\}$.[4]

Though calculated differently in different domains, the notion of optimal cost has its expected meaning of a minimum cost plan between any definable initial and terminating set of conditions. In line with Part I notation, the optimal cost of a plan from the initial state s_s to a goal g is denoted $optc(s_s, g)$ and the optimal cost of a plan from s_s to g embedding observations $\vec{o}$ is denoted $optc(s_s, \vec{o}, g)$. The optimal cost of a plan from s_s to g that does *not* embed observations $\vec{o}$ is given by $optc^{\neg}(s_s, \vec{o}, g)$. The meaning of this negative construction has been discussed previously (see Section 3.4.1, p.80) and is amplified below. Briefly, if $c(\pi)$ is the cost of a plan π, $\Pi(s_s, g)$ is the set of all plans from s_s to g and $\Pi^o(s_s, g)$ is the set of all plans from s_s to g that embed observations $\vec{o}$ then $optc^{\neg}(s_s, \vec{o}, g) = \min_{\pi \in \Pi(s_s,g) \setminus \Pi^o(s_s,g)} c(\pi)$.

As expected, the solution to a generic GR problem $\mathcal{R}_g$ is a posterior probability distribution $P(G \mid \vec{o})$ which prefers those goals whose plans best satisfy the observations $\vec{o}$, that is, plans that embed the observations at least additional cost when compared with

[2] We used a similar function to obtain observations for continuous goal recognition (see p.64).

[3] In this generic framework, we refer now to plans in general rather than paths in particular.

[4] Observations may not always be so closely related to plans; the mapping holds, however, for all domains under consideration here. A comparable result may be achieved for other domains by modifying the function. For example, Sohrabi et al. (2016) treats observations as observable fluents (not actions), which map into the states of a plan's execution trace rather than mapping to actions in the plan itself.

the cost of an optimal plan for the same goal. Intuitively, the more closely observations conform to the optimal plan for a goal $g \in G$, the more likely it is that goal g is being pursued. Models vary, however, about the preferred method of performing the comparison.

Approaches to Obtaining Posterior Probabilities

Recall from Chapter 2 (p.35) that the R&G probability distribution formula derives from Bayes' Rule and makes (amongst others) the following assumptions: (i) that the probability of a plan is inversely proportional to its cost; and (ii) that probabilities for multiple plans for the same goal can be said to be dominated by the highest of those probabilities. The first assumption is central to the model and is encapsulated in the notion of cost difference (discussed in detail in Part I and exploited by us to achieve Single-Observation Recognition, 3.1, p.42).

The R&G formulation relies on comparing, for each goal, the cost difference between the cheapest plan, given the actions already observed, and the cheapest plan that could have achieved the goal if one or more of those actions had not been taken (see Figure 3.3, p.50). If, for example, an agent seated in her armchair has several possible goals, one of which is to mow the lawn, and she is observed playing with the dog, then the cost difference with respect to lawn-mowing is the cost of getting up, playing with the dog *and* mowing the lawn (i.e., the best she can do, given what has been observed) less the cost of just getting up and mowing the lawn (i.e., the best she could have done if one or more of the observed actions had *not* occurred). We remind the reader that we have already proved that the latter cost $optc^{\neg}(s_s, \vec{o}, g)$, which involves negative reasoning, is identical to the optimal cost from initial state to goal, $optc(s_s, g)$, in all cases bar one (Theorem 3, p.52) and *always* identical when $\vec{o}$ conforms to a suboptimal path (Theorem 1, p.50), which is the situation of interest here.

For convenience, we reiterate R&G's cost difference formula (our baseline cost difference formula (RG1) from Part I), using the slightly modified $\mathcal{R}_g$ notation as follows:

$$costdif(s_s, g, \vec{o}) = optc(s_s, \vec{o}, g) - optc^{\neg}(s_s, \vec{o}, g). \tag{4.1}$$

Again, recall that Ramirez and Geffner's key intuition is that any solution to $\mathcal{R}_g$ should have the property that *the lower the cost difference for a particular goal, the higher its probability* and that they achieve this by plugging the cost difference parameters into a Boltzmann equation. In Part I, we reformulated their probability distribution equation and used it as a template into which we could insert various cost difference formulas (p.48). Here, we present it as Ramirez (2012) describes (i.e., retaining the negative temperature parameter β). Using the slightly modified notation of our generic representation, probabilities are derived as follows:

$$P_{RG}(G \mid \vec{o}\,) = \alpha \cdot \frac{1}{1 + e^{-\beta(optc^{\neg}(s_s,\vec{o},g) - optc(s_s,\vec{o},g))}} \tag{4.2}$$

where α is a normalising constant and β is a positive constant (default = 1).[5]

Note. β is a rate—or 'temperature'—parameter, variation of which modifies the distribution in such a way that, as β approaches zero, the distribution flattens out. It is this rate parameter that we exploit later on.

Seeking a similar outcome in the continuous domain and concerned particularly with GR in an online motion-planning environment (e.g., while observing a human agent drawing a number on a piece of paper, where observations are assumed to be revealed incrementally and GR is an iterative, rather than a one-off, process), Vered et al. (2016) take a different approach. Whereas R&G derives its probability distribution formula from Bayes' Rule, V&K appeals to empirical evidence to support use of a ratio between (a) optimal cost and (b) optimal cost embedding the observations. They characterise their probability formula as an heuristic and, indeed, supporting evidence demonstrates that it was the best performing of three competing heuristics for intent recognition when compared with human performance (Bonchek-Dokow & Kaminka, 2014).

An intuitive basis to the V&K solution is given as follows.

> *The underlying assumption . . . is that the ideal plan is optimal; if the observed plan is far from the ideal plan, then the agent must not be rational, and is likely pursuing an alternative goal altogether.* (Vered et al., 2016, p.5)

Concretely, the probability distribution across G in V&K is based on a simple ratio between the costs of (a) an optimal plan (e.g., a perfectly formed 7, 8 or other number, costed by length of line) and (b) an optimal plan that embeds the observations (e.g., whatever was actually drawn), articulated in terms of R_g as:

$$P_{VK}(G \mid \vec{o}\,) = \alpha \cdot \frac{optc(s_s, g)}{optc(s_s, \vec{o}, g)}. \tag{4.3}$$

Having defined the probablity distributions of interest, we next examine the performance of formulas (4.2) and (4.3) more closely. In doing so, we distinguish between ***scores***, that is, the likelihoods calculated *before* normalisation (which may sum to any value) and ***probability values***, that is, the normalised results (which, after multiplication by a constant α, sum to 1).

Note. Deception does not always have a nefarious purpose. It may be employed by an agent for no reason other than to keep her personal intentions private.

[5] This thesis uses various formulations of the Boltzmann equation, all provably equivalent (see Appendix A).

4.1.2 The Rationality Assumption

As explored in Part I, the intuition underlying cost-based GR rests on the assumption of rationality: the more closely an agent is following an optimal plan for A, the more likely it is that A is her intended objective. Published empirical results indicate that, with reasonable plans (i.e., plans that stray not *too* far from the optimal), formulas (4.2) and (4.3) give reasonable results, as good or better than competing offerings (Ramirez & Geffner, 2010; Vered et al., 2016). Problems arise, however, when the rationality assumption breaks down. This is because both frameworks are *based* on rationality but, faced with its absence, neither of them take that assumption into account. Unsurprisingly, unintended consequences arise.

What is Irrational in a GR Domain?

In the context of a GR problem, how exactly should the rationality of a partial plan (or sequence of observations) be defined? Normally, we would say that the less rational plan is the one that is more expensive with respect to the real goal but, in a GR scenario, the ground truth is unknown. The fact that observations seem to suggest a plan that is irrational (suboptimal) with respect to any one particular goal actually tells us very little. When an agent pursues a particular goal, we expect observations to reflect a more-or-less optimal plan for that goal. It stands to reason that the closer the agent is to achieving one goal, the more suboptimal her actions are likely to become with respect to all the others.

Consider, for example, a cooking domain with three candidate goals: A, fried eggs, B, boiled potatoes or C, chicken soup. Now, an agent observed peeling potatoes and filling a pan with water is following a more or less optimal plan for goal B but an increasingly suboptimal one for goals A and C. Is the plan rational or irrational? truthful or deceptive? Without knowing the real goal, it seems we cannot answer the question.

Consider an alternative sequence of observations, however, where the agent is observed heating the oven: a meaningful action in itself, but irrelevant to all three goals! In this case, without needing to know the ground truth, we can confidently describe the observed behaviour as irrational or deceptive: whatever the goal, it is suboptimal. Similarly, if an agent behaves at one moment as if attempting to achieve goal A, at the next goal B, and so on (e.g., by getting out the frying pan, peeling potatoes and opening a can of soup), now her actions have again become suboptimal with respect to all the goals and, again, palpably irrational (or deceptive).

This is the behaviour that we are interested in: behaviour that indisputably betrays irrationality even though the ground truth is unknown.[6]

[6] This is related to epistemic notions of belief and knowledge under 'possible world' semantics (Hintikka, 1962): we believe a plan is irrational if it is irrational *in every possible world* (i.e., for every goal).

Definition 9. *Observation sequence* $\vec{o}' \in O^*$ *is* ***less rational*** *than* $\vec{o} \in O^*$ *iff for all* $g \in G$, $optc(s_s, \vec{o}', g) > optc(s_s, \vec{o}, g)$.

In words, if one plan (via $\vec{o}'$) costs more than another plan (via $\vec{o}$) *no matter which goal is being pursued*, then it is less rational to select the more expensive plan.

For the purpose of demonstrating the limitations of existing systems, we now extend the above definition to describe the special case where an observation sequence is not only less rational than another observation sequence but less rational *by the same degree for all goals*, as follows.

Definition 10. *Observation sequence* $\vec{o}' \in O^*$ *is* ***uniformly less rational*** *than* $\vec{o} \in O^*$ *iff:*

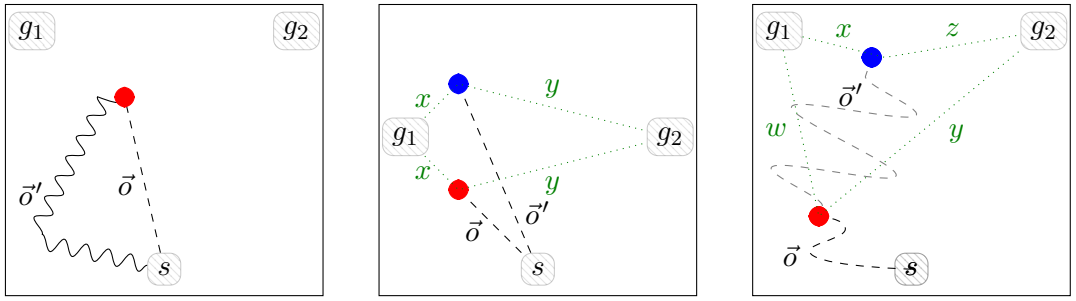
1. $\vec{o}'$ *is less rational than* $\vec{o}$*; and*
2. *for all* $g_1, g_2 \in G$, $optc(s_s, \vec{o}', g_1) - optc(s_s, \vec{o}, g_1) = optc(s_s, \vec{o}', g_2) - optc(s_s, \vec{o}, g_2)$.

Under this definition, we distinguish three distinct classes of uniformly irrational behaviour, as follows.

(a) **Equivalence.** Figure 4.1a depicts two different observation sequences $\vec{o}$ and $\vec{o}'$ with a shared starting point s. Both observation sequences end up in exactly the same state. We say that, in this situation, the observation sequences are equivalent and that the one that costs more is 'uniformly less rational' than the other. This conforms to our usual understanding: if there are two different plans for reaching one state from another, it is less rational to use the plan that costs more.

(b) **Goal-cost equivalence.** Figure 4.1b shows a situation where two different observation sequences $\vec{o}$ and $\vec{o}'$ end up in *different* states (red and blue); nevertheless, the *cost* from either state to each goal is exactly the same (x to reach g_1 and y to reach g_2). In this case we say that the observation sequences are 'goal-cost equivalent' and, in such cases, the more expensive observations $\vec{o}'$ are again uniformly less rational than observations $\vec{o}$.

(c) **Relative equivalence.** In Figure 4.1c, there are two different observation sequences ($\vec{o}$ ending at the red node, $\vec{o}'$ continuing to the blue node) such that an optimal plan which satisfies them ends up in *different* states (red and blue). Now, even though an agent following $\vec{o}'$ has made progress towards both goals (and ends up closer to g_1), progress towards each goal has actually been the same ($w - x = y - z$) and at a greater cost than if the agent had only followed $\vec{o}$ as far as the red node. That is, an optimal path via $\vec{o}'$ costs more than an optimal path via $\vec{o}$ *by the same amount*, regardless of which goal is being considered. In this situation, the observation sequences are 'relatively equivalent' and, again, the one that costs more is uniformly less rational than the other.



(a) Equivalence: observations end up in *the same state.*

(b) Goal-cost equivalence: observations end up in *different states* but the *cost from either state to each goal is the same.*

(c) Relative equivalence: $(w - x = y - z)$ observations progress (or *fail* to progress) towards both goals by the same amount.

Figure 4.1: Irrationality in a GR domain. Whatever the goal, observations $\vec{o}'$ are uniformly less rational than observations $\vec{o}$. Dotted lines marked w, x, y and z represent costs from the states reached (red and blue) when following an optimal plan via the observations to each goal. Note that although, for convenience, the diagrams imply path-planning, the notions are generally applicable.

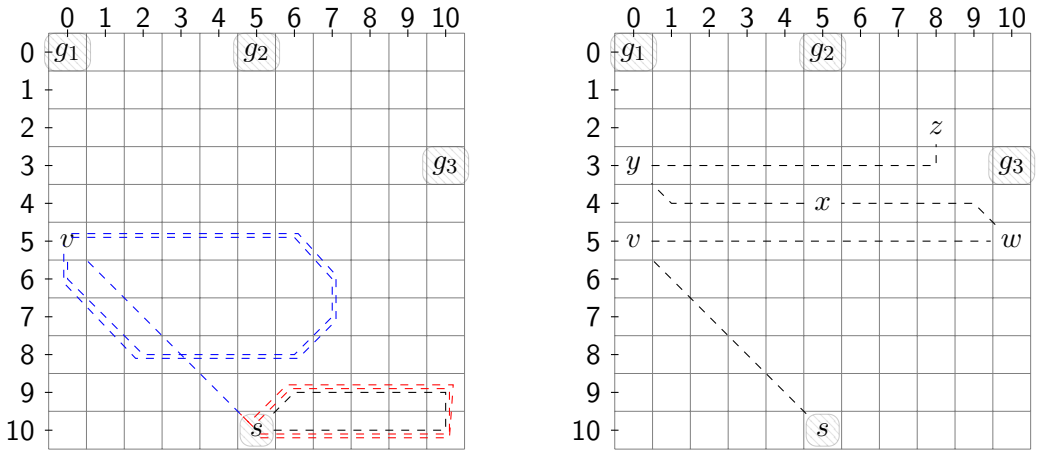
Our description of relative equivalence is a literal rewording of Definition 10. It most obviously arises if the uniformly less optimal plan zigzags (as in Figure 4.1c), advancing on all goals without favouring any one in particular; but notice that the definition actually subsumes both equivalence and goal-cost equivalence (i.e., cases (a) and (b), above).

With these classes in mind, we next look at how R&G and V&K respond.

GR and Irrationality

Consider the navigational scenarios depicted at Figure 4.2. An agent is observed in a gridworld domain with three goals $G = \{g_1, g_2, g_3\}$. In terms of a GR problem $\mathcal{R}_g$, the initial state, s_s, goals G and components of a path O are all possible grid locations (cells). We assume that a path is costed in terms of transitions between adjacent cells, that horizontal and vertical transitions cost 1, diagonal transitions cost $\sqrt{2}$. The bottom right (red) path shows an agent first observed at its initial state s ($s_s = s$), then moving in two loops; that is, instead of progressing it returns to the cell at s each time. The other (blue) path depicts an agent setting off on an apparently optimal path towards goal g_1. Having reached location v, however, instead of continuing on, it loops twice, returning to v each time. In the second diagram 4.2b, the agent again begins on an apparently optimal path towards g_1 via cell v but this time veers off to w, then takes an increasingly irrational route via x, y and z (final destination unknown).

Table 4.1 shows the probability values for each goal on each visit to s and v, with an additional result given for the case where the loop returning to v repeats 10 times. There are three main columns: two for the R&G model, each with different β values (the lower



(a) Loopy paths.

(b) Zigzagging path.

Figure 4.2: Suboptimal paths as traversed by an apparently irrational agent.

β results in a flatter distribution overall); and one for the V&K model (which has no rate parameter). The results are interesting and not necessarily as one would expect.

First, excepting the corner case where the only observation is the initial state s_s,[7] we see that P_{RG} as per Equation (4.2) evaluates probabilities for paths that repeatedly return to the initial state as equal for all goals (i.e., equivalent to priors). This seems reasonable (though it is not true for V&K, as we discuss shortly). A discrepency occurs with respect to R&G, however, if paths track first to v and then loop. Now, whichever goal was most probable on the first visit *becomes more probable at each subsequent visit.*

It turns out that this anomalous situation is not just an issue with 'looping', but with the more general case of uniformly less rational observed behaviour, which includes meaningless noise, such as looping, as a special case. Indeed, as Theorem 8 shows, given two observation sequences $\vec{o}$ and $\vec{o}'$, goal recognition under equation (4.2) becomes *more* confident under the *less* rational observation sequence $\vec{o}'$.

> If two observation sequences are goal-cost equivalent, R&G is *more* confident under the *less* rational observations.

Theorem 8. *Let $\vec{o}, \vec{o}' \in O^*$ be two observation sequences such that $\vec{o}'$ is uniformly less rational than $\vec{o}$ and let $\hat{g} \in G$ be such that $P_{RG}(\hat{g} \mid \vec{o}) > P_{RG}(g \mid \vec{o})$, for all $g \in G \setminus \{\hat{g}\}$ (i.e., goal $\hat{g}$ is the best explanation under the more rational observations $\vec{o}$). Then, $P_{RG}(\hat{g} \mid \vec{o}') > P_{RG}(\hat{g} \mid \vec{o})$.*

Proof. The effect is a by-product of normalising scores generated using the Boltzmann distribution.

[7] This is an example of the extreme case described in Part I at p.55. Owing to the negative reasoning in cost difference equation (4.1), if there exists a goal g such that *every* path to g satisfies the observations, cost difference may evaluate to $-\infty$ yielding an undefined normalised score.

1. Without loss of generality, take $\beta = 1$. And let $a = e^{costdif(s_s,\vec{o},g)}$. Now formula (4.2) can be rewritten as:
$$P_{RG}(G \mid \vec{o}\,) = \alpha \cdot \frac{1}{1+a};$$
and we introduce an alternative *non-sigmoidal* distribution as:
$$P_X(G \mid \vec{o}\,) = \alpha \cdot \frac{1}{a}.$$

2. Considering the scores (i.e., the likelihood of each goal prior to normalisation), clearly, $\frac{1}{1+a} < \frac{1}{a}$. Furthermore, taking the ratio between the formulas P_{RG} and P_X, we see that $\lim_{a\to\infty} \frac{1}{1+a} \div \frac{1}{a} = 1$. That is, as a approaches infinity, P_{RG} converges towards—though it never reaches—P_X.

3. Now, $\frac{1}{a}$ is precisely (by definition) inversely proportional to a, whereas $\frac{1}{1+a}$ is not. Calculating the difference between them (by subtraction), we get:
$$\frac{1}{a} - \frac{1}{a+1} = \frac{(a+1)-a}{a(a+1)} = \frac{1}{a^2+a};$$
which, proportionally, is a decrease of:
$$\frac{1}{a^2+a} \div \frac{1}{a} = \frac{1}{a^2+a} \times \frac{a}{1} = \frac{1}{a+1}.$$
Therefore, the proportional decrease is greatest when a is lowest.

4. Recall that $a = e^{costdif(s_s,\vec{o},g)}$ and $P_{RG}(\hat{g} \mid \vec{o}\,) > P_{RG}(g \mid \vec{o}\,)$ was given. Therefore, $\frac{1}{e^{costdif(s_s,\vec{o},\hat{g})}+1} > \frac{1}{e^{costdif(s_s,\vec{o},g)}+1}$ for all $g \in G \setminus \{\hat{g}\}$. So, from 3, the proportional decrease $\frac{1}{a+1}$ is more for $\hat{g}$ than for other goals. Therefore, $P_{RG}(\hat{g} \mid \vec{o}\,) < P_X(\hat{g} \mid \vec{o}\,)$.

5. From 2, as a (which represents *costdif*) increases, $P_{RG}(\hat{g} \mid \vec{o}\,)$ converges (upward) towards $P_X(\hat{g} \mid \vec{o}\,)$. And since $\vec{o}'$ is uniformly less rational than $\vec{o}$ (as given), $P_{RG}(\hat{g} \mid \vec{o}'\,) > P_{RG}(\hat{g} \mid \vec{o}\,)$. □

The above anomaly occurs whenever cost difference increases by the same amount for all goals (i.e., whenever one observation sequence is uniformly less rational than another). The alternative observation sequence does not need to be wildly suboptimal; even the slightest suboptimality generates the same anomalous result.

Note that, although the above result at first appears to contradict R&G's principle—that lower cost difference should result in higher probability—actually, it does not. That principle applies to the total distribution *across goals* with respect to *one* GR problem, whereas Theorem 8 examines the situation across *two different problems* (because we have substituted for observations $\vec{o}$ the less rational $\vec{o}'$). So, we do not challenge the R&G principle. Nevertheless, Theorem 8 states that: when we change to a uniformly less rational observation sequence (a new GR problem), although the relative order across

Table 4.1: Probabilities for loopy paths.

	R&G ($\beta = 1$)			**R&G** ($\beta = 0.1$)			**V&K**		
	g_1	g_2	g_3	g_1	g_2	g_3	g_1	g_2	g_3
s_1	-	-	-	-	-	-	*0.3333*	0.3333	0.3333
s_2	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	***0.3610***	0.3280	0.3110
s_3	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	***0.3700***	0.3259	0.3042
v_1	***0.9693***	0.0304	0.0003	***0.4200***	0.3343	0.2458	**0.4517**	0.3194	0.2289
v_2	***0.9842***	0.0157	0.0001	***0.4656***	0.3238	0.2106	**0.4162**	0.3224	0.2615
v_3	***0.9842****	0.0157*	0.0001*	***0.4789***	0.3196	0.2014	**0.4060**	0.3223	0.2717
v_{10}	***0.9842****	0.0157*	0.0001*	***0.4820***	0.3186	0.1994	**0.3929**	0.3216	0.2855
	Non-sigmoidal distribution (does not change)								
v_{1-k}	***0.9842***	0.0157	0.0001	***0.4820***	0.3186	0.1994			

Probabilities on each of multiple visits to s and v (see Figure 4.2a). Winners are highlighted. Anomalies are italicised. (*Values have changed by tiny amounts, concealed by rounding.)

Table 4.2: Probabilities on a zigzagging path.

	R&G ($\beta = 1$)			**R&G** ($\beta = 0.1$)			**V&K**		
	g_1	g_2	g_3	g_1	g_2	g_3	g_1	g_2	g_3
v	**0.9694**	0.0303	0.0003	**0.4200**	0.3342	0.2458	**0.4517**	0.3194	0.2289
w	0.0008	0.0156	**0.9835**	0.2484	0.3264	**0.4351**	0.3176	0.3176	**0.3647**
x	0.3368	**0.6050**	0.0581	0.3436	**0.3609**	0.2955	***0.3708***	0.3380	0.2911
y	**0.9951**	0.0049	4.5e-05	**0.4943**	0.3072	0.1984	***0.4233***	0.3174	0.2593
z	0.0200	0.3734	**0.6066**	0.2694	0.3568	**0.3737**	***0.3562***	0.3318	0.3120

Probabilities as calculated at points v - z (see Figure 4.2b). Anomalies italicised, as above.

goals is maintained (the R&G principle applies) their *specific probability values* change in counter-intuitive ways.

The discrepancy exposed by Theorem 8 arises out of the use of a sigmoidal equation which is then normalised. As soon as we substitute a non-sigmoidal equivalent, the problem is resolved (as illustrated in the final row of Table 4.1 and proved formally at Theorem 13 below). Thus, replacing the R&G formula with its non-sigmoidal counterpart corrects an inconsistency.

Note. Our results when using the single-observation formula (Section 3.3, p.74) seemed to foreshadow the above. There, evaluating optimal cost via the observations on the basis of a single observation instead of a sequence, cost differences were reduced across the board (we subtracted from the first term, for all goals, the cost from the starting point to the final observation). In order to restore a semblance of parity with the R&G results, we found it necessary to add a large constant. If, instead of Equation (RG2) we had used a non-sigmoidal distribution such as formula (4.5) below, our results (whether using the baseline, simpler or single-observation cost difference) would match with no adjustment necessary (see proof to Theorem 13).

Turning now to V&K, whereas R&G maintains rankings but increases the probability

of the most probable goal, faced with irrational paths, this distribution has the effect that the goal that is *furthest from the start location* must eventually become more probable than any of the other goals (see highlighted anomalies in Tables 4.1 and 4.2).

Theorem 9. *Let $\vec{o}, \vec{o}' \in O^*$ be observation sequences such that $\vec{o}'$ is uniformly less rational than $\vec{o}$ and let $g_1, g_2 \in G$ be two goals such that $optc(n_s, g_1) > optc(n_s, g_2)$, that is, the optimal cost of achieving g_1 is greater than the optimal cost of achieving g_2. Then, there exists a c such that when $optc(n_s, \vec{o}', g_1) - optc(n_s, \vec{o}, g_1) \geq c$, $P_{VK}(g_1 \mid \vec{o}') > P_{VK}(g_2 \mid \vec{o}')$.*

Proof. The numerators used for $P_{VK}(\cdot)$ formula (4.3) to score each goal are different but constant, based on the optimal cost to each of those goals. Under uniformly less rational observations, by Definition 10, the cost of the denominator increases equally for all goals. Thus, all scores decrease, but the score with the largest numerator decreases most slowly. Since $optc(n_s, g_1) > optc(n_s, g_2)$, the score for g_1 has the largest numerator, and the proposition follows. □

One impact of Theorem 9 is that the cost difference principle is not maintained: lower cost difference does *not* imply higher probability. Furthermore, the fact that, as the size of the denominator increases, it is the most distant (i.e., most *expensive*) goal that begins to be favoured, is again anomalous: the underlying intuition for cost-based GR is that "cheaper is better", yet here the goal's score increases precisely because its attainment costs more.

Observe also that V&K always returns a comparatively flat distribution: even on the first visit to v on an optimal path to g_1, it yields $P(g_1) < 0.5$. In a practical application, where the user might be waiting for the probability of a goal to exceed some threshold before triggering an event, that trigger might never be reached.

Summary of Findings

The above analyses identify problematic cases in which both GR models yield undesirable outcomes. The R&G model is relatively consistent and easy to understand but, faced with apparent irrationality, oscillates between goals depending on the most recent observation. Used with higher β values, it is also beguilingly decisive, able to return, in the zigzagging example (Table 4.2), $P(g_3) = 0.98$ then just a few steps later $P(g_1) = 0.99$! Additionally, the more irrational the agent, the more confidently the distribution points towards the most probable goal (Theorem 8). This is apparent in the looping example (Table 4.1 at v), most obviously for $\beta = 0.1$, where probabilities start from a lower (flatter) base.

V&K, on the other hand, appears inconsistent and indecisive. Even when the agent seems clearly on the optimal path towards a particular goal (Table 4.2 at v), it assigns that goal a probability below 0.5, scarcely outscoring its much more suboptimal competition.

Moreover, faced with an irrational agent who seems not to be targeting any particular goal, it appears biased to prefer the most distant or expensive goal, no matter where the agent is currently located. At first, it oscillates (note that at w, V&K has 'swung' from g_1 to prefer g_3, in agreement with R&G). Once the path becomes excessively suboptimal, however, the ratio on which V&K depends becomes so diluted that the most distant or costly-to-reach goal from the starting point (which supplies the largest numerator) again dominates (Table 4.2, locations x, y and z).

In fairness, both the above accounts were developed under the assumption that the observed agent is rational and honest. It is a 'soft' assumption, however, in that both aim for a GR framework capable of accommodating suboptimal behaviour. Indeed, both models derive their probability distributions based on the degree of suboptimality that they encounter. Furthermore, the fact is that a GR system cannot 'know' what type of recognition it is dealing with and in some settings (e.g., security) there may be adverse consequences if the assumption is incorrect. It appears, therefore, that rationality ought to be accommodated in the framework natively, as our formula aims to do.

4.1.3 Measuring the Degree of Irrationality

We have seen that probabilities generated by V&K's formula (4.3), when confronted by an excessively suboptimal plan, can seem illogical in the way that it biases towards the most distant goal. Nonetheless, the *score* on which the probabilities are based degrades (behind the scenes) in an interesting and useful way.

The ratio $optc(s_s, g) \div optc(s_s, \vec{o}, g)$ used under the V&K model balances optimal cost from start to goal against optimal cost through the observations. Thus, a perfectly rational observed plan, where $optc(s_s, \vec{o}, g) = optc(s_s, g)$, yields a score of 1; but as the observed behaviour becomes increasingly erratic (that is, suboptimal for *all* goals), the size of the denominator *increases* (for all goals) while the size of the numerator (for all goals) remains the same.

Effectively then, the more irrational the observations, the lower *all* the scores become; so the lower the *maximum* score becomes.

> **Note.** If a plan is optimal (or close to optimal) for *some* goal, it is not an erratic or irrational one, and the maximum score approaches 1. Only when the plan is *suboptimal for all goals*—i.e., 'less rational' under Definition 9—is the maximum score diminished.

It turns out, then, that the maximum score at any point in the plan provides a good measure of the degree to which optimality has (in general) become 'diluted'.

Definition 11. *Relative to a GR problem, $\mathcal{P}$, the* ***rationality measure (RM)*** *is given by:*

$$RM(n_s, G, \vec{o}) = \max_{g \in G} \frac{optc(n_s, g)}{optc(n_s, \vec{o}, g)}. \tag{4.4}$$

Notice that the RM represents a more general, cumulative measure than that suggested by the notion of being 'uniformly less rational'. Like Definition 9, although it measures suboptimality across all goals, observations may not be suboptimal for each goal *by the same amount*. Furthermore, by taking the maximum score, we always assess rationality based on the 'best' possible interpretation (i.e., with respect to the goal for which observed behaviour is closest to optimal behaviour). So, given two observation sequences $\vec{o}$ and $\vec{o}'$, where $\vec{o}'$ is less rational than $\vec{o}$, $RM(n_s, G, \vec{o}') < RM(n_s, G, \vec{o})$. That is, the RM for the *less* rational observations is always lower. Observations with a lower RM are *not* always *uniformly* less rational, however.

Note. 'Uniformly less rational' observations involve the *same amount of unnecessary work* for all goals; observations with a low RM also do a lot of unnecessary work but may still be tracking (albeit suboptimally) towards one particular goal: they are 'less rational' but may or may not be 'uniformly' less rational.

Our model for self-modulating GR (below) uses the RM 'on-the-fly' to provide a snapshot of the agent's degree of irrationality, based on her immediate history but the measure has other potential uses. For example, it could also be used from problem to problem, as follows. Once the RM for a particular agent has been established (on the basis of the current, or past, problem), it provides a means of predicting how suboptimal that agent's behaviour is likely to be in future. Though beyond the scope of this thesis, this may have an impact on the sort of (suboptimal) planner that might be used to generate plans with which to compare observations, should the same agent be encountered a second time. If the RM is very low (i.e., highly suggestive of irrationality) then this might be used to flag the need for additional (perhaps human) surveillance on future sightings.

We next present our *self-modulating* account which uses the RM, in combination with a non-sigmoidal variation of R&G, to lift the rationality assumption.

4.1.4 A Self-Modulating Formula

Our objective is to generate a probability distribution—a solution to a GR problem $\mathcal{R}_g = \langle \mathcal{D}, s_s, G, \vec{o}, Prob \rangle$—that preserves the intuition behind R&G that the lower the cost difference, the higher the probability but which modulates its level of confidence relative to the degree of rationality observed so far.

To achieve this, we propose the following self-modulating formula:

$$P(G \mid \vec{o}) = \alpha \cdot \frac{1}{e^{\beta \, costdif(s_s, \vec{o}, g)}}, \tag{4.5}$$

where $\beta = RM(s_s, G, \vec{o})^{\gamma}$, and γ is a positive constant.

Formula (4.5) maintains an R&G-like awareness of the goal the agent seems to be approaching but, if the agent becomes irrational with respect to all goals (i.e., apparently cost-insensitive and possibly deceptive), the formula self-regulates and lowers its level of confidence accordingly. Thus, the mechanism it uses explicitly accommodates the rationality assumption in the process of performing GR. More concretely, while the agent behaves rationally (with respect to at least one of the goals), a confident prediction in returned, at the limit of those we have seen from R&G ($\beta = 1$); but the more irrational the agent becomes (i.e., her observed behaviour is becoming excessively suboptimal with respect to all goals in the domain), a less confident prediction is given, resembling the more subdued distributions of V&K or R&G ($\beta = 0.1$) (i.e., with lower β value).

Importantly, our formula (4.5) substitutes for R&G's Boltzmann equation a non-sigmoidal distribution (which does not suffer from the discrepency captured by Theorem 8). It does, however, draw on a seldom-discussed feature of the R&G model: the β parameter.

The Rate Parameter in R&G

The solution to a GR problem under R&G is achieved using the Boltzmann probability distribution—formula (4.2)—tempered by a *rate parameter* β. As seen in Tables 4.1 and 4.2, while the value of β makes no difference to the relative *ranking* of goals within a probability distribution, it does have considerable impact on the *shape* of that distribution. Indeed, as briefly discussed in Ramirez's PhD thesis (2012, p.63) (though mostly ignored in the papers):

> *This [parameter] allows plan recognition system developers to soften the implicit assumption of the agent being rational as in preferring those plans that minimize their total cost. The smaller the value of β the more will the distribution resemble a uniform distribution ...*

Thus, formula (4.2) already includes a parameter to control the level of confidence in the observed agent's rationality; but the choice of a value for β (given a value of 1.0 in Ramirez's thesis and in code linked from Ramirez and Geffner's 2010 paper) is left to be set by the GR system developer, presumably on the basis of domain knowledge or special information about the particular agent under observation. Our approach, in Equation (4.5), is for the formula to self-adjust this parameter 'on-the-fly' based on the RM, which, recall, we derive by maximising the score (not the probabilities) from V&K.

Properties of the Self-Modulating Formula

Formula (4.5) (via the now dynamic β) synthesises the two accounts discussed above with additional features to achieve the following.

1. In place of the Boltzmann, this exponential distribution precisely enforces the intuition that the lower the cost difference, the higher the probability. Furthermore, unlike formula (4.2), it is closed under scaling and so returns consistent probability values when one sequence of observations is uniformly less rational than another; also, it guarantees probabilities (before modification by the β parameter) always at the limit of those calculated under formula (4.2) (see proof of Theorem 8).

2. The β parameter self-adjusts by reference to the current degree of suboptimality, derived from the maximum score under V&K's probability distribution formula (4.3).

3. We have introduced a confidence parameter γ to regulate how quickly confidence should drop if irrational behaviour is detected. If γ is high, confidence drops rapidly (i.e., the less suboptimal the observations need to be before the probability distribution flattens out).

4. Our formula (4.5) ostensibly requires three calls to a planner per goal—to calculate $optc(s_s, \vec{o}, g)$, $optc^{\neg}(s_s, \vec{o}, g)$ and $optc(s_s, g)$—whereas the R&G distribution requires only the first two. However, the additional call (required for the RM formula 4.4) depends on the domain, not on the observations, so can be precalculated and cached. Taking this approach, self-modulation can be achieved without time penalty (though our main focus is on capturing the intended meaning more accurately, rather than on achieving computational efficiency).

In the following results, $P(\cdot)$ represents our self-modulating formula; $\beta_{\vec{o}}$ represents the β value for observations $\vec{o}$ from Equation (4.5), that is, $\beta = RM(\text{—}\vec{o})^{\gamma}$; and $P_{RG}^{\beta=x}$ represents Equation (4.2) with $\beta = x$.

First, we formalise the observation made at 1 above.

Observation 2. $\lim_{costdif(s_s,\vec{o},g)\to\infty} P_{RG}^{\beta=\beta_{\vec{o}}}(\cdot) \div P(\cdot) = 1$.

This simply follows from $\lim_{a\to\infty} 1/(1+a) \div 1/a = 1$ in the proof to Theorem 8.

Next, the more rational an agent's behaviour (i.e., the observation sequence $\vec{o}$), the higher the β in our account and, thus, the more closely probabilities approach those at the limit of R&G ($\beta = 1$).

Theorem 10. *Let $\vec{o}, \vec{o}' \in O^*$ be two observation sequences such that $\vec{o}'$ is less rational than $\vec{o}$. Then, it is the case that $1 \geq \beta_{\vec{o}} > \beta_{\vec{o}'}$.*

Proof. β is based on the ratio at formula (4.3). Across goals, numerators remain constant. As observed costs increase, all denominators increase and all values decrease: therefore the maximum must decrease. □

Now, by Observation 2, the non-sigmoidal distribution returns probabilities always at the limit of R&G for any particular β value (see Table 4.1, p.107). Moreover, when

observations conform to optimal behaviour, $\beta = 1$ in Equation (4.5). Therefore, when an agent behaves fully rationally—that is, behaves optimally with respect to any one of the goals—our probability distribution is always at the limit of R&G ($\beta = 1$).

Next, see that even when our account diverges from R&G ($\beta = 1$), it still maintains the same relative rankings across goals.

Theorem 11. *For all observations $\vec{o} \in O^*$ and goals $g_1, g_2 \in G$, $P(g_1 \mid \vec{o}\,) > P(g_2 \mid \vec{o}\,)$ iff $P_{RG}(g_1 \mid \vec{o}\,) > P_{RG}(g_2 \mid \vec{o}\,)$.*

Proof. The differences between formulas (4.2) and (4.5) have no impact on probability rankings. Specifically:

1. subtracting $+1$ from the denominator of *every* score does not change their relative order; and

2. β is a multiplicative constant, so changing its value (including by the introduction of γ) effects a monotonic transformation, which (again) does not change the relative order of probabilities. □

Thus, our formula is aligned with the underlying assumption of the R&G framework with respect to the rationality of the observed agent and never alters the qualitative outcome: goal rankings are maintained. Critically, though, as the following important result states, the more erratic the observations (i.e., the more suboptimal for *all* goals), the more even the probability distribution becomes.

> The more erratic the observations, the more the probabilities under 'Self-Mod' even out.

Theorem 12. *Let $\vec{o}, \vec{o}\,' \in O^*$ be two observation sequences such that $\vec{o}\,'$ is less rational than $\vec{o} \in O^*$. Then, for every two goals $g_1, g_2 \in G$ such that $P(g_1 \mid \vec{o}\,') \neq P(g_2 \mid \vec{o}\,')$ (i.e., whenever the two goals are distinguishable):*

$$|P(g_1 \mid \vec{o}\,') - P(g_2 \mid \vec{o}\,')| < |P(g_1 \mid \vec{o}\,) - P(g_2 \mid \vec{o}\,)|.$$

Proof. 1. Since $\vec{o}\,'$ is less rational than $\vec{o}$, by Definition 9, for all $g_1 \in G$, $optc(s_s, \vec{o}\,', g_1) > optc(s_s, \vec{o}, g_1)$.

2. From Theorem 10, β in Formula (4.5) is reduced when the cost of the observation sequence increases.

3. When β is reduced, a monotonic transformation diminishes the difference between probabilities. □

Finally, we confirm that, using our self-modulating formula, the anomalous behaviour of R&G identified in Theorem 8 can never occur.

Theorem 13. *Let $\vec{o}, \vec{o}' \in O^*$ be two observation sequences such that $\vec{o}'$ is uniformly less rational than $\vec{o}$ and let $\hat{g} \in G$ be such that $P(\hat{g} \mid \vec{o}\,) > P(g \mid \vec{o}\,)$, for all $g \in G \backslash \{\hat{g}\}$ (i.e., goal $\hat{g}$ is the best explanation under the more rational observations $\vec{o}$). Then, $P(\hat{g} \mid \vec{o}'\,) < P(\hat{g} \mid \vec{o}\,)$. That is, the probability of goal $\hat{g}$ is lower under the less rational observations.*

Proof. Since $\vec{o}'$ is uniformly less rational than $\vec{o}$, there is a c such that for all $g \in G$, $optc(s_s, \vec{o}', g) - optc(s_s, \vec{o}, g) = c$. The addition c to the exponent is a multiplicative constant, which cancels out on normalisation (recall that $\beta_{\vec{o}'}$ represents the β value from Equation (4.5) when the observation sequence is $\vec{o}'$):

$$
\begin{aligned}
P(G \mid \vec{o}'\,) &= \alpha \cdot \frac{1}{e^{\beta_{\vec{o}'}\,(optc(s_s,\vec{o},g) - optc(s_s,g) + c)}} \\
&= \alpha \cdot \frac{1}{e^{\beta_{\vec{o}'}\,(optc(s_s,\vec{o},g) - optc(s_s,g))}} \cdot \frac{1}{e^{\beta_{\vec{o}'}\,(c)}} \\
&= \frac{\frac{1}{e^{\beta_{\vec{o}'}\,(optc(s_s,\vec{o},g) - optc(s_s,g))}} \cdot \frac{1}{e^{\beta_{\vec{o}'}\,(c)}}}{\sum_{g_i \in G} \frac{1}{e^{\beta_{\vec{o}'}\,(optc(s_s,\vec{o},g_i) - optc(s_s,g_i))}} \cdot \frac{1}{e^{\beta_{\vec{o}'}\,(c)}}} \\
&= \frac{\frac{1}{e^{\beta_{\vec{o}'}\,(optc(s_s,\vec{o},g) - optc(s_s,g))}} \cdot \frac{1}{e^{\beta_{\vec{o}'}\,(c)}}}{\frac{1}{e^{\beta_{\vec{o}'}\,(c)}} \cdot \sum_{g_i \in G} \frac{1}{e^{\beta_{\vec{o}'}\,(optc(s_s,\vec{o},g_i) - optc(s_s,g_i))}}} \\
&= \frac{\frac{1}{e^{\beta_{\vec{o}'}\,(optc(s_s,\vec{o},g) - optc(s_s,g))}}}{\sum_{g_i \in G} \frac{1}{e^{\beta_{\vec{o}'}\,(optc(s_s,\vec{o},g_i) - optc(s_s,g_i))}}} \\
&= \alpha \cdot \frac{1}{e^{\beta_{\vec{o}'}\,(optc(s_s,\vec{o},g) - optc(s_s,g))}}.
\end{aligned}
$$

Now, $P(G \mid \vec{o}\,) = \alpha \cdot \frac{1}{e^{\beta_{\vec{o}}\,(optc(s_s,\vec{o},g) - optc(s_s,g))}}$. That is, under the less rational observations $\vec{o}'$, there is no change in the underlying values; only the β value changes (from $\beta_{\vec{o}}$ to $\beta_{\vec{o}'}$).

From Theorem 10, $1 \geq \beta_{\vec{o}} > \beta_{\vec{o}'}$; and from Theorem 12, when $\beta_{\vec{o}} > \beta_{\vec{o}'}$, for every two goals $g_1, g_2 \in G$ such that $P(g_1 \mid \vec{o}'\,) \neq P(g_2 \mid \vec{o}'\,)$, it is the case that $|P(g_1 \mid \vec{o}'\,) - P(g_2 \mid \vec{o}'\,)| < |P(g_1 \mid \vec{o}\,) - P(g_2 \mid \vec{o}\,)|$. That is, when the cost of observations increases, the difference between probabilities for all distinguishable goals is reduced.

Therefore, since (as given) $P(\hat{g} \mid \vec{o}\,) > P(g \mid \vec{o}\,)$ for all $g \in G \setminus \{\hat{g}\}$ and since probabilities sum to 1 (i.e., when the lower increases, the higher must decrease), $P(\hat{g} \mid \vec{o}'\,) < P(\hat{g} \mid \vec{o}\,)$. □

Comparison of Results

In Tables 4.3 and 4.4, we compare the probabilities returned by our self-modulating formula (4.5) with the original 'static' R&G formula (4.2). Referring to Table 4.4, observe that, at location v, when the agent appears to be on an optimal path to g_1 (though on a suboptimal path to g_2 and g_3), 'Self-Mod' (4.5) maintains $\beta = 1$ and therefore yields a confident prediction. But, as the path becomes increasingly suboptimal, the distribution obtained by formula (4.5) evens out so that, by location z, the most and least likely goals are separated by just 0.07 (compared with 0.58 using static R&G).

Table 4.3: Probabilities revisited: loopy paths.

	R&G ($\beta = 0.1$)			**Self-Mod**			$\boldsymbol{\gamma = 2}$
	g_1	g_2	g_3	g_1	g_2	g_3	β
s_2	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.2642
s_3	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333	0.1196
v_1	***0.4200***	0.3342	0.2458	**0.9842**	0.0156	0.0001	1
v_2	***0.4656***	0.3237	0.2106	**0.5752**	0.2906	0.1342	0.1649
v_3	***0.4789***	0.3196	0.2015	**0.4295**	0.3283	0.2422	0.0649
v_{10}	***0.4820***	0.3186	0.1994	**0.3406**	0.3336	0.3259	0.0050

Table 4.4: Probabilities revisited: zigzagging path.

	R&G ($\beta = 1$)			**Self-Mod**			$\boldsymbol{\gamma = 2}$
	g_1	g_2	g_3	g_1	g_2	g_3	β
v	**0.9694**	0.0303	0.0003	**0.9842**	0.0156	0.0001	1
w	0.0008	0.0156	**0.9835**	0.1267	0.2458	**0.6275**	0.2262
x	0.3368	**0.6050**	0.0581	0.3514	**0.3886**	0.2500	0.1716
y	**0.9951**	0.0049	4.5e-05	**0.6018**	0.2674	0.1308	0.1526
z	0.0200	0.3734	**0.6066**	0.2894	0.3513	**0.3637**	0.0715

Notice that 'Self-Mod' always maintains the same rankings as R&G but, referring now to Table 4.3, we clearly see that whereas a looping path causes 'Self-Mod' to return a less confident prediction (the more the path loops, the more the distribution flattens out), the opposite is true of R&G, which increases in confidence with every loop. Thus, using formula (4.5), the validity of the assumption with respect to the rationality of the observed agent has been accounted for.

Putting it all together, our self-modulating formula provides the performance we set out to achieve: as the agent becomes more erratic (suboptimal), it yields a distribution closer to uniform. Practically, faced with apparently irrational—and possibly deceptive—behaviour, the GR system judges goals 'more equally', displaying a reduced level of confidence.

In this section, we presented a self-modulating model of GR. Our model 'lifts' what is perhaps the strongest assumption in contemporary state-of-the-art cost-based GR approaches: the apparent rationality of the observed agent. As a result, we can handle agents ranging from the strictly rational to the arbitrarily irrational in a principled manner.

Notwithstanding the advantages of our approach, from the deceiver's point of view, deception by dazzling is still somewhat successful: it leaves the observer unable to reliably arrive at a correct conclusion. At worst (for the deceiver), a GR system using our model 'knows that it does not know' and, potentially, uses the levelling out of probabilities to trigger human intervention or some other method of anomaly detection; at best (for the deceiver) a GR system using other methods jumps to a false conclusion long before the

true goal has been revealed.

Whatever the outcome, deception by dazzling is an expensive strategy: the best plan (from the deceiver's point of view) seems to be whichever plan costs the most. In the next section, we look for a more economical solution.

4.2 A Model for Deceptive Path-Planning

He will conquer who has learnt the artifice of deviation. Sun Tzu

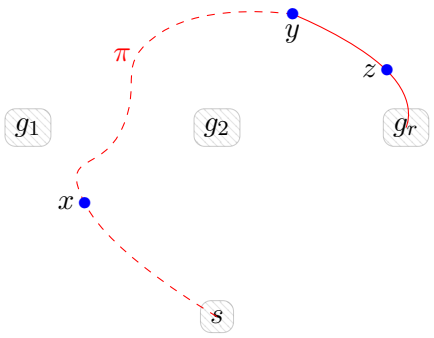
We have seen, from Section 4.1 that, given a large enough budget, it should be possible to confound (if not deceive) an observer by adopting a wildly irrational plan. Our objective, however, goes further: we want to maximise deceptivity but at the lowest possible cost. In what follows, we reduce this problem back to path-planning in order to establish a straightforward way in which it can be done.

Our definition of deception, recall, is from Bowyer (1982, p.47). It is the "distortion of perceived reality" which may be achieved in one of two ways: by *simulation* (showing the false) or *dissimulation* (hiding the real).[8] In path-planning, where the only reality is movement towards a goal, this equates either to obscuring the path-planner's true destination or to creating the impression that she is going somewhere that she is not. Considered in terms of the GR systems that we examined in Part I and in Section 4.1 of Part II, the problem can be framed as the task of generating a sequence of locations such that, when they are submitted to a GR system as *observations*, that system (whether computational or human) either: (i) is unable to determine the agent's true destination (because there are multiple possible goals all with the same probability) or (ii) wrongly identifies the agent's destination (because the probability of some bogus goal exceeds the probability of the real one).

Thus, our method is essentially an 'inversion' of probabilistic GR. We have seen that probabilistic GR systems take as input a sequence of observations and output the probability of each goal. Observation sequences may be broken or continuous but, in GR, they are always partial (i.e., incomplete) insofar as the purpose of GR is to determine the goal *before* it has been reached. Equipped with such a system, therefore, having generated an unbroken path all the way from the starting point to the real goal (e.g., as shown in Figure 4.3), we can present for GR a sequence of all observations from the starting point up to any (or every) step in the path and obtain the relative probabilities of every goal *at that step*. If the probability of the real goal is less than or equal to the probability of some other goal (as at x and y in the figure), then we say that *that step* is 'deceptive'.

In this section, we present a DPP model, which uses GR as a black box. That is, the model is agnostic with respect to the particular GR framework employed, requiring only

[8] The theory was introduced under a pseudonym (Bowyer, 1982), then rearticulated independently by Bell (2003) and (Whaley, 1982) as discussed in Section 2.4.2, p.28.



Point in π	Probabilities g_1	g_2	g_r	D
x	0.75	0.20	0.05	✓
y	0.10	0.45	0.45	✓
z	0.05	0.10	0.85	✗

Figure 4.3: Deception as an inversion of probabilistic GR: if we can rank for likelihood, we can rank for *un*likelihood. Column D shows deceptivity when g_r is the real goal.

that it be capable of returning a probability distribution across goals at any point, given a sequence of observations. Building on this routine capability of any probabilistic GR system, our model quantifies a path's deceptivity on the following three dimensions.

- **Magnitude**. At step level, we measure the relative probability values across goals to determine whether the step is truthful or deceptive. In Section 4.4.2, we discuss how an extended interpretation of magnitude—which also evaluates the degree of simulation versus the degree of dissimulation—can be incorporated into the model.
- **Density**. At path level, we measure *how much* of the path is deceptive. In fact, we take a 'worst case' approach to density, inverting our quantification of how *truthful* (and therefore liable to detection) it has become.
- **Extent**. In a fully observable domain, every path eventually becomes truthful (because it is observed reaching its goal). At path level, extent measures the distance travelled along a path until it reaches its *last deceptive point*. In evaluating extent, we introduce the concept of 'path completion': a means of comparing distance travelled along different suboptimal paths between the same two points.

As Bowyer (1982) points out, in combat, it is deception, not honesty, that saves lives. Nevertheless, deception has had such a 'bad press' that, when advocating its use, the literature tends to characterise the practice instead as protection of privacy (e.g., Keren et al., 2016). The distinction is moot. Consider, for example, a convoy escorting a VIP to one of three possible destinations. An observer plans to deploy an assassin once the VIP's destination is known. What (deceptive) path will (protect the VIP's privacy so as to) minimise the likelihood of the observer correctly identifying the convoy's destination? If Figure 4.3 illustrates the convoy's path π, then an assassin deployed after observation point x will probably be deceived into setting up at g_1. At point y, she will be unable to decide whether to go to g_2 or the real goal g_r. Thus, whether characterised as privacy or deception, the ability to conceal the convoy's real goal could save the VIP's life.

We have described the DPP problem informally as one of finding a path, such that an observer watching an agent make her way along that path is unable to determine, until the last possible moment, where the agent is going. We next set out the formal definition.

4.2.1 Technical Framework: DPP

Without loss of generality, we express the framework here in a discrete setting of nodes and edges.[9] However, we abbreviate the notation, referring henceforth to the starting location as s (not n_s) and to a goal location as $g \in G$ (not $n_g \in G$). Thus, from Definition 2 (p.45), a discrete path-planning problem $\mathcal{P}_d = \langle \mathcal{D}_d, s, g \rangle$ (for convenience, redefined within Definition 12 below) is the problem of finding a path from a starting location s to a goal g in a discrete path-planning domain $\mathcal{D}_d$ (originally defined p.44). Meanwhile, from Definition 3 (p.46), a discrete GR problem for path-planning $\mathcal{R}_d = \langle \mathcal{D}_d, G, s, \vec{o}, Prob \rangle$ attempts to identify from a set of goals G and a sequence of observations (or partial path) $\vec{o}$, which particular goal that partial path is most likely to be targeting.

DPP is a path-planning problem that inverts goal recognition. Like path-planning, it is the problem of finding a path from the starting location s to the only real goal g_r (in a set of possible goals); but, whereas a GR system determines the most likely goal based on a sequence of observations, DPP assembles a continuous sequence of observations (i.e., a path) from which an observer (or GR system) is *least likely* to identify the real goal.

Definition 12. *A **deceptive path-planning problem** is a tuple $\langle \mathcal{P}_d, G, P \rangle$, where:*

- *$\mathcal{P}_d = \langle \mathcal{D}_d, s, g_r \rangle$ is a discrete path-planning problem such that:*
  - *$\mathcal{D}_d = \langle N, E, c \rangle$ is a discrete path-planning domain, where:*
    - *N is a non-empty countable set of nodes,*
    - *$E \subseteq N \times N$ is a set of edges between nodes, and*
    - *$c : E \mapsto \mathbb{R}_0^+$ returns the non-negative cost of traversing each edge;*
  - *$s \in N$ is the start location;*
  - *$g_r \in G$ is the real goal;*
- *$G \subseteq N$ is the set of all candidate goal locations, including bogus goals and the real goal; and*
- *$P(G \mid \vec{o} \cdot n)$ is the posterior probability of a goal, given a sequence of observations ending at n, where:*
  - *$\vec{o} = o_1, \ldots, o_k$, such that $o_i \in N$ for each $1 \leq i \leq k$, $k \geq 0$,*

[9] Our framework depends on the ability to obtain a probability distribution at any point along a path where an agent could be observed. This may be achieved in a discrete domain (as described in Section 3.1.1, p.44) or in a continuous domain (as described in Section 3.2.1, p.62).

– $n \in N$, *and*

– *the observation sequence is feasible, that is,* $optc(s, \vec{o}, n) \neq \infty$.

That is, P stands for the model of the observer, a black box against which the most recent step n or the 'path so far' $\vec{o} \cdot n$ can be tested.[10]

The solution to a DPP problem is a solution to its path-planning problem (i.e., a path from s to g_r, as defined p.44) that is *deceptive*. The quality of that solution depends on the magnitude, density and extent of the deception, as we now discuss.

4.2.2 Measuring Deception

We first examine deception as it applies to an individual node or step along the path.[11] GR, captured in our model as the probability function $P(\cdot)$, gives us a means of determining, at any particular point, whether the path (at that point) is truthful or deceptive, based on the ***magnitude*** of the probabilities. In Section 4.4.2, we discuss how a step's magnitude can be further evaluated in terms of simulation and dissimulation. For now, we use it as a binary measure.

Definition 13. *A **truthful** step is the final node n in an observation sequence $\vec{o} \cdot n$ such that the probability of the real goal g_r exceeds the probability of any other candidate goal, that is, $P(g_r \mid \vec{o} \cdot n) > P(g \mid \vec{o} \cdot n)$, for all $g \in G \setminus \{g_r\}$. Otherwise, the step is **deceptive**.*

Clearly, we want to minimise the opportunities for an observer to correctly identify the real goal. Since we have no way of knowing when or how often the observer will be making observations, that means minimising the number of truthful steps. The fewer such steps a path π contains, the greater its deceptive ***density***:

$$density(\pi) = \frac{1}{|N_t|}, \tag{4.6}$$

where N_t is the set of all truthful steps in π.

Now, in life, a deception might not be uncovered for months or years after it occurs (if ever); but a deceptive path, with full observability, is always ultimately truthful because the final step always arrives—and is seen to terminate—at its goal.[12] So, a path that is deceptive all the way to the goal has the minimum number of truthful steps ($|N_t| = 1$) and, therefore, a maximum density of 1. Furthermore, referring to Figure 4.4, observe that, since every path is deceptive at its start and truthful at its goal, there must always

[10] A GR system using single-observation recognition as set out in Part I, requires only a single current location n, not the complete observation sequence $\vec{o} \cdot n$.

[11] To apply the concept of 'steps' in a continuous domain, a path may be discretised into segments of equal length between notional waypoints, as suggested by Kaminka et al. (2018), which are then treated as nodes.

[12] Implicitly, we assume an observation at the *final time-step+1*, at which the agent is seen to remain—i.e., terminate—at her goal.

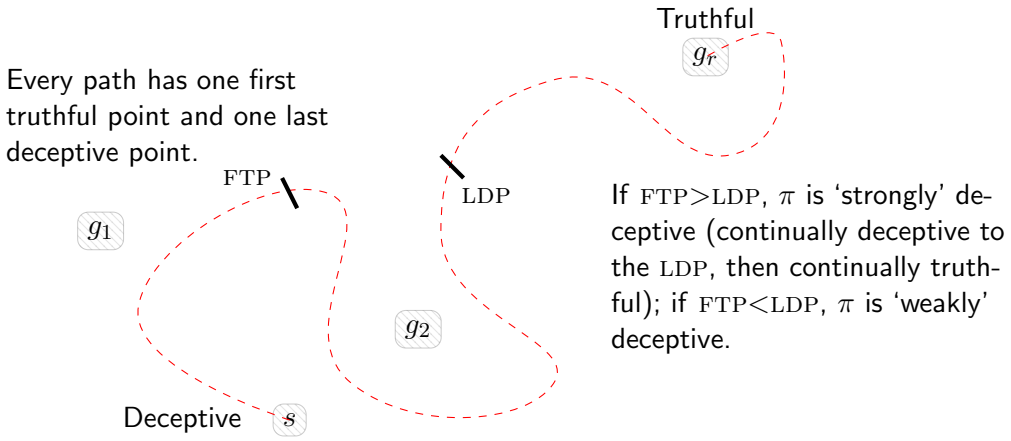


Figure 4.4: A deceptive path.

Every path is deceptive before setting off and truthful once it has reached its destination. Therefore every path has a first truthful point and a last deceptive point. When the observed path from s is assessed at point FTP above, g_r is the most probable goal. When the path detours around g_2, however, g_2 becomes more probable. Finally, assessed at any point in the path *after* LDP, g_r is again the most probable goal.

be *one unique truthful node* (even if it is the goal) prior to which all previous steps (if any) are deceptive; and *one unique deceptive node* (even if it is the starting point) beyond which all subsequent steps are truthful. We call these the 'first truthful point' (FTP) and 'last deceptive point' (LDP) respectively.

Definition 14. *Given a path π, its **first truthful point** FTP_π is a node π^i which is itself truthful whereas all (if any) previous nodes π^j, for all $j \in \{0, \ldots, i-1\}$, are deceptive.*[13]

Definition 15. *Given a path π, its **last deceptive point** LDP_π is a node π^i, which is itself deceptive whereas all (if any) subsequent nodes π^j, for all $j \in \{i+1, \ldots, |\pi|\}$, are truthful.*

Intuitively, the location of the LDP is a measure of the path's deceptive *extent* (it is a deceptive point beyond which all subsequent points are truthful) but, by considering also the location of the FTP, we can make a more nuanced assessment as to the degree of deceptivity that the path achieves overall.

By their nature, in any given path, there can be only one LDP and only one FTP. However, as illustrated at Figure 4.4, after the first truthful point has occurred there may be many subsequent steps that are deceptive before the last deceptive point is reached. Thus, depending on the relative location of these two points, we can identify two extreme notions of a deceptive path. If all truthful steps in a path occur after the LDP, we say that the path is 'strongly deceptive'; whereas a path that includes truthful segments *interspersed* with deceptive segments, we say, is 'weakly deceptive'.

[13] For a reminder of our notation conventions with respect to paths, see Definition 1, p.44.

Definition 16. *A **strongly deceptive** path π is continually deceptive to its LDP, that is, if* $\text{LDP}_\pi = \pi^i$*, then* $\text{FTP}_\pi = \pi^{i+1}$.

Definition 17. *A **weakly deceptive** path π includes truthful steps before its LDP, that is, if* $\text{LDP}_\pi = \pi^i$*, then* $\text{FTP}_\pi = \pi^j$*, for some* $j < i$.

Now, in order to maximise the deceptive *density* of a path, we need it to be strongly deceptive (i.e., to have no truthful steps before the LDP) but we also need to minimise the number of truthful steps that occur afterwards. That is, we want the LDP—which indicates the path's deceptive extent—to occur 'as late in the path as possible'.

Although we are able to identify the location of the LDP as a particular node in the path (or, in a continuous domain, by its position proportional to the length of the path as a whole), that information by itself does not properly match our intended meaning. If a deceptive path includes two or three meaningless additional 'loops', for example, should we regard its deceptive extent as having increased? This is the problem that we consider next.

Note. Before proceeding, we briefly mention the potentially confusing relationship between path optimality and deceptivity. Note that a path can be truthful without being optimal (it may favour the real goal more than any bogus goal but still be a suboptimal path) and deceptive without being suboptimal (it may be an optimal path to multiple goals).

4.2.3 Maximising Deceptive Extent

In order to compare the relative deceptivity of two competing paths, we want to know which of them has the greatest deceptive density and which has been deceptive 'for longer'; but the latter concept is not straightforward. Given that deceptive paths (in fact *all* paths!) are typically suboptimal to some degree, we need to consider how, practically, one should go about assessing the distance travelled along one suboptimal path by comparison with the distance travelled along another.

Path Completion

The LDP is significant because it represents the point in a path at which a rational observer ceases to be deceived; that is, the moment when, in the eyes of the observer, the probability of the real goal comes to outweigh the probability of any other possible goal and beyond which the probability of that real goal dominates continuously until the goal is reached. Intuitively, we want to delay this point until 'as late in the path as possible'; but although the LDP is easy to identify (i.e., we can say with certainty that it occurs at the i^{th} or j^{th} step), expressing its location in terms that allow for comparison between paths of different lengths and shapes is problematic.

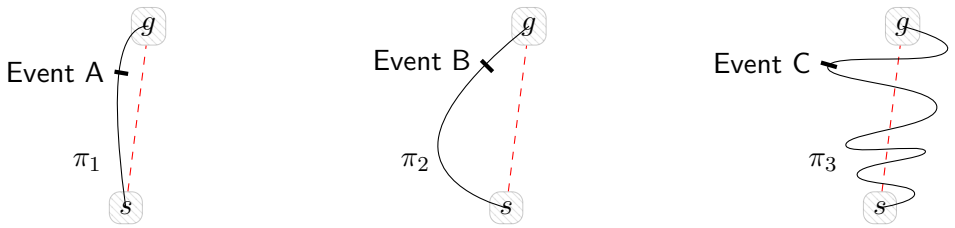


Figure 4.5: Progress along a suboptimal path: which event occurs 'later'?

This problem is not unique to the case of an LDP; it arises whenever we want to compare progress along two different suboptimal paths. Consider the three paths in Figure 4.5. The black markers represent different events but which of them, relative to their respective paths, should we say occurs later? Considered purely in terms of distance travelled, Event B occurs later than Event A, even though they both end up at approximately the same distance from goal. By the same reckoning, Event C occurs much later than either of the others; after it has occurred, however, the traveller on π_3 has further left to go than a traveller on either of the other paths. Does it make sense to say that we are later in the path, if we have ended up further from the goal?

Returning to our special case of the LDP, in the absence of a cost or time constraint, a deceptive path may loop or backtrack indefinitely, delaying the LDP without making any progress towards the goal or, as in the following example, causing an agent to traverse *more* truthful steps, not less!

Example 9. *Consider two paths $\pi_a, \pi_b \in \Pi(s, g_r)$ of different lengths taking entirely different routes from the starting location s to the real goal g_r. Suppose that LDP_{π_a} occurs at π_a^{312} and LDP_{π_b} at π_b^{203}. LDP_{π_a} is clearly 'later' if we are counting steps; but does that make π_a the more deceptive path? Suppose that $|\pi_a| - 312 = 200$ and $|\pi_b| - 203 = 10$, that is, there are many more truthful steps from LDP_{π_a} to g_r than from LDP_{π_b} to g_r. This means an agent on π_b (where the LDP occurs 'sooner') gets closer to the goal before the path stops being deceptive; and this, after all, is the main point of the LDP: to establish how far an agent on the path will need to travel truthfully 'in plain sight'.*

Our method of measuring deceptive extent eliminates this confusion by completely ignoring path length, which is potentially infinite and therefore ultimately immeasurable. Instead, we focus on the position of a node in terms of how much 'true work' has been done towards achieving its purpose.

Path completion measures true work done, based on how much there is left still to do.

Now, if a path can be said to have a purpose, then its purpose is its goal. So to capture 'true work', we define the notion of ***path completion***. This concept is similar to 'task completion' in a project plan. No matter what resources have been used up or how many wrong, unnecessary or costly subtasks have been attempted, task completion is ultimately based on

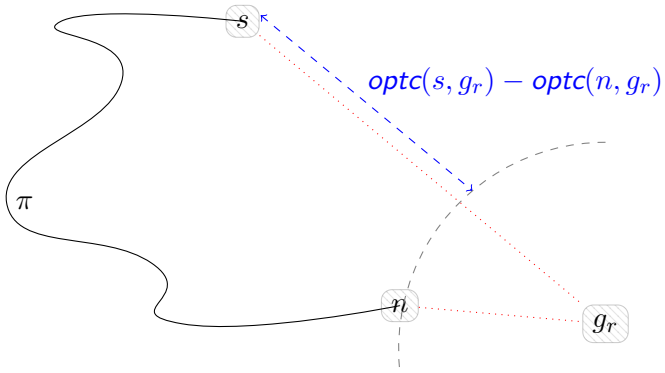


Figure 4.6: Path completion. The dashed blue line represents true work done, the result of Equation 4.7. (Straight lines represent optimal cost, the arc represents points from which the optimal cost to g_r is identical.)

how much work there is left still to do. Similarly, to measure path completion, we use optimal cost (from s to g_r) to calculate how close to achieving its purpose (i.e., reaching its goal g_r) a potentially meandering suboptimal path π has come when it arrives at some particular node $n \in N$. Formally, $pcomp : N \times N \times N \to \mathbb{R}$ is defined as follows:

$$pcomp(n, s, g_r) = optc(s, g_r) - optc(n, g_r). \tag{4.7}$$

Figure 4.6 depicts the equation graphically.

When we measure the position of the LDP in terms of path completion, we learn how far an agent has been able to travel deceptively and (therefore) how far remains to be travelled 'in plain sight'. It is a method that explicitly enables us to distinguish between 'deceiving longer' (e.g., a path that repeatedly circles a bogus goal without making any progress towards the real goal) and 'deceiving further' (i.e., a path that makes progress towards the real goal before it ceases to be deceptive).

With this concept in hand, we can express the notion of delaying the LDP for 'as long as possible' and so maximising a path's deceptive extent.

Definition 18. *A path, π, is deceptive to the* ***maximum extent*** *if $pcomp(\text{LDP}_\pi, s, g_r) \geq pcomp(\text{LDP}_{\pi_i}, s, g_r)$, for all $\pi_i \in \Pi(s, g_r)$.*

In words, π gets closer (or as close) to goal before it stops being deceptive than (as) any other path from s to g_r in the domain. Since we calculate the 'value' of the LDP in terms of path completion, when a path π is deceptive to the maximum extent, we say that its last deceptive point LDP_π has been ***maximised***.

We now have all the tools we need to craft a deceptive path in a principled manner. At the step level, we use GR to evaluate the magnitude of probability values and determine whether (at that step) the path is truthful or deceptive. At the path level, by minimising the number of truthful steps, we maximise the path's deceptive density. Finally, using path

completion, we calculate the position of the path's LDP and so determine its deceptive extent.

To maximise deceptivity, a path must be *strongly* deceptive *to the maximum extent*. We next consider how such a path can be generated.

The Radius of Maximum Probability (Revisited)

So far our framework for deception has been agnostic with regard to the model of the observer (denoted by P in a DPP task, p.118), which may be any probabilistic GR system capable of returning a probability distribution across goals based on a sequence of observations (i.e., the path 'so far'). For the purposes of a practical application, however, it is useful to assume some particular observer and a particular domain. In this case, we take the particular observer represented by the GR model that we developed in Part I of this thesis and describe the process as it applies in a discrete environment. To summarise some of the key properties of our GR model:

- it is derived from the R&G framework, specialised to path-planning;
- it enables us to generate a probability distribution based on a single-observation or step in the path; and
- it enables us to calculate the radius of maximum probability (RMP) for any goal of interest.

The last point is important. Recall that the RMP (defined at p.68 and, for convenience, illustrated afresh at Figure 4.7) is *a cost-distance from goal within which that goal is guaranteed to be the most probable*. Now, compare the LDP (defined at p.120): *a deceptive point in a path beyond which all subsequent points (between the LDP and real goal) are truthful.*

The RMP for the real goal imposes a constraint on the maximum value of any path's LDP.

Redefined in terms of deception, the RMP for the real goal, RMP_{g_r}, could be framed as *a cost-distance from the real goal within which all points are guaranteed to be truthful.* Considered in these terms, we see that RMP_{g_r} measures the distance between the real goal and its closest deceptive point. The LDP (a deceptive point by definition) *cannot be closer to the real goal than this.* Thus, the RMP for the *real* goal, RMP_{g_r}, represents a constraint, imposed by the domain, on the maximum value of any path's LDP.

Under our single-observation model, a probability distribution across goals is generated using $\mathcal{P}_2(\cdot)$, which requires only the single most recent observation (see Equation (3.2), p.56, and (RG2), p.48). For clarity, therefore, we redefine a truthful step for use with $\mathcal{P}_2(\cdot)$, assuming $\vec{o} = \emptyset$ (defined previously in the context of $\vec{o} \cdot n$, p.119).

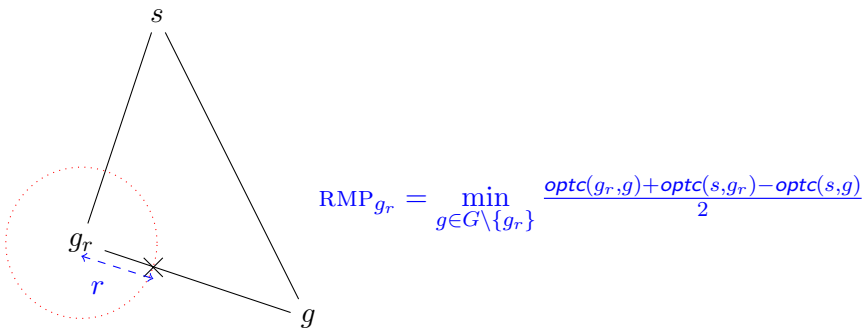


Figure 4.7: RMP revisited. The LDP cannot be closer to goal than RMP_{g_r}. (Straight lines imply optimal cost-distance. The tipping point is marked by a cross.)

Definition 19. *A **truthful** step is a node n at which $P_2(g_r \mid n) > P_2(g \mid n)$, for all $g \in G \setminus \{g_r\}$. Otherwise, the step is **deceptive**.*

As our next result shows, the last deceptive point in a path cannot lie within the real goal's RMP.

Theorem 14. *Let π be a path such that $\pi \in \Pi(s, g_r)$. Then, $optc(\text{LDP}_\pi, g_r) \geq \text{RMP}_{g_r}$.*

Proof. By definitions 7 and 19, for all $n \in N$ such that $optc(n, g_r) < \text{RMP}_{g_r}$, n is truthful. But LDP_π is deceptive. Therefore $optc(\text{LDP}_\pi, g_r) \not< \text{RMP}_{g_r}$ and the proposition follows. □

Corollary 1. *The value of an LDP at RMP_{g_r} from g_r cannot be exceeded: if $\text{RMP}_{g_r} = optc(\text{LDP}_\pi, g_r)$, then $pcomp(\text{LDP}_\pi, s, g_r) \geq pcomp(\text{LDP}_{\pi_i}, s, g_r)$, for all $\pi_i \in \Pi(s, g_r)$.*

Thus, if we know the value of RMP_{g_r}, which we can readily calculate using Equation (3.6) (reproduced alongside Figure 4.7), we can use path completion to find the maximum LDP for the domain. That is:

$$\max_{\pi \in \Pi(s,g_r)} pcomp(\text{LDP}_\pi, s, g_r) = optc(s, g_r) - \text{RMP}_{g_r}. \tag{4.8}$$

This equation provides the benefits previously noted with respect to the RMP but in the setting of DPP. Namely, it enables us to establish a zone within which all points are *truthful*, without having to calculate probabilities at any particular point (or at any point at all). In addition, we can now use it to identify deceptive 'target nodes' at which an LDP can be maximised.

Definition 20. *A **target node** $t \in N$ is a deceptive node such that $optc(t, g_r) \approx \text{RMP}_{g_r}$.*[14]

[14] The approximation $\approx$ is necessary because, in a discrete domain, there may be no node at precisely RMP_{g_r} (as discussed in Part I, p.70). In this case, an approximation (always greater) suffices; and we 'retreat' to the closest actual node, as described in the text.

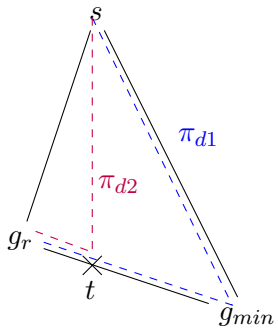


Figure 4.8: A target node at t. Paths generated by strategies 1 and 2 (p.127) are shown.

Note that a target node is, by definition, deceptive; but not all (and perhaps few) nodes at RMP_{g_r} from g_r are deceptive. (The RMP for the real goal signifies a radius *within* which all nodes are truthful, which does not mean that all nodes *outside* that radius are not.) We can, however, identify a suitable target node, as follows.

Recall that our calculation of the RMP involved identifying a 'tipping point' between goals (marked by a cross in Figures 4.7 and 4.8). In the context of DPP, that process now provides a simple means of locating (programmatically) a target node which (a) is guaranteed to be accessible (its location is calculated based on the cost of an optimal, navigable path between goals); and (b) is a point at which the LDP is guaranteed to be maximised.

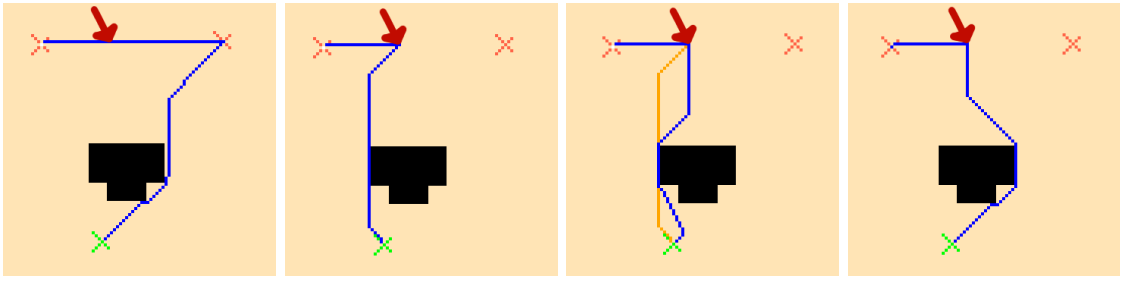
In developing deceptive strategies (in Section 4.3), we use $g_{\min}$ to denote the goal referenced in calculation of RMP_{g_r}. It is on the optimal path between $g_{\min}$ and g_r that the tipping point closest to g_r is located.

$$g_{\min} = \underset{g_i \in G}{\operatorname{argmin}} \frac{optc(g_r, g_i) + optc(s, g_r) - optc(s, g_i)}{2}. \tag{4.9}$$

As a brute-force solution, we could find a suitable target node by performing a best first search starting at the real goal g_r and continuing until we reach the first node at a distance greater than RMP_{g_r} from that goal. Alternatively, referring to Figure 4.8, we can first identify g_{min} using Equation 4.9, then retreat along (g_r, g_{min}), the optimal path between the goals, to a distance of RMP_{g_r} (i.e., the point marked by 't'). In a continuous domain, we can specify our target at precisely this point (the distance, calculated as a real, RMP_{g_r} from g_r); in a discrete domain, however—where there may be no node at that exact location—we continue to 'backtrack' along (g_r, g_{min}) to the first *actual* node that we encounter.

4.3 Deceptive Strategies

The following approaches involve the computation of paths whose deceptivity is maximised in terms of extent. We consider how the two fundamental deception strategies, simulation



(a) Path π_{d1}. (b) Path π_{d2}. (c) Path π_{d3}. (d) Path π_{d4}.

Figure 4.9: Deceptive path-planning strategies.
The above screenshots show automated path-plans through a gridworld domain (n.b., the algorithm draws the diagonal path segment first, then the vertical, rather than alternating between the two, making some optimal paths appear suboptimal to the human eye). The green cross (bottom centre) marks the start of the path and the red cross (top left) is the real goal. The red arrow overlaid indicates the location of the target node t. Path π_{d3} is superimposed on π_{d2} to highlight the differences. Paths π_{d1} and π_{d4} both have maximum deceptive density but π_{d4} is optimal (amongst deceptive solutions that pass through t) with respect to cost.

(showing the false) and dissimulation (hiding the real) can help to maximise such paths' deceptivity and minimise their cost. Screenshots from simulations of the four strategies are reproduced at Figure 4.9 with corresponding algorithms in Listings 4.1 to 4.5.[15]

4.3.1 Simulation

Deceptive strategy 1 (π_{d1}). The simplest simulation (considered by Keren et al. (2015) with respect to 'bounded deception') first takes an optimal path towards a bogus goal. Referring to Figure 4.8, this strategy generates $\pi_{d1} = s, .., g_{\min}, ..., g_r$. Computationally inexpensive, this achieves a *strongly* deceptive path (every step to the LDP is deceptive), maximises deceptive density (by minimising truthful steps) and maximises deceptive extent (its LDP occurs at RMP_{g_r}, in this case, t). However, path cost is likely to be high and, although it initially deceives both human and automated observers, reaching but not stopping at $g_{\min}$ immediately signals to a human that $g_{\min}$ is *not* the real goal; though an automated observer depending on the R&G model is deceived all the way to t.

4.3.2 Dissimulation

Deceptive strategy 2 (π_{d2}). Dissimulation seeks an ambiguous path. The simplest such strategy takes an optimal path π_{d2} direct from s to t, then on to g_r. This generates the cheapest path that can pass through t. It *might* be deceptive to a human observer until later in the path than π_{d1} above. However, we would expect π_{d2} to be only *weakly* deceptive, that is, truthful steps are likely to occur before the LDP without additional checks and balances. We therefore propose two refinements:

[15] Algorithms assume availability of a path-planner, similar to the A* listing in Chapter 2 (p.15).

Listing 4.1: Strategy 1.

```
1  Requires: starting node, s; possible goals, G; real goal r
2  Returns: path or failure
3
4  RMP, argmin = getrmp(s, G, r)
5
6  path1 = buildPath(s, argmin) //returns shortest path
7  path2 = buildPath(argmin, r)
8
9  if not failure then
10    return path1 + path2
11
12 return failure
```

Deceptive strategy 3 (π_{d3}). A path π_{d3} can be assembled using a modified heuristic so that, while still targeting t, whenever there is a choice of routes, it favours the bogus goal, increasing its likelihood of remaining deceptive. Still using an off-the-shelf path-planner, the usual heuristic $h(n, t)$—which returns the estimated cost from node n to target h—is modified to also evaluate heuristics for g_r and $g_{\min}$:

$$\text{if } h(n, g_r) < h(n, g_{\min}) \text{ then } h(n, t) = \alpha h(n, t),$$

where constant $\alpha > 1$. Path π_{d3} is computationally more demanding than π_{d1} or π_{d2} (it evaluates more heuristics), but aims to approach π_{d1}'s deceptive density at something close to π_{d2}'s cost.

Deceptive strategy 4 (π_{d4}). As an alternative—and perhaps definitive refinement with respect to cost—we can use single-observation recognition to precalculate a heatmap of probabilities (as described in Section 3.1.3, p.55)—or calculate them on-the-fly—to prune truthful nodes in the search. The resulting path π_{d4} is strongly deceptive with maximised LDP and maximum density at minimum cost.

Contrary to the common idea of a deceptive path—that is, a rambling suboptimal path, full of expensive loops and unnecessary detours—this brute force strategy demonstrates that there is such a thing as a fully deceptive path that is also fully rational.

Differences between strategies are highlighted at Figure 4.9.

4.3.3 Evaluation

Though not proposed as optimised algorithms, we evaluated the relative efficiencies (time/cost) of the above strategies and the effectiveness (deceptivity) of paths they can produce. We generated a problem set based on game maps from the Moving-AI benchmarks (Sturtevant, 2012) to which we added three extra candidate goals at random locations. For each of 50 problems, we generated one optimal path using a standard implementation of A*

Listing 4.2: Strategy 2.

```
1  Require: starting node, s; possible goals, G; real goal r
2  Returns: path or failure
3
4  RMP, argmin = getrmp(s, G, r)
5  t = findTarget(RMP, argmin, r)
6
7  path1 = buildPath(s, t) //direct to target
8  path2 = buildPath(t, r)
9
10 if not failure then
11    return path1 + path2
12
13 return failure
```

Listing 4.3: Strategies 3 and 4.

```
1  Require: starting node, s; possible goals, G; real goal r
2  Returns: path or failure
3
4  RMP, argmin = getrmp(s, G, r)
5  t = findTarget(RMP, argmin, r)
6
7  path1 = customAstar(s, t, argmin) //calls one of customised routines (below)
8  path2 = buildPath(t, r)
9
10 if not failure then
11    return path1 + path2
12
13 return failure
```

Listing 4.4: Heuristic routine (Strategy 3).

```
1  Require: t, r, argmin and current coord, c
2  Returns: Real
3
4  tHeur = octile(c, t) //calculates usual heuristic
5  rHeur = octile(c, r)
6  aHeur = octile(c, argmin)
7
8  if rHeur < aHeur then
9     tHeur = tHeur * 1.5 //constant
10
11 return tHeur
```

and four deceptive paths (each using a different strategy). We timed path generation and recorded path costs. We truncated paths at the RMP (beyond which all paths would be truthful) and, using single-observation recognition (Section 3.1, p.42), calculated probabilities at intervals to assess (and confirm) deceptive density and extent.

Figure 4.10 captures our results. Comparison with A* shows a clear trade-off between

Listing 4.5: Truth check (Strategy 4).

```
Require: starting node, s; goals, G; real goal r, current coord c
Returns: true or false

for g in G
   if g not r
      costdif(s, g, c) <= costdif(s, r, c) then
      return false

return true

costdif(start, goal, coord)
   return optCost(start, goal) − optCost(coord, goal)
```

	Path cost	Gen. time	10%	25%	50%	75%	90%	99%
π_{A^*}	215.9	0.208	78	68	40	32	22	12
π_{d1}	375.2	1.378	**100**	**100**	**100**	**100**	**100**	**100**
π_{d2}	245.2	1.997	92	88	76	72	62	74
π_{d3}	245.6	1.924	90	90	72	66	68	70
π_{d4}	248.7	1423.8	**100**	**100**	**100**	**100**	**100**	**100**

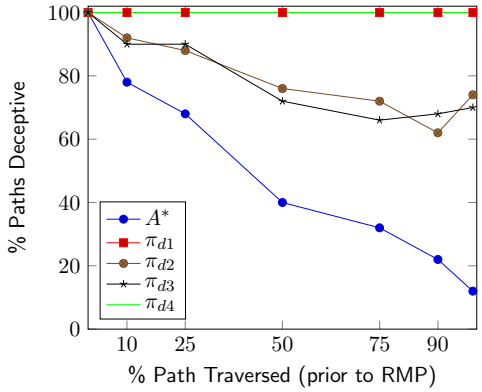


Figure 4.10: Deceptive paths.

Results show the percentage of paths returned by each strategy that were deceptive when tested at 10%, 25%, etc., of their path length *prior to the RMP* (beyond the RMP, all paths were truthful). Table columns show average (total) path costs and average time taken to generate the (total) path. Generation time for all strategies exceeded that of A* by an order of magnitude. See inline text for discussion of time taken to generate π_{d4}. π_{d1} and π_{d4} were both strongly deceptive to the full extent but π_{d4} achieved this at much lower cost. (Two maps, 50 Moving-AI scenarios, each modified to include three extra goals. Experiments were conducted on a i7 3.6GHz machine with 8GB RAM.)

cost and deceptivity. Strategies π_{d2} and π_{d3} returned comparatively cheap paths and were computationally efficient but performed erratically, each showing an increase in the number of paths deceptive immediately before the RMP. This implies that they are only *weakly* deceptive (i.e., deceptive nodes may follow truthful ones). Simulation (π_{d1}) was strongly deceptive but the least efficient strategy in terms of cost. Dissimulation with pruning (π_{d4}) was fully deceptive at much lower cost. (In fact, as we know, paths generated using this strategy are optimal amongst deceptive paths.) Although generation of π_{d4} was slow, this is only because we calculated each node's deceptivity on-the-fly. If repeatedly considering deceptivity in a known domain, a probability heatmap could be precalculated (as previously discussed), enabling truthful nodes to be pruned in constant time, so closing the time difference between this and other strategies.

4.4 Discussion

In the three previous sections of this chapter, we have considered deception from the point of view of a GR system, responding to observations consistent with 'deception by dazzling', and from the deceptive path-planner's point of view, developing a model that facilitates evaluation and generation of deceptive paths. In this section, we discuss some of the many refinements, avenues and extensions still waiting to be explored. First, we look at some of the more sophisticated aspects of Bell and Whaley's theory of deception. Then, we examine more thoroughly the notion of magnitude, which we use in our model to determine whether any one particular step in a path is truthful or deceptive but which can also tell us the extent to which the path is simulating or dissimulating from point to point. Finally, we consider truthful path-planning, the flip-side of the model we presented in Section 4.2.

4.4.1 More Sophisticated Strategies

In setting out their theory of deception, Bell and Whaley (Bowyer, 1982) subdivide the broad strategies of simulation and dissimulation into six sub-stratagems, revisited in Table 4.5. Of these, we have examined only two: simulation by mimicking (which could be said to describe the strategy of targeting a bogus goal before diverting towards the true goal); and dissimulation by dazzling (taking a wildly suboptimal route, the main topic of Section 4.1). The other sub-strategems can also be achieved in a path-planning context although, in some cases, the route-planning type of domain on which we have focused does need slight enhancement.

Masking

Masking is simply another word for hiding. In a realistic three-dimensional terrain, this can be achieved quite literally by keeping an obstacle between the agent and the observer but there are other possible ruses.

For purposes of GR, we assume that observations are incomplete. Conversely, for DPP, although observations are unlikely to be continual, since we cannot know precisely when they will take place, we must plan our path on the assumption that all the agent's movements are fully observable. If, however, there were some means of controlling the availability of the agent's location (e.g., because the source of observations is the signal from the agent's mobile phone, which can be switched off, as suggested by Keren et al., 2016), the agent would not only be able to hide but could then adopt a wide variety of alternative strategies. For example, she might simulate (by mimicking), taking a course that particularly favours a bogus goal, maximising its probability by comparison to the others, then turn off the phone and track directly (by the shortest, quickest or most hidden path) to the real goal.

Table 4.5: Bell and Whaley's six strategies revisited.

Dissimulation	**Simulation**
Masking - hide the real by making it invisible.	**Mimicking** - show the false by having one thing imitate another.
Repackaging - hide the real by disguising.	**Inventing** - show the false by displaying an alternate reality.
Dazzling - hide the real by creating confusion.	**Decoying** - show the false by diverting attention.

Note. With full observability, masking can be achieved by dazzling or 'hiding in plain sight', the strategy already discussed, whereby the agent traverses a path so randomised that no clear intention can be identified.

Repackaging

Repackaging involves adopting a disguise. In a path-planning domain, there does not immediately seem to be a 'charc' (i.e., characteristic) that can be disguised. However, there is cost.

Cost-based GR assumes a cost model. If the agent can persuade her observer (whether AI or human) that she is following a cost model different from the one that she is actually following, then she can disguise her intent. Suppose, for example, that the agent's goal is located on the coast and suppose that the deception is to take place at some future date (i.e., she has time to set it up). By repeatedly taking paths that favour a coast road, even when her ultimate goal is inland, she could give the impression that her cost model favours the coast (i.e., that she evaluates coast roads as being cheaper than inland roads). Now, on the foray of importance, she can take the coast road *directly to her coastal goal* leaving her observer unable to determine whether it is the coastal goal she is making for or the inland goal, via a coastal route.

Inventing

Similar to mimicking (which, in path-planning we have suggested may be achieved by seeming to target a bogus goal, only to deviate at the last moment towards the real goal), inventing involves fabrication of a false reality. Simplistically, in path-planning, one might behave as for mimicking but move directly towards some non-goal feature of the terrain, giving an impression of intentionality where none exists.

More creatively, a deceptive path-planner can make use of a fundamental strategy used in magic, known as 'one-ahead'. In the context of a card trick, for example, audience members believe that they know when the trick started (e.g., when they were offered a deck of cards from which to choose) not realising that the *real* start of the trick occurred

at the end of the *previous* trick (e.g., at the moment when, while replacing the cards on the table, the magician checked the value of the card on the bottom of the deck).

Applying this principle to deceptive path-planning, recall that the cost-based GR systems discussed in this thesis have all assumed a known starting location. If an agent can persuade the GR system (her observer) that her start location is different from her actual start location, she can significantly change the underlying domain and thereby (potentially) the probability distributions at key locations on her preferred path to goal. Concretely, this could be achieved if she were able to 'mask' the movement from her real start location to her preferred bogus starting location so that, from the observer's point of view, the first significant observation involves the agent setting off from her bogus starting point.

Decoying

Decoying shows the false by diverting attention. A conjuror would call it 'misdirection'; a ball-player might call it 'selling the candy'. In path-planning the strategy suggests a zig-zagging route between goals. Every time the observer thinks she knows where the agent is going, the agent changes direction and sets off the opposite way. We note also that, though beyond the scope of this thesis, there is an emerging trend towards the development of *multi*-agent path-planning (e.g., Le & Plaku, 2018) that has the potential to facilitate a more traditional approach to decoying whereby, for example, one agent A (deemed dispensible) is visibly sent off in some direction, sacrificed for the benefit of another agent B (deemed more valuable), sent elsewhere but only when the observer has commited to following agent A.

4.4.2 Deceptive Magnitude

Our model has treated deception as a yes or no proposition. If the probability of the real goal exceeds the probability of all others, then the step is truthful, if not, it is deceptive. This is consistent with many real-world applications: a dupe (whether we regard her as observer or spy) either *is* deceived or she is not. In evaluating a path's deceptivity, however, it is possible for us to achieve a (potentially more useful) graduated response: how much more likely is it that an observer would be deceived by this path rather than that path?

To suggest the possibilities, consider how we might capture the two distinct notions of simulation and dissimulation at the step level (i.e., at a node n).

Definition 21. ***Simulation*** *(showing the false) occurs when the probability of a bogus goal is strictly greater than the probability of the real goal g_r, that is, there exists a goal $g \in G \setminus \{g_r\}$ such that $P(g_r \mid \vec{o} \cdot n) < P(g \mid \vec{o} \cdot n)$.*

We can quantify simulation by measuring the amount by which a false goal dominates the real goal. The greater the dominance, the greater the deception.

$$simulation(\vec{o} \cdot n) = \max_{g \in G \setminus \{g_r\}} P(g \mid \vec{o} \cdot n) - P(g_r \mid \vec{o} \cdot n). \tag{4.10}$$

Recall our example of a convoy, escorting a VIP to one of three possible destinations (p.117). If we simulate successfully, our hypothetical assassin is deployed to the wrong location and the VIP survives.

Note. The result of (4.10) is in the range [-1, 1] and tells us more than just the degree of simulation that has been achieved. A negative result indicates that the step is truthful (i.e., non-deceptive: the probability of the real goal is highest). A zero result indicates dissimulation (i.e., there is at least one bogus goal whose probability equals that of the real goal).

Definition 22. ***Dissimulation*** *(hiding the real) occurs when the probability of the real goal g_r is less than or equal to the probability of another goal, that is, there exists a goal $g \in G \setminus \{g_r\}$ such that $P(g_r \mid \vec{o} \cdot n) \leq P(g \mid \vec{o} \cdot n)$.*

Following Bowyer (1982), this definition of deception *always* involves dissimulation and *may also* involve simulation. We can quantify that aspect of deception exclusive to dissimulation (the degree of ambiguity) using Shannon's entropy:[16]

$$dissimulation(\vec{o} \cdot n) = -\kappa \sum_{g \in G} P(g \mid \vec{o} \cdot n) \times log_2(P(g \mid \vec{o} \cdot n)), \tag{4.11}$$

where κ is a normalisation constant.[17]

In the context of our convoy, successful dissimulation means the controller will not know where to send the assassin; though she may guess correctly.

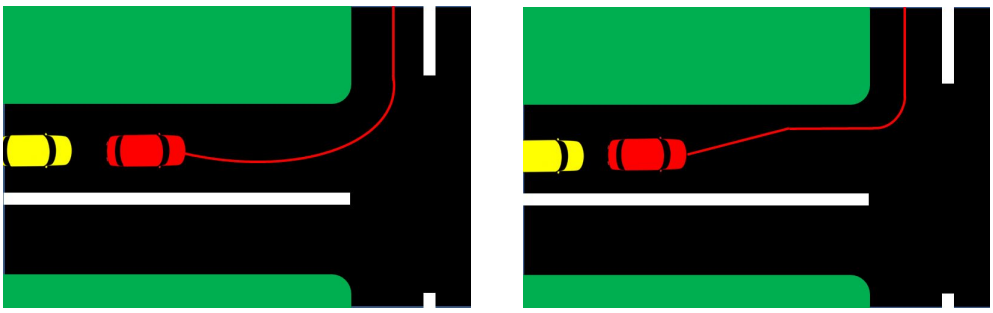
4.4.3 Truthful Path-Planning

Recall from Chapter 2 (p.22) that, traditionally, GR can be categorised into three types: keyhole, intended and adversarial. This thesis has considered keyhole recognition, in which the observed agent is unaware of (or unaffected by) the GR process, and adversarial recognition, where the agent believes herself observed and attempts to thwart the recognition process. Intended recognition is just the opposite; it arises when the observed agent assumes herself observed and attempts to *reveal* her goal.

Once neglected, intended recognition has lately become an important area of research owing to the explosion of interest in human-machine interaction (e.g., Kulkarni, Srivastava,

[16] Shannon's entropy is used in information theory to measure information gain. Any change towards equalisation of probabilities increases the uncertainty, which increases the entropy.

[17] The range of Shannon's entropy is $0 \leq entropy \leq log_2(n)$ where n is the number of possible outcomes, so it is convenient to use $\kappa = log_2(|G|)$ to normalise (4.11) in the range [0-1].



(a) A vehicle minimising cost or travel time may take a deceptive path.

(b) A vehicle minimising the probability of other goals takes a truthful path.

Figure 4.11: Truthful path-planning.

& Kambhampati, 2018; Sreedharan et al., 2017) and several authors have used probabilistic reasoning to identify transparent or legible (as opposed to 'predictable') motion (Dragan, Lee, & Srinivasa, 2013; MacNally, Lipovetzky, Ramirez, & Pearce, 2018).

In this context, our model, though developed with a view to identifying and evaluating *deceptive* paths can similarly be inverted to identify truthful or *least* deceptive paths. Now, the problem becomes one of finding a path from starting point to goal such that the observer can determine *as quickly as possible and with minimal computational effort* where the agent is going. For example, an autonomous vehicle approaching an intersection could take the least deceptive path in order to convey to other vehicles (whether autonomous or human-driven) a human-like non-verbal signal of its intended exit.

Only a minor modification to our current model would be required. Recall that we distinguish between dissimulation, which can be satisfied by an ambiguous solution, and simulation, which is only satisfied while the probability of an alternative goal is strictly greater than the probability of the real goal. Now, when we use simulation to increase a path's deceptivity, we seek to maximise the probability of some (any) alternative to the real goal. So, the *magnitude* of the simulation (repeated here from Definition 4.10, p.134 with respect to a DPP problem) is given by:

$$\mathit{simulation}(\vec{o} \cdot n) = \max_{g \in G \setminus \{g_r\}} P(g \mid \vec{o} \cdot n) - P(g_r \mid \vec{o} \cdot n).$$

Reformulating the above equation to identify the most truthful sequence of observations or path π_t, we could say that, given a set Π_{g_r} of all possible (legal) paths to g_r:

$$\pi_t(G, g_r) = \arg\max_{\pi \in \Pi_{g_r}} P(g_r|\pi) - P(g'|\pi)$$

$$\text{where } g' \in G \text{ is such that } P(g'|\pi) \geq P(g|\pi) \text{ for all } g \in G \setminus \{g_r\}.$$

The impact of employing this probabilistic measure of truthfulness is that, rather than minimising cost, the preferred path minimises the probability of any alternative goal.

Example 10. *Consider the situation depicted in Figure 4.11. A vehicle is preparing to turn left. The most efficient trajectory (to save time) is to start the manoeuvre from as far to the right as possible and make a shallow turn to the left at high speed. The vehicle's primary objective, however, is to demonstrate its intent and therefore it moves to the left side of its current lane before making the turn. The vehicle maximises the opportunity to display its intent rather than minimising its costs.*

As when implementing particular deceptive strategies, although a model for truthful path-planning may be agnostic, in order to devise a particular plan, it can only be 'truthful' relative to some particular observational model. Here we assume a cost-sensitive observer. If we assume that observers evaluate probabilities based on alternative criteria, this inverted model to support truthful path-planning (or 'intended' recognition) would still apply but might generate quite different trajectories.

The extensions and adaptations discussed in this section are by no means exhaustive. They are preliminary ideas and suggestions to illustrate how much is yet to be explored with regard to deception, even in path-planning.

4.5 Summary

In this chapter, we examined deception in the context of path-planning, focusing particularly on the simulation and dissimulation strategies of Bell and Whaley.

First, we analysed the responses of two contemporary GR systems when faced with observation sequences consistent with dissimulation by dazzling and found that the probability distribution formulas that they use have unintended and unhelpful consequences when faced with paths that appear to be irrational. We presented an alternative model based on a self-modulating formula, which takes the observed agent's history into account. Our formula respects the principle (from Ramirez and Geffner's probabilistic model, 2010) that, typically, a plan's probability is inversely proportional to its cost; but it degrades gracefully if the underlying assumption of rationality is compromised. As a result, the improved model can handle agents ranging from the strictly rational (and honest) to the arbitrarily irrational (or dishonest) in a principled manner.

Although dissimulation by dazzling can confuse a GR system, it is unnecessarily expensive with no means of assessing whether or not 'deceptivity' has been achieved. In the second half of this chapter, we addressed the problems of (a) measuring deceptivity, and (b) controlling its cost. Our model for DPP operates at three levels of granularity: magnitude, density and extent. We introduced the notions of a first truthful point and last deceptive point and, rather than counting steps or measuring a path's length, used path completion (similar to 'task completion' in a project plan) to identify distance travelled along a suboptimal path. Using an observational model based on our GR system from Part I,

we considered strategies to compute deceptive paths, including one capable of achieving paths that are 'optimally deceptive', that is, continually deceptive to the maximum extent at the lowest cost. Finally, we considered potential extensions to our DPP model.

In summary, this chapter made the following contributions.

- We presented a definition of rationality applicable even when the ground truth is unknown; and we provided a rationality measure (RM) based on an agent's current behaviour which could be used as a predictive tool.
- We analysed the performance of two state-of-the-art GR systems confronted by increasingly irrational behaviour and showed that one (Ramirez & Geffner, 2010) becomes *more* confident in predicting the most probable goal as the observed behaviour becomes *less* rational, while the other (Vered et al., 2016) ultimately evaluates as most likely whichever goal is most distant (or most expensive to reach) from the start location.
- We developed a self-modulating formula that lifts the rationality assumption implicit in many contemporary cost-based accounts of GR: the less rational the observed agent's actions seem to be, the less confident its prediction.
- We presented a model for DPP that can use (as a black box) any probabilistic GR system capable of measuring the deceptivity of a path at a given point. Our model measures deceptivity in terms of magnitude, density and extent.
- We introduced the notion of a last deceptive point (LDP) and, in path completion, provided a reliable method of identifying its location even in a suboptimal path. A path continually deceptive up to its LDP is strongly deceptive to the maximum extent. The cheapest path deceptive up to this point is *optimally* deceptive.
- We suggested that our DPP model could be extended to measure simulation and dissimulation at step level or could be inverted for use in truthful path-planning, also known as 'intended recognition'.

CHAPTER 5

Conclusion

"Everything should be made as simple as possible, but not simpler."

–Albert Einstein (attrib.)

Path-planning is an uncomplicated undertaking. A path-planning agent has a destination and chooses the 'best' way to get there. A surveillance agent, assuming herself to know the definition of 'best' being applied, asks herself, "If it were me taking that route, where would I be going?" On the 'flipside', a deceptive path-planner assumes she is being watched and asks herself, "If I want to confuse my observer or make her think I am going somewhere I am not, which route should I take?" Simplifying the problem back to these terms, the main achievement of this thesis has been to demonstrate that, provided we know an agent's starting point, goal recognition in path-planning depends on the nature of the domain, not on the observation history of the agent whose goal we are trying to determine. This realisation not only facilitates swift performance of goal recognition (using single-observation recognition, a heatmap or the radius of maximum probability); applied in the context of deceptive path-planning, we see that the maximum extent of a deceptive path (i.e., its last deceptive point) is also constrained, not by the agent's behaviour, but by the domain itself.

At the start of our literature review, we observed that preliminary explanations of task-planning algorithms are often expressed in relation to a 'toy' gridworld domain. People frequently construct graph-like brain maps to help them understand potentially confusing ideas. But while simplistic navigational domains are frequently used as an example, we have followed the "Keep It Simple, Stupid" principle and made them our focus. It is an approach that has the advantage of exposing fundamental aspects of the core problem (which can easily become obscured by confounding variables) without sacrificing the possibility of generalising lessons learnt back to task-planning at a later stage.

We began in Chapter 2 by reviewing the literature and found that long-established path-planning techniques continue to be employed in contemporary applications. In goal recognition, we found a focus on minimising computational cost and saw in the recent emergence of 'plan recognition as planning'—whereby goal recognition can be achieved using automated planning techniques, with no dependence on a plan library—a method well-suited to path-planning domains. We saw the problem encountered by authors seeking to implement Ramirez and Geffner's approach (2010) whilst avoiding the computational cost of negative reasoning and observed that, although a simpler formula was in use, it had not been subjected to formal scrutiny. We observed that, until recently, goal recognition for task-planning has concentrated on 'keyhole' recognition with an expectation of rationality and that 'adversarial' recognition, where it does appear, has typically been either domain-specific or studied in terms of anomaly detection. In relation to deception itself, we found that, although it has been discussed since the very inception of AI, it has usually been considered with a view to prevention, for example, in the context of network security. Active deception (where the agent deliberately sets out to deceive) has mostly arisen incidentally in a game-theoretic context, where the agent's behaviour is expedient rather than deliberate: that is, she is doing whatever is most likely to optimise the reward, deceptive or not. Meanwhile, computational theories of deception (also game-theoretic) seemed to concern themselves more with ethical considerations than any systematic approach as to how deception could best be achieved. Furthermore, we found little discussion as to how the potential deceptivity of a sequence of actions might be measured. Even looking outside computer science, we found the 'measurement' of deception only in the context of ethical considerations (i.e., how relatively 'bad' each different type of deception should be considered). These are the gaps that our work has aimed to fill.

In Chapter 3, we asked the first of our research questions: *"What gains can we make when we apply state-of-the-art task-plan recognition in core path-planning domains?"* We found many gains and, indeed, have been able to minimise the computational effort required to determine an agent's most probable destination to the point where we only need to know the relative locations of the possible goals and the agent's starting point to calculate a radius around a goal within which that goal is *guaranteed* to be the most probable. In arriving at our radius of maximum probability (Definition 7, p.68), we have made several contributions.

- We presented a novel cost difference formula (Equation 3.2, p.56) that can be used in discrete and continuous navigational domains. Provided that the start location is known in advance, it relies on a single observation, whereas competing goal recognition techniques depend on an agent's complete observation history. Our formula can be plugged into any probability distribution formula that enforces the principle that 'the lower the cost difference, the higher the probability' including the Boltzmann probability distribution formula used by Ramirez and Geffner (2010) in their

seminal work, where they first introduced the idea of using a cost difference formula in this context.

- While others (e.g., Escudero-Martin et al., 2015; Vered et al., 2016) have recognised that the negative reasoning used by Ramirez and Geffner (i.e., negative because it compares an optimal plan via the observations with an optimal plan that does *not* go via all of them) can usually be avoided by substituting a simpler term, we are the first to prove that the replacement impacts the result in only one corner case (Theorems 1, 2 and 3, pp.50-52). We are the first to define that case; and we also show that the Ramirez and Geffner interpretation in that corner case is not necessarily to be preferred (see Section 3.4.1, p.80). Our results provide formal justification for the previous informal treatments, which do not demonstrate what is lost (or gained) by the substitution.

- Writing in parallel with us, Vered and Kaminka (2017) identified (as we do) that when calculating the cost of a path through the observations, the first part of that path (from starting point to most recent observation) is the same for every goal. They reuse their calculation; however, we show that—used with Ramirez and Geffner's probability distribution formula—that cost can actually be disregarded altogether. Furthermore, they imply that their reuse of that cost (from initial state to final observation) is applicable for task-planning domains in general, whereas—as we discuss in Section 3.4.2 (p.84)—it turns out not to be applicable in task-planning domains generally, not even in slightly enriched path-planning domains.

In Chapter 4.1, we asked: *"How can we improve on the response of goal recognition systems when the observed agent's behaviour is (apparently) irrational?"* Approaches to keyhole recognition typically assume rationality. However, for probabilistic accounts, this is a soft assumption since, simultaneously, they rely on suboptimality to construct a probability distribution across goals. Furthermore, a blanket assumption ignores the fact that, typically, an agent engaged in goal recognition cannot know in advance that it is not dealing with an adversary, so the assumption of keyhole recognition (or intended recognition) may be inadvisable in many domains (e.g., most obviously, surveillance).

- We analysed the output from the distribution formulas used by Ramirez and Geffner (2010) and Vered et al. (2016) and demonstrated anomalies in their behaviour when confronted by suboptimally (Theorems 8 and 9, pp.105-108).

- We suggested an approach that expects keyhole recognition while nevertheless accommodating the possibility that the observed agent may be adversarial (or behaving suboptimally for other reasons). Our self-modulating formula (Equation 4.5, p.110) 'lifts' the rationality assumption. It not only self-adjusts its level of confidence rel-

ative to the apparent irrationality of the observed agent but is also more consistent than either of the state-of-the-art formulas that we analysed.

In Chapter 4.2, we asked: *"How can we define and generate measurably deceptive paths?"* We presented our model for deceptive path-planning, built on ideas derived from the theory of deception that first appeared in the book "Cheating" by military strategists Bell and Whaley (Bowyer, 1982). Although, as acknowledged in Section 4.4, there is considerable scope to apply other aspects of their theory in future work, our model is not just a single strategy or one-off formula.

- Our model inverts probabilistic goal recognition so that, instead of measuring which goal is most likely, we examine the relative *un*likeliness of the real goal.
- The model provides a mechanism for the first time (to our knowledge) whereby the relative deceptivity of multiple paths can be evaluated and compared. Importantly, many different strategies (in addition to the four that we suggest, pp.126-128) can be developed with the objective of maximising the target dimensions that our model defines.
- We introduced the notions of a last deceptive point and a first truthful point (Definitions 14 and 15, p.120): concepts that seem obvious once stated but which are less obvious outside the path-planning context (although they still apply). Based on these, we defined three dimensions against which to measure the path's potential to deceive: extent, density and magnitude.

We conclude by drawing attention to several interesting overlaps between Keren's work on goal recognition design (Keren et al., 2014, 2015, 2016) and our work on deceptive path-planning.

First, their concept of worst case distinctiveness (which measures distance from the initial state to a point where disambiguation can be achieved) is superficially similar to our last deceptive point. Worst case distinctiveness, however, is a property of the domain (not the path) and is measured by counting steps. The problem with this approach is that, since the length of a suboptimal path is potentially infinite, as soon as an attempt is made to accommodate suboptimality (Keren et al., 2015), worst case distinctiveness is potentially infinite too! To handle this situation, the authors use a process that they call 'bounded non-optimality', that is, they cap the number of permissible steps within a fixed budget. This is a constraint that could be avoided if, instead of measuring distance-travelled in terms of steps (or plan-length from the initial state), they adopted a method such as path completion (Section 4.2.3, p.121) to measure distance travelled along a suboptimal path.

Computation of worst case distinctiveness is ingenious but computationally expensive. Since, like us, Keren et al. (2014) applies the goal recognition model from Ramirez

and Geffner (2010), our RMP calculation (Section 3.2, p.61) might be used in appropriate domains (e.g., real-world path-planning) to abbreviate calculations. Instead of working from the initial state, as Keren does, our model could provide a computationally economical means (with no dependence on classical planning) of finding the precise cost-distance *from* goal at which 'distinctiveness' is achieved.

It is worth noting, finally, that modification of a domain using goal recognition design actually *facilitates* deceptive path-planning because it allows simulation (i.e., the unambiguous targeting of a bogus goal) to begin at an earlier point in the path than would otherwise have been possible.

5.1 Limitations

Our model of single-observation recognition inherits certain limitations from Ramirez and Geffner's account (2010) from which it derives. First, it assumes rationality. Also, as highlighted by Sohrabi et al. (2016), it fails to accommodate the possibility of noisy or missing observations (e.g., actions that were recorded but did not occur or that did occur but went unrecorded as a result of unreliable equipment, such as a malfunctioning sensor).

Our model of deception can use any goal recognition system as a 'black box'. Nevertheless, it is limited by the assumption that the deceiver has access to that system (or at least the cost model that it is using). There is also an assumption that the observer is naive. If, for example, the same observer were to see the deceiving agent on multiple occasions, one might expect the goal recognition system to adapt (e.g., using a machine learning approach such as that described by Liao, Patterson, Fox, and Kautz, 2007). No provision is made for this, however, nor for double-bluffs or counterplanning (Pozanco et al., 2018).

5.2 Future Work

In Part I, we showed that single-observation recognition ranked goals in the same order as the Ramirez and Geffner (2010) model. However, in order to ensure that probability values were similar, we found it necessary to add a large constant to our cost difference results. In Part II we showed that this anomalous behaviour resulted from using the Boltzmann equation as a probability distribution formula and we showed that our non-sigmoidal variation would return probabilities at the limit of that formula. Our interest there was to amend our self-modulating goal recognition formula in such a way that it would provide a correct and consistent baseline to be adjusted strictly according to our level of confidence (flattening the probability distribution as the agent's behaviour becomes less rational). With this improved understanding, it would be useful now to explicitly extend our model for single-observation recognition, using our improved version of the

probability distribution formula. With that minor change (which corrects an anomaly in Ramirez and Geffner's original output), the probability distribution achieved using single-observation recognition should become identical to that obtained using Ramirez and Geffner's formula, not only in terms of rank, but in terms of the probability values themselves; and this applies in all but the one corner case which, as discussed in Section 3.4.1, differentiates between goals in a subtle way that can sometimes be more misleading than useful.

A second—and perhaps even more useful—extension would be to fully generalise our solution back to task-planning. As discussed in Section 3.4.2, this involves either the (unreasonable) assumption of full observability or determining—in a domain-independent fashion—precisely which fluents need to be observed in order to extrapolate accurate costs from the most recently observed action to each candidate goal. Although this is a substantial piece of work, it may be achievable. As a brute-force solution, for example, one might work back from each goal through all possible preconditions to the problem's initial state to arrive at a definition of partial observability that would suffice.

Building on the above, the notion of a heatmap or a radius of maximum probability for task-planning are interesting propositions: the idea of being able to 'look up' an action and immediately determine an agent's most likely objective is an attractive one.

Our self-modulating formula for goal recognition does not represent a typical approach to adversarial recognition. Taking a cost-based approach, we identify behaviour that is suboptimal with respect to every goal as being symptomatic of irrational (or rational but deceptive) behaviour. This could be characterised as domain-independent anomaly-detection—and, indeed, it can be used that way—but this was not our primary objective. Rather, we wanted to factor into *keyhole* recognition the possibility of encountering an adversarial agent. The crucial difference is that, whereas an adversarial recognition system has achieved its objective as soon as the anomaly has been detected, our system simply proceeds as before but with less confidence in its predictions. This means that when, eventually, the deceptive agent does begin to approach her real goal, our system is still operating. In future work, therefore, there is scope to consider how confidence could be restored if, after a period of irrationality or erratic behaviour, the observed agent seems once again to be 'back on track'.

The self-modulating formula depends on a sequence of observations to assess an agent's degree of rationality. This is quite different from single-observation recognition, which minimises the number of observations required. Recall, however, that single-observation recognition does use the β parameter (a rate or 'heat' parameter, which we use to modulate the shape of the distribution). This means that, although the two processes cannot be unified (we cannot have self-modulating single-observation recognition, for example), we could use a rationality measure obtained during one event as the β value for single-observation recognition in another. Using this aproach, it should be possible to

determine the most likely destination of a 'highly rational' agent, on the basis of fewer than usual observations.

In relation to deception, several opportunities for further research were flagged in Section 4.4: full incorporation of magnitude into our model, development of more sophisticated strategies that exploit other aspects of Bell and Whaley's theory of deception and the potential for inverting our model of deceptive path-planning so that it can be applied to *truthful* path-planning (i.e., intended recognition). There is also scope to develop optimised algorithms to implement the strategies already proposed.

Another promising aspect of deceptive path-planning concerns exploitation of known (or suspected) psychological idiosyncrasies and biases when it is known that the observer is human. Recall that (on the negative side) we noted, in Section 4.3, that a 'pure' simulation strategy (which heads straight towards a bogus goal, then diverts towards the real goal) ceases to deceive a human observer almost the moment the diversion towards the real goal begins; whereas, for a computerised goal recognition system, deception—i.e., the probability of the bogus goal—may persist for some considerable distance. The flipside of this human tendency to jump quickly to conclusions was noted by Baker et al. (2011) in their work on Bayesian theory of mind. In humans, they observed that opinions formed early tend to dominate opinions that are formed late. We could take advantage of this computationally, for example, by introducing a discount factor (or value gain) for paths that 'simulate' strongly at the start. Other possibilities of a similar nature include the idea of incorporating rewards (or discounts) for consecutive moves in the same direction, based on the idea (untested) that continuity increases the perception of intent.

If we were to enrich the domain so that it became capable of dealing with more complete motion-planning and extended navigational scenarios, we could consider the impact of speed. For example, is fast, direct movement more persuasive of intentionality than slow, indirect movement, even if both movements are generally tending towards the same goal? There may be many other psychological factors that could be brought to bear on the deceptive path-planning problem. Cognitive science has already had considerable influence on goal recognition (e.g., in the work of Baker et al., 2011 and Vered et al., 2016) much of which may also apply to deception, particularly when treated—as it has been by us—as an inversion of the goal recognition problem.

Our approach to goal recognition and deceptive path-planning has involved specialising from task-planning to path-planning in the hope that, by reducing complexity, we might gain new insights. Perhaps the greatest potential for future work lies in following through on an attempt to generalise those insights back to task-planning.

Appendices

APPENDIX A

The Boltzmann Equation

The Boltzmann equation is one of the major equations used to describe the behaviour of thermodynamic systems but, drawing parallels between systems of particles in thermodynamic equilibrium and systems of neurons, it is also used by computer scientists to model neurons in simulated neural networks (Maren, 1989). The equation describes the probability of a particle being in a particular state α and is usually given as:

$$P_\alpha = \frac{1}{1 + e^{-\Delta E/T}}, \tag{A.1}$$

where ΔE is the energy difference between states and T is the temperature of the system. Thus, when ΔE is large, $e^{-\Delta E/T}$ is very small, owing to the negative exponential.

In his thesis, Ramirez (2012) arrives at an almost identical formulation to describe his probabilistic model of plan recognition:

$$P(O \mid G) = \frac{1}{1 + e^{-\beta\Delta(G)}}, \tag{A.2}$$

where O is the observations, G the set of possible goals and $\Delta(G) = optc^{\neg}(G, O) - optc(G, O)$, with $optc(\cdot)$ and $optc^{\neg}(\cdot)$ the optimal cost of a plan for goal via and 'not' via the observations, respectively. Similar to the temperature of the system, β is a rate parameter which 'modulates' the assumption that the observed agent is pursuing plans sensitive to the same cost function used by the observer: as β approaches zero, the distribution flattens out (Ramirez, 2012, p.63).

Observe that $\Delta(G)$ is a representation of the cost difference formula (similar to that given at (RG1), p.47) but *with terms reversed* though, for legibility, we reorganised the terms when articulating this as a template for cost difference formulas in Part I (at p.48). Observe also that the equation describes the probability of observations given the goal, whereas we require the probability of the goal given the observations. From Bayes' Rule, however:

$$P(G \mid O) = \alpha P(O \mid G) \cdot P(G), \tag{A.3}$$

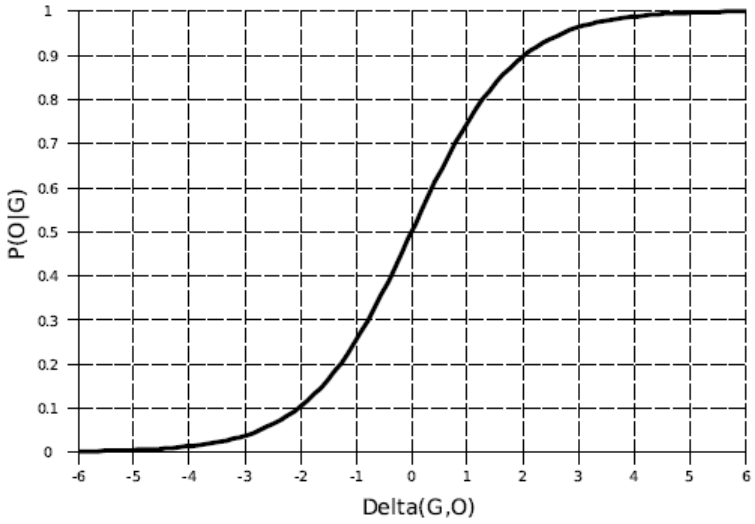


Figure A.1: Curve given by Equation (A.2) where $\Delta(G,O) = optc^{\neg}(G,O) - optc(G,O)$ and $\beta = 1$. Reproduced from Ramirez's thesis (2012, p.65).

where α is a normalising constant and $P(G)$ represents prior probabilities. Since we assume that $P(G)$ is given, Equation (A.2) returns the distribution we require.

An alternative formulation appears in the codebase referenced from Ramirez and Geffner's 2010 paper, as follows.

$$P_X(G \mid O) = \alpha \frac{e^{-\beta X}}{1 + e^{-\beta X}}, \tag{A.4}$$

where again α is a normalising constant and β a positive constant. Here, however, $X = costdif(G,O) = optc(G,O) - optc^{\neg}(G,O)$, not $optc^{\neg}(G,O) - optc(G,O)$, that is, the terms of the cost difference formula appear in the expected order.

In any event, the two formulations are provably equivalent. Here, let $A = optc^{\neg}(G,O)$ and $B = optc(G,O)$:

$$\begin{aligned}
(A.4) = \alpha \frac{e^{-\beta(B-A)}}{1 + e^{-\beta(B-A)}} &= \alpha \times \frac{1}{e^{\beta(B-A)}} \div (1 + \frac{1}{e^{\beta(B-A)}}) \\
&= \alpha \times \frac{1}{e^{\beta(B-A)}} \div \frac{e^{\beta(B-A)} + 1}{e^{\beta(B-A)}} \\
&= \alpha \times \frac{1}{e^{\beta(B-A)}} \times \frac{e^{\beta(B-A)}}{e^{\beta(B-A)} + 1} \\
&= \alpha \frac{1}{e^{\beta(B-A)} + 1} = \alpha \frac{1}{e^{\beta B - \beta A} + 1} \\
&= \alpha \frac{1}{1 + e^{-\beta(A-B)}} = (A.2).
\end{aligned}$$

Figure A.1 shows the curve generated by formula (A.2). The Boltzmann is a sigmoid equation, guaranteed never to exceed 1 or drop below zero. In the context of a probability distribution, this 'protection' is superfluous, however, since probabilities are anyway guaranteed (by the normalisation constant α) to fall within this range.

References

Alloway, T. P., McCallum, F., Alloway, R. G., & Hoicka, E. (2015). Liar, liar, working memory on fire: investigating the role of working memory in childhood verbal deception. *Journal of Experimental Child Psychology*, *137*, 30–38.

Arkin, R. C., Ulam, P., & Wagner, A. R. (2012). Moral decision making in autonomous systems: enforcement, moral emotions, dignity, trust, and deception. In *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)* (Vol. 100, pp. 571–589).

Avrahami-Zilberbrand, D., & Kaminka, G. A. (2007). Incorporating observer biases in keyhole plan recognition (efficiently!). In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (Vol. 7, pp. 944–949).

Avrahami-Zilberbrand, D., & Kaminka, G. A. (2014). Keyhole adversarial plan recognition for recognition of suspicious and anomalous behavior. In *AAAI Workshop on Plan, Activity, and Intent Recognition* (pp. 87–121).

Baker, C. L., Saxe, R. R., & Tenenbaum, J. B. (2009). Action understanding as inverse planning. *Cognition*, *113*(3), 329–349.

Baker, C. L., Saxe, R. R., & Tenenbaum, J. B. (2011). Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the Cognitive Science Society* (pp. 2469–2474).

Bast, H., Delling, D., Goldberg, A., Muller-Hannemann, M., Pajor, T., Sanders, P., ... Werneck, R. F. (2015). *Route planning in transportation networks* (Tech. Rep. No. MSR-TR-2014-4). Microsoft Corporation.

Bauer, M. (1995). A Dempster-Shafer approach to modeling agent preferences for plan recognition. *User Modeling and User-Adapted Interaction*, *5*(3-4), 317–348.

Bell, J. B. (2003). Toward a theory of deception. *International Journal of Intelligence and Counterintelligence*, *16*(2), 244–279.

Blaylock, N., & Allen, J. (2006). Fast hierarchical goal schema recognition. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (Vol. 21, pp. 796–801).

Bonchek-Dokow, E., & Kaminka, G. A. (2014). Towards computational models of intention detection and intention prediction. *Cognitive Systems Research*, *28*, 44–79.

Bond, J., Charles F., & Robinson, M. (1988). The evolution of deception. *Journal of Nonverbal Behavior*, *12*(4), 295–307.

Borenstein, J., & Arkin, R. (2016). Robotic nudges: the ethics of engineering a more socially just human being. *Science and Engineering Ethics*, *22*(1), 31–46.

Bowyer, J. B. (1982). *Cheating: Deception in war & magic, games & sports.* St Martin's Press.

Bratman, M. E., Israel, D., & Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, *4*(4), 349–355.

Bui, H. H. (2003). A general model for online probabilistic plan recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (Vol. 3, pp. 1309–1315).

Carberry, S. (1990). Incorporating default inferences into plan recognition. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 471–478).

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

Carson, T. L. (2010). *Lying and deception: Theory and practice.* Oxford University Press.

Castelfranchi, C. (2000). Artificial liars: Why computers will (necessarily) deceive us and each other. *Ethics and Information Technology*, *2*(2), 113–119.

Charniak, E., & Goldman, R. P. (1991). *Probabilistic abduction for plan recognition* (Tech. Rep. Nos. CS-91–12). Brown University. Department of Computer Science.

Cherkassky, B. V., Goldberg, A. V., & Radzik, T. (1996). Shortest paths algorithms: Theory and experimental evaluation. *Mathematical Programming*, *73*(2), 129–174.

Chisholm, R. M., & Feehan, T. D. (1977, March). The intent to deceive. *The Journal of Philosophy*, *74*(3), 143–159.

Coman, A., & Aha, D. W. (2018). AI rebel agents. *AI Magazine*, *39*(3), 16–27.

Dadouch, S. (2018). *Air strike warning app helps Syrians dodge death from the skies.* Retrieved from https://www.reuters.com/article/us-mideast-crisis-syria-warning/air-strike-warning-app-helps-syrians-dodge-death-from-the-skies-idUSKCN1LT2HV (Accessed: January 2019)

Demolombe, R., & Hamon, E. (2002). What does it mean that an agent is performing a typical procedure? A formal definition in the situation calculus. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 905–911).

Dennett, D. (1987). *The intentional stance* (First MIT Press Paperback ed.). Cambridge, Massachusetts: Massachusetts Institute of Technology.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271.

Dragan, A. D., Lee, K. C., & Srinivasa, S. S. (2013). Legibility and predictability of robot motion. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction* (pp. 301–308).

Elsaesser, C., & Stech, F. J. (2007). Detecting deception. In A. Kott & W. M. McEneaney (Eds.), *Adversarial reasoning: Computational approaches to reading the opponent's mind* (pp. 101–124). Chapman & Hall/CRC.

Escudero-Martin, Y., Rodriguez-Moreno, M. D., & Smith, D. E. (2015). A fast goal recognition technique based on interaction estimates. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 761–768).

Ettinger, D., & Jehiel, P. (2010). A theory of deception. *American Economic Journal: Microeconomics*, *2*(1), 1–20.

Ferguson, D., & Stentz, A. (2006). Using interpolation to improve path planning: The Field D* algorithm. *Journal of Field Robotics*, *23*(2), 79–101.

Firl, J., & Tran, Q. (2011). Probabilistic maneuver prediction in traffic scenarios. In *European Conference on Mobile Robots* (pp. 89–94).

Freedman, R. G., Fung, Y. R., Ganchin, R., & Zilberstein, S. (2018). Towards quicker probabilistic recognition with multiple goal heuristic search. In *AAAI Workshop on Plan, Activity, and Intent Recognition* (pp. 601–606).

Freedman, R. G., & Zilberstein, S. (2017). Integration of planning with recognition for responsive interaction using classical planners. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 4581–4588).

Freska, C. (1991). Qualitative spatial reasoning. *Cognitive and Linguistic Aspects of Geographic Space*, 361–372.

Gao, X., Firner, B., Sugrim, S., Kaiser-Pendergrast, V., Yang, Y., & Lindqvist, J. (2014). Elastic pathing: Your speed is enough to track you. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 975–986).

GCHQ. (2014). *The art of deception: training for a new generation of online covert operations.* Retrieved from https://theintercept.com/document/2014/02/24/art-deception-training-new-generation-online-covert-operations (Accessed: January 2019)

Geib, C. (2006). Plan recognition. *Adversarial Reasoning*, 77–95.

Geib, C., & Goldman, R. (2001). Probabilistic plan recognition for hostile agents. In *Florida Artificial Intelligence Research Society Conference* (pp. 580–584).

Geib, C., & Goldman, R. (2009). A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, *173*(11), 1101–1132.

Google. (2018). *Google duplex demo from google io 2018.* Retrieved from https://www.youtube.com/watch?v=bd1mEm2Fy08 (Accessed: September 2018)

Graf, R., Deusch, H., Seeliger, F., Fritzsche, M., & Dietmayer, K. (2014). A learning

concept for behavior prediction at intersections. In *IEEE Intelligent Vehicles Symposium* (pp. 939–945).

Halpern, J. J. (1998). Bonded rationality: The rationality of everyday decision making in a social context. *J. J. Halpern and R. N. Stern (Eds.), Debating Rationality: Nonrational Aspects of Organizational Decision-Making*, 219–238.

Harabor, D., & Grastien, A. (2012). The JPS pathfinding system. In *Symposium on Combinatorial Search (SOCS)* (pp. 207–208).

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

Hespanha, J. P. (2007). Application and value of deception. In A. Kott & W. M. McEneaney (Eds.), *Adversarial reasoning: Computational approaches to reading the opponent's mind* (pp. 145–165). Chapman & Hall/CRC.

Hespanha, J. P., Ateskan, Y. S., Kizilocak, H., et al. (2000). Deception in non-cooperative games with partial information. In *Proceedings of the DARPA-JFACC Symposium on Advances in Enterprise Control* (pp. 139–147).

Hintikka, J. (1962). *Knowledge and belief: An introduction to the logic of the two notions.* Cornell University Press.

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

Hong, J. (2001). Goal recognition through goal graph analysis. *Journal of Artificial Intelligence Research (JAIR)*, *15*, 1–30.

Jian, J.-Y., Matsuka, T., & Nickerson, J. V. (2006). Recognizing deception in trajectories. In *Proceedings of the Cognitive Science Society* (pp. 1563–1568).

Kahneman, D. (2011). *Thinking, fast and slow.* U.S.A.: Farrar, Straus and Giroux.

Kaminka, G. A., Vered, M., & Agmon, N. (2018). Plan recognition in continuous domains. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 6202–6210).

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

Keren, S., Gal, A., & Karpas, E. (2014). Goal recognition design. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)* (pp. 154–162).

Keren, S., Gal, A., & Karpas, E. (2015). Goal recognition design for non-optimal agents. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 3298–3304).

Keren, S., Gal, A., & Karpas, E. (2016). Privacy preserving plans in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 3170–3176).

Kooij, J. F. P., Schneider, N., Flohr, F., & Gavrila, D. M. (2014). Context-based pedestrian path prediction. In *European Conference on Computer Vision* (pp. 618–633).

Kott, A., & McEneaney, W. M. (Eds.). (2007). *Adversarial reasoning: Computational approaches to reading the opponent's mind*. Chapman & Hall/CRC.

Kulkarni, A., Klenk, M., Rane, S., & Soroush, H. (2018). Resource bounded secure goal obfuscation. In *AAAI Fall Symposium on Integrating Planning, Diagnosis and Causal Reasoning.*

Kulkarni, A., Srivastava, S., & Kambhampati, S. (2018). A unified framework for planning in adversarial and cooperative environments. In *ICAPS Workshop on Planning and Robotics.*

LaValle, S. M. (2006). *Planning algorithms*. Cambridge, U.K.: Cambridge University Press. (Available at http://planning.cs.uiuc.edu/; accessed: January 2019)

Le, D., & Plaku, E. (2018). Multi-robot motion planning with dynamics guided by multi-agent search. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 5314–5318).

Levine, T. R. (2014a). *Encyclopedia of deception*. SAGE Publications.

Levine, T. R. (2014b). Truth-default theory (TDT) a theory of human deception and deception detection. *Journal of Language and Social Psychology*, *33*(4), 378–392.

Liao, L., Patterson, D. J., Fox, D., & Kautz, H. A. (2007). Learning and inferring transportation routines. *Artificial Intelligence Journal (AIJ)*, *171*(5-6), 311–331.

MacNally, A. M., Lipovetzky, N., Ramirez, M., & Pearce, A. R. (2018). Action selection for transparent planning. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)* (pp. 1327–1335).

Mahon, J. E. (2008). *The definition of lying and deception.* Retrieved from http://stanford.library.usyd.edu.au/archives/sum2009/entries/lying-definition (Accessed: September 2018)

Mao, W., & Gratch, J. (2004). A utility-based approach to intention recognition. In *AAMAS Workshop on Agent Tracking: Modeling Other Agents from Observations* (Vol. 46, pp. 59–65).

Maren, A. J. (1989). *A tutorial on the Boltzmann distribution and energy minimization for neural network ensembles* (White Paper No. DOI: 10.13140/2.1.3044.7685). The University of Tennessee Space Institute.

Masters, P., & Sardina, S. (2017a). Cost-based goal recognition for path-planning. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)* (pp. 750–758).

Masters, P., & Sardina, S. (2017b). Deceptive path-planning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 4368–4375).

Masters, P., & Sardina, S. (2019a). Cost-based goal recognition in navigational domains.

Journal of Artificial Intelligence Research (JAIR), *64*, 197–242.

Masters, P., & Sardina, S. (2019b). Goal recognition for rational and irrational agents. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)* (pp. 440–448).

McCarthy, J. (1980). Circumscription - a form of nonmonotonic reasoning. *Artificial Intelligence*, *13*, 27–39.

McClean, J., Stull, C., Farrar, C., & Mascareñas, D. (2013). A preliminary cyber-physical security assessment of the robot operating system (ROS). *Unmanned Systems Technology XV*, *8741*, 1–8.

Millington, I., & Funge, J. (2009). *Artificial intelligence in games* (2nd ed.). Burlington, Massachusetts: Morgan Kaufmann.

Mirsky, R., & Gal, Y. (2016). SLIM: Semi-lazy inference mechanism for plan recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 394–400).

Murray, J. S. (1988). Disputation, deception, and dialectic: Plato on the true rhetoric ('Phaedrus' 261-266). *Philosophy & rhetoric*, *21*(4), 279–289.

Newell, A. (1982). The knowledge level. *Artificial intelligence*, *18*(1), 87–127.

O'Kane, J. M. (2009). On the value of ignorance: Balancing tracking and privacy using a two-bit sensor. In *Algorithmic Foundation of Robotics VIII* (pp. 235–249).

Patterson, D. J., Liao, L., Fox, D., & Kautz, H. (2003). Inferring high-level behavior from low-level sensors. In *UbiComp* (pp. 73–89).

Pattison, D., & Long, D. (2010). Domain independent goal recognition. In *Proceedings of the Starting AI Researchers' Symposium* (Vol. 222, pp. 238–250).

Pattison, D., & Long, D. (2013). Accurately determining intermediate and terminal plan states using Bayesian goal recognition. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 32–37).

Pereira, R. F., Oren, N., & Meneguzzi, F. (2017). Landmark-based heuristics for goal recognition. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 3622–3628).

Pita, J., Jain, M., Marecki, J., Ordóñez, F., Portway, C., Tambe, M., ... Kraus, S. (2008). Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)* (pp. 125–132).

Pohl, I. (1970). Heuristic search viewed as path finding in a graph. *Artificial Intelligence*, *1*(3-4), 193–204.

Pozanco, A., Yolanda, E., Fernández, S., & Borrajo, D. (2018). Counterplanning using goal recognition and landmarks. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 4808–4814).

Pynadath, D. V., & Marsella, S. C. (2005). Psychsim: Modeling theory of mind with

decision-theoretic agents. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (Vol. 5, pp. 1181–1186).

Raffe, W. L., Zambetta, F., & Li, X. (2012). A survey of procedural terrain generation techniques using evolutionary algorithms. In *IEEE Congress on Evolutionary Computation (CEC)* (pp. 1–8).

Ramirez, M. (2012). *Plan recognition as planning* (Unpublished doctoral dissertation). Univeristat Pompeu Fabra, Spain.

Ramirez, M., & Geffner, H. (2009). Plan recognition as planning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1778–1783).

Ramirez, M., & Geffner, H. (2010). Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 1121–1126).

Ramırez, M., & Geffner, H. (2011). Goal recognition over POMDPs: Inferring the intention of a POMDP agent. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 2009–2014).

Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. *Knowledge Representation*, *91*, 473–484.

Ren, Y., Tomko, M., Salim, F. D., Chan, J., Clarke, C., & Sanderson, M. (2017). A location-query-browse graph for contextual recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *30*(2), 204–218.

Root, P., De Mot, J., & Feron, E. (2005). Randomized path planning with deceptive strategies. In *Proceedings of the American Control Conference* (pp. 1551–1556).

Roy, P. C., Bouzouane, A., Giroux, S., & Bouchard, B. (2011). Possibilistic activity recognition in smart homes for cognitively impaired people. *Applied Artificial Intelligence*, *25*(10), 883–926.

Russell, S., & Norvig, P. (2013). *Artificial intelligence: A modern approach* (3rd ed.). New Jersey, USA: Prentice Hall.

Sartre, J.-P. (1956). *Being and nothingness*. New York: The Philosophical Library.

Sharif, M., Bhagavatula, S., Reiter, M., & Bauer, L. (2016). Accessorize to a crime: Real and stealthy attachs on state-of-the-art face reco. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security* (pp. 1528–1540).

Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., ... Meyer, G. (2012). PROTECT: A deployed game theoretic system to protect the ports of the United States. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)* (pp. 13–20).

Shim, J., & Arkin, R. C. (2012). Biologically-inspired deceptive behavior for a robot. *From Animals to Animats*, *12*, 401–411.

Shim, J., & Arkin, R. C. (2013). A taxonomy of robot deception and its benefits in HRI. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp.

2328–2335).

Short, E., Hart, J., Vu, M., & Scassellati, B. (2010). No fair!! an interaction with a cheating robot. In *IEEE International Conference on Human-Robot Interaction* (pp. 219–226).

Shvo, M., Sohrabi, S., & McIlraith, S. A. (2018). An AI planning-based approach to the multi-agent plan recognition problem. In *Advances in Artificial Intelligence: Proceedings of the Canadian Conference on Artificial Intelligence* (pp. 253–258).

Sohrabi, S., Riabov, A. V., & Udrea, O. (2016). Plan recognition as planning revisited. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 3258–3264).

Sreedharan, S., Chakraborti, T., & Kambhampati, S. (2017). Balancing explicability and explanation in human-aware planning. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (Vol. FS-17, pp. 61–68).

Sturtevant, N. R. (2012). Benchmarks for grid-based pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games*, *4*(2), 144–148.

Sukthankar, G., Geib, C., Bui, H. H., Pynadath, D., & Goldman, R. P. (2014). *Plan, activity, and intent recognition: Theory and practice.* Newnes.

Sukthankar, G., & Sycara, K. (2005). A cost minimization approach to human behavior recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1067–1074).

Tambe, M. (2015). *Security games: Key algorithmic principles, deployed applications and research challenges.* Retrieved from https://simons.berkeley.edu/talks/milind-tambe-2015-11-18 (Accessed: January 2019)

Tambe, M., & Rosenbloom, P. S. (1996). Event tracking in a dynamic multiagent environment. *Computational Intelligence*, *12*(3), 499–522.

Tastan, B., & Sukthankar, G. (2011). Leveraging human behavior models to predict paths in indoor environments. *Pervasive and Mobile Computing*, *7*(3), 319–330.

The Future of Life Institute. (2015). *The 26518 open letter signatories.* Retrieved from https://futureoflife.org/awos-signatories/ (Accessed: October 2018)

Tovey, C., & Koenig, S. (2000). Gridworlds as testbeds for planning with incomplete information. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 819–824).

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *59*(236), 433–460.

van Ditmarsch, H., van der Hoek, W., & Kooi, B. P. (2007). *Dynamic epistemic logic.* Springer Science & Business Media.

Vered, M., & Kaminka, G. A. (2017). Heuristic online goal recognition in continuous domains. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 4447–4454).

Vered, M., Kaminka, G. A., & Biham, S. (2016). Online goal recognition through mirroring: Humans and agents. In *Conference on Advances in Cognitive Systems.*

Vered, M., Pereira, R., Magnaguagno, M., Kaminka, G., & Meneguzzi, F. (2018). Towards online goal recognition combining goal miroring and landmarks. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)* (pp. 2112–2114).

Whaley, B. (1982). Toward a general theory of deception. *The Journal of Strategic Studies*, *5*(1), 178–192.

Wiest, J., Höffken, M., Kreßel, U., & Dietmayer, K. (2012). Probabilistic trajectory prediction with gaussian mixture models. In *IEEE Intelligent Vehicles Symposium* (pp. 141–146).