Journal of Artificial Intelligence Research 64 (2019) Submitted 06/18; published 02/19

Cost-Based Goal Recognition in Navigational Domains

Peta Masters PETA.MASTERS@RMIT.EDU.AU
Sebastian Sardina SEBASTIAN.SARDINA@RMIT.EDU.AU
RMIT University, 124 La Trobe St
Melbourne, Vic 3000, Australia

Abstract

Goal recognition is the problem of determining an agent's intent by observing her behaviour. Contemporary solutions for general task-planning relate the probability of a goal to the cost of reaching it. We adapt this approach to goal recognition in the strict context of path-planning. We show (1) that a simpler formula provides an identical result to current state-of-the-art in less than half the time under all but one set of conditions. Further, we prove (2) that the probability distribution based on this technique is independent of an agent's past behaviour and present a revised formula that achieves goal recognition by reference to the agent's starting point and current location only. Building on this, we demonstrate (3) that a Radius of Maximum Probability (i.e., the distance from a goal within which that goal is guaranteed to be the most probable) can be calculated from relative cost-distances between the candidate goals and a start location, without needing to calculate any actual probabilities. In this extended version of earlier work, we generalise our framework to the continuous domain and discuss our results, including the conditions under which our findings can be generalised back to goal recognition in general task-planning.

1. Introduction

Consider an airport surveillance system. An agent of interest is observed entering the terminal. The problem is to assess whether they are making for some particular boarding gate; if so, the system will raise a flag to trigger their interception. Now, using state-of-the-art goal recognition, the accepted solution would be to track the agent's movements throughout the airport accumulating as many observations as possible (which might include watching them criss-cross the terminal many times as they buy a paper, use the bathroom, get coffee, etc.) repeatedly calculating and recalculating the probabilities of each gate they 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 paper, however, we develop solutions such that, given the agent's entry point into the domain, we only need to know their current location to generate a probability distribution which ranks goals in the same order as if we had tracked them all over the airport. A human operator might use this model to make spot-checks (e.g., on an agent seen acting suspiciously) or to pre-calculate probabilities at any point even before the agent enters the terminal. Alternatively, by extending the model—with a single calculation and without generating any actual probabilities—we can predetermine a radius within which the target gate is guaranteed to have become the most probable. Thus, 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 they are spotted here, the flag should be raised.

Goal recognition (GR)[1] is the problem of determining an agent's intent by observing its behaviour. In this paper, we investigate GR in navigation: the problem of determining where an agent may be travelling, given a set of potential destinations and a sequence of one or more of the locations where it has already been. It is a flourishing field of research (Sukthankar, Geib, Bui, Pynadath, & Goldman, 2014) with numerous navigation- and movement-related applications, for example in support of adversarial reasoning for games and the military (Kott & McEneaney, 2006; Sukthankar & Sycara, 2005). Moreover, with an ageing population and the spectre of autonomous vehicles on the horizon, there is increasing emphasis on research into movement within smart homes (Roy, Bouzouane, Giroux, & Bouchard, 2011), wheelchair navigation (Demeester, Nuttin, Vanhooydonck, & Van Brussel, 2003), trajectory prediction (Lefèvre, Vasquez, & Laugier, 2014), manoeuvre prediction (Firl & Tran, 2011) and so on.

Traditionally, GR has involved matching a sequence of observations to a plan in a *plan library*, the winning plan being the one that "best" matches the observations. Recent developments, however, dispense with the overhead of a plan library and instead—based on the assumption that the observed agent is behaving rationally—take a cost-based approach, using classical planning technology to generate plans as-needed over a model of the domain. Ramirez and Geffner (2009, 2010) were among the first to propose this "plan recognition as planning" approach and introduced the notion of *cost difference* to determine which plan is "best". Under this model, Ramirez and Geffner (2009) presented an initial solution that identified a crisp subset of "most likely goals", consisting solely of those goals for which observations conform to the optimal plan (i.e., the cost difference between the optimal plan and the observed plan is zero). The inability of the system to accommodate even marginally suboptimal plans represented a significant limitation, however, and their 2010 paper presented a more flexible model which generates a *probability distribution* across the set of possible goals, based on the cost difference between the cheapest available plan that conforms to observations and the cheapest plan that does not.

Notwithstanding its scholarship and convenience, Ramirez and Geffner's probabilistic account (2010) is computationally expensive: it requires two calls to the planner for every goal and the use of a planner capable of handling negative conditions. These issues make it particularly ill-suited for online goal recognition (i.e., where observations are provided incrementally in real-time) and several authors have suggested modifications to improve its efficiency (Escudero-Martin, Rodriguez-Moreno, & Smith, 2015; Pereira, Oren, & Meneguzzi, 2017; Vered, Kaminka, & Biham, 2016).

In this article, we apply Ramirez and Geffner's 2009 notion of cost difference to their 2010 framework and show that, in the context of path-planning, their cost-based approach to GR can be exploited to derive solutions that are much more economical. We start by demonstrating formally that—even in the context of task-planning—their simpler 2009 formula (previously used by Escudero-Martin et al. (2015) but without theoretical guarantees) yields an almost identical result to the more complex 2010 formula with less computational effort. We set out the one case where the two accounts do differ and discuss the implications. More surprisingly, we then show (for path-planning) that a probability distribution which ranks candidate goals in the same order can be obtained by reference to the agent's starting point and current location only (i.e., without reference to its observation history). This "single-observation formula"[2] provides an economical means of

1. Goal recognition is properly a subproblem of plan recognition, although both terms are used in the literature. We will use the terms interchangeably within this text but note that goal (not plan) recognition is our primary focus.
2. We previously used the term "observation-free" (Masters & Sardina, 2017a). However, the method does depend on observing one location (usually the agent's *current* location).

performing online GR. Moreover, it can be advantageous in any practical application because the fewer the observations, the less infrastructure required to retrieve and process them. Furthermore, being independent of the 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—this formula makes it possible for probability distributions to be pre-computed offline and retrieved in constant time. We have shown elsewhere (Masters & Sardina, 2017b) that we can build on these results to calculate a radius around goal within which that goal becomes the most probable. Here, we generalise our framework to continuous domains and redefine this concept of a "radius of maximum probability" (RMP) in the more adequate continuous setting, enabling us to examine its implications in a real-world context.

We have restricted our focus to 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). In maintaining this focus, we have assumed a deterministic, static environment. Observations are complete (i.e., capture the full state) and are treated as "observable fluents" (Sohrabi, Riabov, & Udrea, 2016), in our case, locations. Given these assumptions, it may be that our work is applicable in other areas of planning; such usage is beyond the scope of this paper, however. We take the view that (a) core navigational domains in themselves present an abundance of worthwhile applications for goal recognition; and (b) by focusing on the main properties first, we can aim to understand the key issues and subtleties of the problem, leaving elaborations on the core to be studied later.

The remainder of this article is organised as follows. First, we position our contribution in the context of related work. In section 3, we set out our formalism for "goal recognition by path-planning" in discrete and continuous domains. In section 4, we present an efficient new method of goal recognition: first we prove the cases where the simpler (and faster) 2009 formula is equivalent to Ramirez and Geffner's 2010 account (and formally identify the one case where it is not); then we demonstrate our "single-observation" formula, which ranks candidate goals in the same order as the simpler formula but without reference to the observation sequence. In section 5, we present our third contribution: a formula to calculate the RMP (that radius around a goal within which it is guaranteed to be the most probable). In section 6, we provide empirical support for our theoretical conclusions using the well-known Moving-AI benchmarks (Sturtevant, 2012) and we close the paper with discussion, including appraisal of our model's applicability to general task-planning.

2. Related Work

In this section, we examine the body of work from which our contribution has emerged, then focus on those aspects of Ramirez and Geffner's 2009 and 2010 papers which are central to our approach.

As a prelude, we remind the reader that the path-planning domain with which we are concerned is one in which a map - whether it represents the real-world, a game environment or other kind of search space - is treated either as a continuous Euclidean world with infinite points (LaValle, 2006) or abstracted into a graph (or grid). The *task* of path-planning consists of finding a route—typically, the optimal (i.e., shortest or minimum cost) route— from a starting location to a goal (Harabor & Grastien, 2012; Hart, Nilsson, & Raphael, 1968). The standard approach to finding an optimal path in a *discrete* domain depends on Djikstra's algorithm (Dijkstra, 1959)—still widely used as the basis for route-planning algorithms across road networks (Bast et al., 2015)—or A* (Hart et al., 1968), with which path-finding in games is said to be "synonymous" (Millington & Funge, 2009, pg.215).

Even in *continuous* domains, Djikstra and its variants are commonly used because, practically, the space must usually be discretised to a grid or a graph to make the problem tractable (LaValle, 2006).

Having clarified the domain of interest, we look at the relevant literature on goal recognition.

2.1 Goal Recognition

Goal recognition 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 include adversarial reasoning for games and the military (Kott & McEneaney, 2006), monitoring the movement of residents in smart homes for the cognitively impaired (Roy et al., 2011) and human-machine interaction (Lesh, Rich, & Sidner, 1999). With the advent of autonomous robotic vehicles, such uses have become increasingly important as an adjunct to trajectory prediction (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).

Formalisation of the plan recognition problem is widely attributed to Kautz and Allen (1986). 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 (Bui, 2003; Mirsky & Gal, 2016). See the recent book by Sukthankar et al. (2014) for a comprehensive survey of contemporary approaches.

Ambiguity is a major obstacle to PAIR and several authors have used cost as a means of disambiguation. Sukthankar and Sycara (2005) disambiguate in favour of whichever goal can be accessed at least cost 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 expected value. An alternative technique is to make a "worst case assumption" whereby, rather than considering the *preferences of the observed* agent, the system selects whichever goal will have the greatest (i.e., worst) impact on the *expected cost to the observer* (Avrahami-Zilberbrand & Kaminka, 2007; Tambe & Rosenbloom, 1995).

The traditional—and still prevalent—approach to goal recognition 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). The difficulty of acquiring and/or hand-coding the plans, however, has made 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).

2.1.1 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, independently arrived at by others (Baker, Saxe, & Tenenbaum, 2009; Pattison & Long, 2013), 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. We discuss these papers more fully in Section 2.2.

In most applications, goal recognition tasks must be performed "online" (Blaylock & Allen, 2006). That is, the objective is to identify the 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. 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). Kaminka, Vered, and Agmon (2018) present a unified model that covers both discrete and continuous environments. The authors point out that the continuous environment has demonstrable advantages when dealing with goal recognition that corresponds to real-world path-planning and, in formulating cost-based goal recognition for the continuous domain, we have built on insights from Vered et al.'s 2016 paper, which defines observed points and trajectories as a function of the time during which those observations were made.

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-like reasoning. Thus although, like us, they build on the formula at the centre of Ramirez and Geffner's non-probabilistic model (2009), which (effectively) subtracts the optimal cost of a plan from the optimal cost *given the observations already seen*, they use an alternative formula to derive the probability distribution across goals which takes 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. In Section 4, we demonstrate with our single-observation formula that—using Ramirez and Geffner's probability distribution formula instead of Vered and Kaminka's ratio-based alternative—no incremental processing is necessary: the cost of the path-prefix need never be calculated or factored in at all.

Ramirez and Geffner (2010) make no allowance for the possibility that observations may be unreliable and Sohrabi et al. (2016) extend their work by introducing weights to accommodate noisy and missing observations. In the same paper, the authors 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. Sohrabi et al. also point out that, although a plan in classical task-planning is given as a sequence of *actions*, in practise, the actions themselves are rarely visible. They suggest, therefore, that observations may be better understood as the *effects* of actions. This approach, whereby each individual observation is treated as an "observable fluent" (not an action, and not a state either) 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); and we adopt a similar approach, conflating a sequence of assumed actions, $go(x_1), go(x_2), ..$, to their associated locations, $x_1, x_2, ..$ and so on.

Freedman and Zilberstein (2017) extend Ramirez and Geffner's framework into the realm of human machine interaction. Their work uses the output of Ramirez and Geffner's plan recognition framework as input to a planning problem. Effectively this provides an agent with seeming "foresight" as to the intentions of the agent it is observing, which can then be used either to help that agent or hinder it. When assisting a colleague or blocking an adversary, it is useful to identify an optimum point or moment for intervention. This clearly has a relationship with landmarks (discussed in Section 2.1.2 below), particularly in the context of counterplanning (Pozanco, Yolanda, Fernández, & Borrajo, 2018). It also has some overlap with our work on the radius of maximum probability (presented in Section 5), which can be used to recognise, not a point, but a "cost-boundary" that must be crossed before the observed agent can reach its goal.

Freedman, Fung, Ganchin, and Zilberstein (2018) demonstrate a further extension to Ramirez and Geffner's framework, which saves time by ranking multiple goals at once, rather than carry out a computation for each goal separately. As the authors point out, these many and various extensions share a common objective in attempting to achieve goal recognition *more quickly*.

2.1.2 "DOMAIN THEORY" WITHOUT PLANNING

Classical planning can be computationally expensive in itself and it is not the only way to avoid the cost involved with plan libraries. Pereira et al. (2017) extend Hoffmann, Porteous, and Sebastia's work (2004) on landmarks (actions that *must* occur for a goal to be achieved) to develop a non-probabilistic model that follows a similar approach to that of Sohrabi et al. (2016) in that it treats landmarks as *facts* that must be satisfied rather than actions that must occur. Briefly, a set 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.

Escudero-Martin et al.'s probabilistic solution (2015) involves creating a plan graph, based on the problem domain, from which to generate cost estimates for each goal. They then prune the graph based on observed actions and generate a second set of cost estimates, 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 the "most likely" goals can be considered further under some other process.

Vered et al. (2018) combine goal mirroring with the use of landmarks 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 the context of 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 (absence of obstacles) between enclosed points and the goal. The end result is superficially similar to our radius of maximum probability (RMP) though calculated very differently and without the same theoretical guarantees. Notwithstanding the differences, some of the ways that landmarks are used—e.g., to prune goals if a landmark is exited and reinstate them if it is re-entered—suggest practical applications for the RMP.

2.1.3 GOAL RECOGNITION DESIGN

Keren, Gal, and Karpas (2014) introduced a new field of study in relation to *offline* goal recognition. "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 an optimal plan before its goal can be uniquely identified. Briefly, a shared optimal path prefix is identified using a compilation whereby two agents, in 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. In their 2015 paper, the authors extend the model to also consider *sub*optimal plans by introducing a budgetary constraint (to minimise the total number of plans that need to be considered). In appropriate domains (e.g., real-world path-planning), our RMP calculation might be used to abbreviate calculations involved in determining worst case distinctiveness. Instead of working from the starting location, as Keren does, our model provides a computationally economical means (with no dependence on classical planning) of finding the precise cost-distance *from* goal at which "distinctiveness" is achieved.

Keren et al.'s approach implicitly acknowledges a relationship between goal recognition and the domain itself. Our work confirms this relationship, demonstrating that the probability of an agent's destination is domain- rather than observation-dependent. One offline application of our single-observation cost difference formula is to create a probabilistic heatmap of the domain. This construct could be used as an extension to goal recognition design, adding a quantitative element and a means of accounting for suboptimality without resorting to the use of budgetary constraints.

2.1.4 REAL-WORLD NAVIGATION

In a real-world navigation scenario, we could take many additional features into account beyond the "core" account which is the focus of our work. Several of these have been explored by other researchers in this field. Vered and Kaminka (2017), for example, track an agent's trajectory from observation to observation in order to detect changes of direction. If the agent turns 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. 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 its goal. The only constraints are that the agent should be operating in a domain (such as the road network) where additional available information includes its likely start location (e.g., home address), maximum legal speeds along each path segment and the location of intersections (where it may be necessary to stop or slow down). 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 their mode of transport (Patterson, Liao, Fox, & Kautz, 2003). Furthermore, setting aside physical considerations (such as speed and direction), an agent's plans can be predicted, even before they act, from their browsing habits. Indeed, researchers have demonstrated that an agent's information needs in the virtual network can be mapped to their purchasing needs in a real-world shopping centre (Ren et al., 2017).

Our framework does not need to be seen as an alternative to these approaches but as orthogonal to them; together they can provide a richer model.

2.2 Foundation to our Approach

We now offer 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".

Ramirez and Geffner (2009) define the goal recognition problem as one of "planning in reverse" (p.1778). The input is a STRIPS-style task-planning domain (of fluents and actions), a fully observable initial state, a set of *possible* goal states and a sequence of observed actions. The solution is a set of goals such that, for each goal, there is an optimal plan that embeds the observations. 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 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 can be said to be dominated by the highest of those probabilities (an approximation without which, we infer, the problem could become intractable). The first assumption is essential to their 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). Based on these assumptions, Ramirez and Geffner arrive at a probability distribution—concretely, a Boltzmann distribution that takes cost difference as the temperature of the system—with the following important property: *the lower the cost difference, the higher the probability*.

These two seminal papers moved the focus 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 the cost difference formula, 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 2010 model, 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 a 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 4), 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 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 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 negative reasoning is to substitute, as we do, the more straightforward 2009 formulation based on the cost of an optimal path. We note that Ramirez and Geffner explicitly reject this simpler formulation (Ramirez & Geffner, 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). Nevertheless, it is the approach taken by Escudero-Martin et al. (2015), who justify the substitution by arguing—as we do—that both terms will return the same result in the majority of situations. Escudero-Martin et al. stop short of presenting a formal proof of equivalence, which we do supply; neither do they formally define the special case where results returned are different, nor 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" (i.e., "for most problems" the special case cannot arise).

We conclude this section by reminding the reader of the most important difference between the STRIPS-like problem domain, which is the setting for Ramirez and Geffner's 2010 framework, and that used in our account: whereas they treat observations as actions, we follow a similar approach to Sohrabi et al. (2016) and treat them as locations. Although this transformation has been used elsewhere in a motion-planning context (Vered et al., 2016), it amounts to a considerable simplification which, in a core navigational domain such as ours, has the effect of making every observation a *fully observable state*. We discuss the implications of this simplification more fully in Section 7. For now, note that this approach is nevertheless broadly consistent with the original model, insofar as observations and plans are directly comparable: a partial (i.e., potentially unconnected) sequence of observed nodes (in the discrete domain) or observed points and trajectories (in the continuous domain) can be matched to a connected path just as a partial sequence of observed actions in a STRIPS-like GR problem can be matched to a consecutive sequence of actions in a plan.

3. Goal Recognition as Path-Planning

In this section, we set out our formalism for goal recognition in a path-planning context, first for the discrete then the continuous domain.

Path-planning is the problem of finding a path from an initial location to a final destination in some map or "model" of the world, often expressed in the discrete domain as a graph (Hart et al., 1968) or (in the special case) a grid (Harabor & Grastien, 2012) and in the continuous domain as Euclidean space (LaValle, 2006). Goal recognition *as path-planning* is the problem of identifying an agent's destination by generating and comparing the possible paths it may be taking to get there over a map or model of the underlying terrain. We model that terrain in terms of core navigational domains, which define location and cost/distance but exclude richer notions such as velocity, acceleration, heading, fuel consumption, localisation, and so on. In maintaining this focus, we assume a deterministic, static environment within which states are fully observable and paths are always sequential (i.e., subpaths cannot be arbitrarily reversed or traversed simultaneously).

Recall from Section 2 that there are significant differences between the STRIPS-like classical planning environment of Ramirez and Geffner's 2010 framework and the core path-planning domain with which we are concerned: paths are described in terms of states, not actions; and observations are fully observed states. There may be other domains where these constraints apply but they are not considered here. We discuss the ramifications of our assumptions (and provide examples of navigational domains where our framework may not apply) in Section 7.

3.1 GR as Path-Planning in the Discrete Domain

Here, we import Ramirez and Geffner's task-plan recognition framework (R&G) into a discrete navigational context. Whereas their framework operates in a STRIPS-style domain of fluents and actions, we express the underlying map or model (the "domain theory") as a graph with costed edges.

Definition 1 *A **discrete path-planning domain** is a triple $\mathcal{D} = \langle N, E, c \rangle$ 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 $\mathcal{D}$ is a sequence of node locations (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 the domain is denoted by Π, and the set of all paths π starting at $\pi^0 = n_1$ and ending at $\pi^{|\pi|} = n_2$ is denoted by $\Pi(n_1, n_2)$.

A path-planning problem instance includes the underlying domain, start location and goal.

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

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

As expected, a solution to a path-planning problem is a path in the corresponding domain $\mathcal{D}$ from the start location s to the goal location g. Technically, a ***solution path*** π is a path π such that $\pi^0 = s$ and $\pi^{|\pi|} = g$; an ***optimal path*** is a solution path with the lowest cost among all solution paths; and $\Pi^*(s, g)$ denotes the ***set of all optimal solution paths***.

In our work, it will be convenient to specify ***waypoints***: nodes that must be visited. A solution path *via waypoints* embeds those waypoints in such a way as to preserve their order. That is, given a path π and a sequence of waypoints $W = w_0, w_1, \ldots, w_k$, where $w_i \in N$, we say that π ***embeds waypoints*** W, if there exists a monotonic function $f : \{0, \ldots, k\} \mapsto \{0, \ldots, |\pi|\}$ mapping waypoint indices into path indices such that $\pi^{f(i)} = w_i$. As with paths, we use W^i to denote the i-th waypoint w_i in W, and $|W|$ is the length of W. The ***optimal cost*** of a path from n_i to n_j ***via waypoints*** W—that is, the cheapest path possible that embeds the waypoints—is denoted by $optc(n_i, W, n_j)$. When $W = \epsilon$ (i.e., no waypoints), we just write $optc(n_i, n_j)$; and we write $optc(W)$ as a shorthand for $optc(w_0, W, w_k)$, that is, the optimal cost through the waypoints themselves. We generalise sets $\Pi(s, g)$ and $\Pi^*(s, g)$ to those embedding waypoints W as $\Pi(s, W, g)$ and $\Pi^*(s, W, g)$, respectively (see that $\Pi(s, g) = \Pi(s, \epsilon, g)$ and $\Pi^*(s, g) = \Pi^*(s, \epsilon, g)$).

With the basic framework in place, we now formulate the goal recognition problem itself. In goal recognition for task-planning, we seek to determine an agent's intent from observation of one or more of its actions. In path-planning, we seek to determine its destination from one or more of the locations it has already visited.

Definition 3 *A* ***goal recognition problem for path-planning in the discrete domain (GR-D)*** *is a tuple* $\mathcal{R} = \langle \mathcal{D}, \mathcal{G}, s, O, Prob \rangle$*, where:*

- $\mathcal{D} = \langle N, E, c \rangle$ *is a path-planning domain;*
- $\mathcal{G} \subseteq N$ *is the set of possible goal locations;*
- $s \in N$ *is the start location;*
- $O = o_1, \ldots, o_k$*, where* $k \geq 0$ *and* $o_i \in N$ *for all* $i \in \{1, \ldots, k\}$*, is a sequence of observations that is feasible, that is,* $optc((s, o_1, \ldots, o_k)) \neq \infty$*, and such that* $o_1 \neq s$*; and*
- *Prob represents the prior probabilities of the goals.*

Note that consecutive nodes in an observation sequence O may or may not be connected, as the observations may be scattered and incomplete (i.e., some visited locations may not have been seen). Note also that since observations are nodes (just like steps in a path or the start location in $\mathcal{R}$), we could omit specification of s and, instead, use the first observation o_1. Indeed, other authors have taken this approach (Vered et al., 2016). Our design decision, however, has been to treat 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 s as part of the problem definition means of course that, where a domain has multiple possible start locations (door1, door2, etc.), use of a different door implies construction of a different problem.

The solution to a GR-D problem $\mathcal{R}$ is a posterior probability distribution $P(\mathcal{G} \mid O, s)$ over the set of possible goals, given the start location and the sequence of observations. Intuitively, under the principle of rationality, the intended meaning is that P should prefer those goals for which the optimal paths "best satisfy" the observations. As a baseline, we obtain the probability distribution via Ramirez and Geffner's 2010 framework: by comparing, for each goal, the cost difference between the optimal cost of a solution path does *not* embed observations O (i.e., because it avoids one or more of them or attains them out of order) from the optimal cost of a path that *does* embed them (i.e., by treating them as waypoints that must be visited). Formally, cost difference is the formula $\mathit{costdif}_{RG} : N \times N \times N^* \mapsto \mathbb{R}$:

$$\mathit{costdif}_{RG}(s, g, O) = \mathit{optc}(s, O, g) - \mathit{optc}^{\neg}(s, O, g), \tag{RG1}$$

where $\mathit{optc}^{\neg}(s, O, g)$ denotes the optimal cost of navigating from location s to location g without embedding the observed waypoints, that is:

$$\mathit{optc}^{\neg}(s, O, g) = \min_{\pi \in \Pi(s,g) \setminus \Pi(s,O,g)} \mathit{cost}(\pi).$$

Notice that if $\Pi(s, g) \setminus \Pi(s, O, g) = \emptyset$ (because *all* paths from s to g embed the observations O) then $\mathit{optc}^{\neg}(s, O, g) = \infty$ and $\mathit{costdif}_{RG}(s, g, O)$ is undefined.

Finally, we generate a solution to the GR-D problem using the following template (Ramirez & Geffner, 2010), which has the important property that *the lower the cost difference for a goal, the higher its probability*:[3]

$$P_f(g \mid O, s) = \alpha \frac{e^{-\beta\, \mathit{costdif}_f}}{(1 + e^{-\beta\, \mathit{costdif}_f})}, \tag{RG2}$$

where f identifies the particular cost difference formula in use, α is a normalising constant, and β a positive constant.[4] So in words, substituting RG for f in (RG2), $P_{RG}(g \mid O, s)$ uses $\mathit{costdif}_{RG}(s, g, O)$ to determine the probability that the observed agent is travelling to goal $g \in \mathcal{G}$, relative to GR-D problem $\mathcal{R}$, when the observation sequence is O.

3.2 GR as Path-Planning in the Continuous Domain

In Section 5, we will wish to consider path-planning in the continuous domain, where movement from one location to another necessarily involves traversal through an infinitely divisible sequence of points in between. Therefore, we now transpose the GR problem from a graph-based setting to a continuous two-dimensional plane (consistent with a traditional map or groundplan) or three-dimensional setting (consistent with our real-world experience of navigation). In place of nodes

3. This formulation appears in the codebase referenced by Ramirez and Geffner (2010) and is provably equivalent to the Boltzmann distribution in Ramirez's dissertation (2012). Though omitted for legibility, we assume that values may be multiplied by priors before normalisation.

4. 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). For technical convenience, we abuse notation and take $e^{-\beta\, \mathit{costdif}_f} = 0$, whenever $\mathit{costdif}_f = \infty$.

and edges, there is now an *infinite* number of points capable of becoming path-connected whenever reachable from one another through a connected region of traversable space.

This generalised account occurs in the context of *metric* spaces, which are (standard) topological spaces that define connectedness but also distance (LaValle, 2006).[5] 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).

With that in hand, let us restate Definition 1 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.[6] Without loss of generality, we assume 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).[7] 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 will wish to calculate shortest paths (and potentially make contact with the edges of obstacles), we define X_{free} as the *closure* of $\{X \setminus X_{obst}\}$.[8]

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 a sequence of such points, as in discrete domains. Hence, a ***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

5. 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.

6. If the agent were not embodied, either cost difference formula (RG3) loses meaning (as **?**, 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 practical application, an agent is embodied and capable of reaching its destination. Therefore, we follow LaValle (2006).

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

8. "Closed" and "open" sets are complementary notions: sets, resepectively, that do or do not include their boundary points. "Closure" of an open set is that 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).

is the path in space. So 0 is the first point, 1 the last and 0.5 the point half way along. (Note that 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.)

Hence, $\pi(0)$ 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 $i \in [0, 1]$.

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:

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

Notice that $\pi_{[t_1,t_2]}$ is in itself a path (i.e., its domain is $[0, 1]$). Finally, we will use $\Pi(\mathcal{D}_c)$, or simply Π, to denote the set of *all* paths in a domain $\mathcal{D}_c$.

To define the ***length of a path***, we see the path as partitioned into segments of equal length, 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 follows:

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

A path π is a ***solution path*** for $\mathcal{P}_c$ if $\pi(0) = x_s$ and $\pi(1) = x_g$. We use $\Pi(\mathcal{P}_c)$ 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, also 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 paper, we adopt the simple and often applicable model of cost as distance, that is, $\mathit{cost}(\pi) = L(\pi)$.[9]

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:

$$\pi^*(x_s, x_g) \in \underset{\{\pi \mid \pi\in\Pi, \pi(0)=x_s, \pi(1)=x_g\}}{\operatorname{argmin}} L(\pi).$$

Note that, 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).

There is one further difference with respect to our previous discrete account in the treatment of observations. Here, we follow Vered et al. (2016)[10] in making each observed point (or trajectory) a function of the time interval during which it was observed. Formally, an ***observation model*** for a continuous domain $\mathcal{D}_c$ is a pair $\mathcal{O} = \langle T_o, o \rangle$ where:

9. Alternative cost models tend to be either: *(i)* linear to distance anyway, (e.g., the sum of distances traversed through different terrain types such as ground, water, swamp, etc.); *(ii)* necessitate discretising the domain back to a graph or grid (e.g., road networks for SatNav/GPS or sampled spaces for Rapidly-exploring Random Trees). In more sophisticated accounts, such as motion-plans 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).

10. We have modified the notation to more completely express the intervals represented in T_o.

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 $t_2 < T$; 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 the time intervals during which the agent has been observed and $o([t_1, t_2])$ yields a *path* that represents the *continuous observation during the* $[t_1, t_2] \in T_o$. Since the domain of a path (function) is $[0, 1]$, $o([t_1, t_2])(0)$ denotes the first observed location (in the observed interval), whereas $o([t_1, t_2])(1)$ stands for the last observation (in the interval).

Using this observation account, we identify the ***set of all paths that embed the observations*** as the set of paths $\Pi(\mathcal{O})$ such that $\pi \in \Pi(\mathcal{O})$ *iff* there exists a mapping $\sigma : T_o \mapsto [0, 1] \times [0, 1]$ such that: *(i)* $o([t_1, t_2]) = \pi_{\sigma([t_1,t_2])}$; and *(ii)* for all $\vec{t}, \vec{t'} \in T_o$, $\vec{t} < \vec{t'}$ iff $\sigma(\vec{t}) < \sigma(\vec{t'})$. In words, every path in $\Pi(\mathcal{O})$ includes as subpaths all observations that occurred during the time interval T in the order that they were observed.

With this set at hand, we can now precisely state the optimal cost *from point* x_1 *to point* x_2 that embeds the observations $\mathcal{O}$ as follows:[11]

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

Similarly, the optimal cost *not* embedding (all) the observations can be defined as follows:

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

We now have all the technical machinery to turn our attention to the goal recognition problem itself and, most importantly, to its solution concept.

Definition 6 *A* ***goal recognition problem for path-planning in the continuous domain (GR-C)*** *is a tuple* $\mathcal{R}_c = \langle \mathcal{D}_c, A, \mathcal{G}, x_s, T, \mathcal{O}, \textit{Prob} \rangle$*, where:*

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

11. As standard, we assume the minimum of an empty set/range (here, when no path exists between the points) is ∞.

Observe that given an interval $[t_1, t_2] \in T_o$ in which the agent has been observed, $o([t_1, t_2])$ yields a *path* that represents the *continuous observation during the whole interval*. Since the domain of a path (function) is $[0, 1]$, $o([t_1, t_2])(0)$ denotes the first observed location (in the observed interval), whereas $o([t_1, t_2])(1)$ stands for the last observation (in the interval).

The solution to a GR-C problem $\mathcal{R}_c$ is a probability distribution across $\mathcal{G}$ and—given that Ramirez and Geffner's insight is as relevant in this domain as in the other—we propose to achieve this probability distribution exactly as before, based on the cost difference between two optimal paths from the starting point to each goal: one that *embeds* the observations and one that does not. In other words, we can now restate Ramirez and Geffner's cost difference formula (Equation RG1) for the continuous setting as follows (here $x_g \in \mathcal{G}$ is a possible goal):

$$\textit{costdif}_{RG}(x_s, x_g, \mathcal{O}) = \textit{optc}(x_s, \mathcal{O}, x_g) - \textit{optc}^{\neg}(x_s, \mathcal{O}, x_g). \tag{RG3}$$

Using the above cost difference formula, we can calculate the probability distribution given by the template (Equation RG2) as before, without any loss of meaning.

This concludes reformulation of our account for goal recognition into the continuous setting. As stated before, our motivation for generalising the domain has not been merely to demonstrate equivalence of results, but to set up the ground for Section 5.

4. "Single-Observation" Goal Recognition

In this section, we show that—in navigational domains—a probability distribution that ranks goals in the same order as Ramirez and Geffner's model can be achieved without negative reasoning and without even referencing the observation history of the agent whose goal is of interest, making the solution simpler and faster to calculate.

Consider an agent using probability distribution $P_{rg}(\cdot)$ to reason about another agent's travel. Using cost difference Equation (RG1), the first agent must perform two planning tasks for each goal g in $\mathcal{G}$: one to extract $\textit{optc}(s, O, g)$ and another to extract $\textit{optc}^{\neg}(s, O, g)$. 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.

In our re-working of this solution, we first eliminate the need for negative reasoning; we then eliminate the need to track incremental observations and show that—provided we know the agent's starting location and where it is "now"—we can rank goals in the same order by probability as if we were using the original, slower and more computationally demanding formula. In Sections 4.1 and 4.2, we present our solution for the discrete domain, assuming a GR problem of the form $\mathcal{R} = \langle \mathcal{D}, \mathcal{G}, s, O, \textit{Prob} \rangle$, as per Definition 3. We then (in Subsection 4.3) demonstrate that the formulas we present here also apply in the continuous domain.

4.1 GR without Negative Reasoning

Unlike task-planners, path-planning systems can readily accommodate observation requirements because they are simply treated as waypoints (nodes that must be visited). However, having a less expressive representation, 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.[12] Of course, one can achieve the desired result by calling a path-planner multiple times or by modifying the path-planner; but either method makes the minimum-cost calculation cumbersome and computationally expensive.

To address this, we adopt an alternative formulation (as used by Ramirez and Geffner in their 2009 paper to achieve a *non*-probabilistic solution) whereby, instead of calculating and deducting optimal cost "given not the observations", we simply deduct the more readily available "optimal path cost". As discussed by Escudero-Martin et al. (2015), this coincides with the intuition that, in the great majority of cases, an optimal path that "does not pass through all observed locations" *is* an optimal path. Formally:

$$\mathit{costdif}_1(s, g, O) = \mathit{optc}(s, O, g) - \mathit{optc}(s, g). \tag{1}$$

This formulation is conceptually simpler *and* computationally less demanding, in that there is no requirement to reason negatively about the observations in the second term. What is more, since the cost of an optimal path to each potential goal $g \in \mathcal{G}$ is not dependent on the observations, the second term can be pre-computed once for each goal at the outset. 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, which has a fixed, finite number of entrances and boarding gates, then all $\mathit{optc}(s, g)$ terms can be pre-computed and stored for retrieval as needed in constant time.

We point out that Ramirez and Geffner explicitly reject this simpler formulation (Ramirez & Geffner, 2010, p.1123) for general goal recognition by reference to a particular example (which we review in Section 7). Here, we aim to better understand the differences between Equations (RG1) and (1) and their actual impact. To that end, we now demonstrate not only that Equation (1) is simpler and easier to compute, but that it provides an identical result in all cases bar one; and that, even then, the difference has minimal impact on the overall probability distribution across potential goals $\mathcal{G}$. In addition, we show that in one corner-case, Equation (1) actually enables calculation of the posterior probability distribution when the original, more involved formula (RG1) may not.

Note that all of the cases (1-4), set out below with respect to 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 where observations conform to a suboptimal path, and hence to the observation of an agent whose behaviour is not completely rational. That is, given the steps already observed, the cheapest available path from s to a potential goal $g \in \mathcal{G}$ will inevitably be suboptimal.

We remind the reader that *the need to accommodate suboptimality in observations was the primary motivation behind the development of the probabilistic GR framework* (Ramirez & Geffner, 2010), as compared to the previous non-probabilistic framework (Ramirez & Geffner, 2009). We argue too that accommodating observations from agents whose behaviour is not completely rational/optimal is fundamental: in most real-world settings, intelligent agents (and humans) are indeed rational but only to some degree.

So, our first result states that, when the observed behaviour is suboptimal, the simpler formula (1) yields *exactly the same* value as the original formula (RG1).

12. It would be possible (though inefficient) to encode a path-planning problem as STRIPS but we wish to use one of the numerous algorithms (such as Dijkstra's algorithm (1959) and its derivatives), optimised for this specialised task.

Theorem 1 *Let O be an observation sequence such that $\mathit{optc}(s, O, g) > \mathit{optc}(s, g)$ (i.e., the observed behaviour is not optimal). Then, $\mathit{costdif}_{RG}(s, g, O) = \mathit{costdif}_1(s, g, O)$.*

Proof. Let π^* be an optimal solution path from s to g, that is, $\pi^* \in \Pi^*(s, g)$. Then, $\mathit{cost}(\pi^*) = \mathit{optc}(s, g)$. We can then conclude that path π^* does *not* embed O, otherwise, we would have $\mathit{optc}(s, O, g) = \mathit{optc}(s, g)$. Hence, since π^* does not embed O and is optimal among all solution paths, we get $\mathit{optc}^{\neg}(s, O, g) = \mathit{cost}(\pi^*) = \mathit{optc}(s, g)$, and since $\mathit{optc}^{\neg}(s, O, g) = \mathit{optc}(s, g)$, $\mathit{costdif}_{RG}(s, g, O) = \mathit{costdif}_1(s, g, O)$ follows. $\square$

Case 2: Optimal paths (non-exclusive). We now consider the case in which observations do conform to an optimal path, but they are not the only way to behave optimally. In path-planning, it is unusual to encounter a solution path, optimal or suboptimal, whose cost is unique. This is particularly true in a gridworld environment, where there may be thousands of optimal paths to a goal due to symmetries (Harabor & Grastien, 2012).

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

Theorem 2 *Let O be an observation sequence s.t. $\mathit{optc}(s, O, g) = \mathit{optc}(s, g)$ (i.e., the observed behaviour is optimal). If $\Pi^*(s, g) \setminus \Pi^*(s, O, g) \neq \emptyset$, then $\mathit{costdif}_{RG}(s, g, O) = \mathit{costdif}_1(s, g, O)$.*

Proof. Take $\pi' \in \Pi^*(s, g) \setminus \Pi^*(s, O, g)$: path π' is an optimal path from s to g, but it does *not* embed O. Because π' is optimal, $\mathit{cost}(\pi') = \mathit{optc}(s, g)$, and since it does not embed O we can conclude that $\mathit{optc}^{\neg}(s, O, g) = \mathit{cost}(\pi')$. Thus, $\mathit{optc}^{\neg}(s, O, g) = \mathit{optc}(s, g)$, and $\mathit{costdif}_{RG}(s, g, O) = \mathit{costdif}_1(s, g, O)$ follows. $\square$

That is, even if the observed behaviour is indeed fully rational, if there are other ways of behaving rationally, then the simpler formula (1) is, once again, *exactly equivalent* to the original formula (RG1).

Case 3: Optimal paths (exclusive). We now consider the only situation in which cost difference Equations (RG1) and (1) return *different* results: when observations are not only sufficient for optimal behaviour, but also *necessary*. In this case we say that the observations are ***exclusively optimal***: the only way of behaving fully rationally involves taking a path that embeds the observations.

Theorem 3 *Let O be an observation sequence and $g \in \mathcal{G}$. It is the case that $\mathit{costdif}_{RG}(s, g, O) \neq \mathit{costdif}_1(s, g, O)$ iff $\Pi^*(s, O, g) = \Pi^*(s, g)$ (i.e., all optimal paths to g embed the observations).*

Proof. The (ONLY-IF) follows directly from Theorem 2 and the fact that $\Pi^*(s, O, g) \subseteq \Pi^*(s, g)$. For the (IF) direction, suppose that $\Pi^*(s, O, g) = \Pi^*(s, g)$, that is, all optimal paths embed the observations. Take any path π from s to g (i.e., $\pi \in \Pi(s, g)$) that does not embed O, that is, $\pi \notin \Pi(s, O, g)$. Then, $\pi \notin \Pi^*(s, O, g)$ and since $\Pi^*(s, O, g) = \Pi^*(s, g)$, $\pi \notin \Pi^*(s, g)$ follows. Given that $\pi \in \Pi(s, g)$, we get that $\mathit{cost}(\pi) > \mathit{optc}(s, g)$. As path π was arbitrarily chosen, $\mathit{optc}^{\neg}(s, O, g) > \mathit{optc}(s, g)$, and $\mathit{costdif}_{RG}(s, g, O) \neq \mathit{costdif}_1(s, g, O)$ follows. $\square$

Observe that exclusive optimality is a corner case and, arguably, less relevant to the expected suboptimal behaviour that the probabilistic GR framework was designed to handle; but regardless of how relevant or interesting this corner case 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 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*.

With this in mind, we prove in Theorem 4 below that, in practice, even if an agent is observed taking an exclusively optimal path to a goal (i.e., all optimal paths 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. First, we make the following auxiliary observation.

Observation 1 *Let $costdif_f$ be some (cost difference) function and let P_f be the template probability distribution* (RG2). *If $costdif_f(s, g_1, O) < costdif_f(s, g_2, O)$ then $P_f(g_1 \mid O) > P_f(g_2 \mid O)$.*

This just restates the intuition that the lower the cost difference, the more probable the goal, and it follows from the fact that (RG2) is provably equivalent to the account given by Ramirez (2012).

Theorem 4 *Let O be an observation sequence such that $\Pi^*(s, O, g) = \Pi^*(s, g)$ for some $g \in \mathcal{G}$, that is, the observations are exclusively optimal for potential goal g (i.e., case of Theorem 3). Suppose further that $optc(s, O, g') > optc(s, g')$, for every $g' \in \mathcal{G} \setminus \{g\}$, that is, observations would result in suboptimal paths to all the other possible goals. Then, for all distinct $g_1, g_2 \in \mathcal{G}$, it is the case that $P_1(g_1 \mid O) > P_1(g_2 \mid O)$ if and only if $P_{RG}(g_1 \mid O) > P_{RG}(g_2 \mid O)$.*

Proof. Take any $g' \in \mathcal{G} \setminus \{g\}$. From Theorem 1, we know that $costdif_{RG}(s, g', O) = costdif_1(s, g', O)$ and $optc^{\neg}(s, O, g') = optc(s, g')$. Since $optc(s, O, g') > optc(s, g')$, using Equation (1) we conclude that:

$$costdif_{RG}(s, g', O) = costdif_1(s, g', O) > 0.$$

So, all cost difference values will be the same and greater than zero, using either formula, for all goals different from g. It remains to verify goal g. Because $\Pi^*(s, O, g) = \Pi^*(s, g)$, that is, O is necessary to travel from s to g in any optimal way, any route that does not embed O must be suboptimal. Formally, $optc^{\neg}(s, O, g) > optc(s, g) = optc(s, O, g)$. Using this in Equation (RG1) we get that $costdif_{RG}(s, g, O) < 0$. In turn, from Equation (1), we get that $costdif_1(s, g, O) = 0$. Thus, from what we have shown, for all $g' \in \mathcal{G} \setminus \{g\}$, we conclude that:

- $costdif_1(s, g, O) = 0 < costdif_1(s, g', O)$; and
- $costdif_{RG}(s, g, O) < 0 < costdif_{RG}(s, g', O)$.

Putting it all together, both cost difference accounts rank all goals in $\mathcal{G}$ equivalently. That is, for all $g_1 \neq g_2 \in \mathcal{G}$, $costdif_{RG}(s, g_1, O) < costdif_{RG}(s, g_2, O)$ iff $costdif_1(s, g_1, O) < costdif_1(s, g_2, O)$ and, using Observation 1, the thesis follows. □

Thus, even in this corner case, the simpler cost difference formula (1) lets us determine the same ranking among potential goals as (RG1) and hence is sufficient to identify the "most probable" goal.

The only case of exclusive optimality for which Theorem 4 does not apply arises if 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 could arise, for example, if the goals in question were aligned "sequentially". In this case, the complex formula (RG1) would return multiple (negative) cost differences to yield a ranking across the goals, whereas the simple formula (1) would rank all goals for which observations match their optimal paths equally (arguably, the outcome that one would expect). There are realistic scenarios where this situation could arise, which we will discuss at Section 7. For now, observe that this is a corner case and concerns *the very set of goals in which the probabilistic account (Ramirez & Geffner, 2010) is least interested.*

Case 4: A single solution path. Finally, in the extreme case, where observations conform not just to the only optimal path to a goal g but to the *only* path per se, 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) point out, (RG1) ought to return $-\infty$ giving 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 is easily rectified by allocating some minimum value instead of $-\infty$ or treating this case separately. In an identical situation, however, Equation (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.

4.2 GR without the Observation Sequence

As we have seen, in all but one corner case, the simple cost difference formula (1) can be used interchangeably with the more complex formula (RG1). We now go further and prove that, given a known starting point, the ranking among potential goals, as judged by probability distribution formula $P_1(\cdot)$, can be achieved without needing to know anything about the agent other where it is "now". This finding seems counter-intuitive and surprising; indeed, it implies that we can perform goal recognition without observing how the agent behaves over time! Nevertheless, as we will show in Theorem 5—under the Ramirez and Geffner model in the context of path-planning—probability rankings at any point in the domain 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, it is defined as $costdif_2 : N \times N \times N \mapsto \mathbb{R}$:

$$\mathit{costdif}_2(s, g, n) = \mathit{optc}(n, g) - \mathit{optc}(s, g). \tag{2}$$

Note that n, here, 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 their destination? Recall that $P_2(\cdot)$ is the templated probability function (RG2) generated using cost difference formula (2) above, while $P_1(\cdot)$ uses cost difference formula (1). We now show the following.

Theorem 5 *Let O be an observation sequence such that $n = O^{|O|}$ (i.e., n is the last observation in O). Then, for all $g_1, g_2 \in \mathcal{G}$, $P_1(g_1 \mid O) > P_1(g_2 \mid O)$ iff $P_2(g_1 \mid n) > P_2(g_2 \mid n)$.*

Proof. From Observation 1, $P_1(g_1 \mid O) > P_1(g_2 \mid O)$ if and only if $\mathit{costdif}_1(s, g_1, O) < \mathit{costdif}_1(s, g_2, O)$. Recall, from Equation (1), that for each $i \in \{1, 2\}$:

$$\mathit{costdif}_1(s, g_i, O) = \mathit{optc}(s, O, g_i) - \mathit{optc}(s, g_i),$$

where (remembering $n = O^{|O|}$) the first term can be written as:

$$\mathit{optc}(s, O, g_i) = \mathit{optc}(s, O^0) + \mathit{optc}(O) + \mathit{optc}(O^{|O|}, g_i) = \mathit{optc}(s, O^0) + \mathit{optc}(O) + \mathit{optc}(n, g_i).$$

Now, from Observation 1, recall that the relative ranking between g_1 and g_2 with respect to their posterior probabilities can be deduced directly from the relative value of their cost difference formulas. So, let us expand that value:

$$\begin{aligned}
&\mathit{costdif}_1(s, g_1, O) - \mathit{costdif}_1(s, g_2, O) \\
&\quad = [\mathit{optc}(s, O^0) + \mathit{optc}(O) + \mathit{optc}(n, g_1) - \mathit{optc}(s, g_1)] - \\
&\qquad\quad [\mathit{optc}(s, O^0) + \mathit{optc}(O) + \mathit{optc}(n, g_2) - \mathit{optc}(s, g_2)] \\
&\quad = \mathit{optc}(s, O^0) + \mathit{optc}(O) + \mathit{optc}(n, g_1) - \mathit{optc}(s, g_1) - \\
&\qquad\quad \mathit{optc}(s, O^0) - \mathit{optc}(O) - \mathit{optc}(n, g_2) + \mathit{optc}(s, g_2) \\
&\quad = \mathit{optc}(n, g_1) - \mathit{optc}(s, g_1) - \mathit{optc}(n, g_2) + \mathit{optc}(s, g_2) \\
&\quad = [\mathit{optc}(n, g_1) - \mathit{optc}(s, g_1)] - [\mathit{optc}(n, g_2) - \mathit{optc}(s, g_2)] \\
&\quad = \mathit{costdif}_2(s, g_1, n) - \mathit{costdif}_2(s, g_2, n).
\end{aligned}$$

So, $\mathit{costdif}_1(s, g_1, O) > \mathit{costdif}_1(s, g_2, O)$ iff $\mathit{costdif}_2(s, g_1, n) > \mathit{costdif}_2(s, g_2, n)$, and by applying Observation 1, we get $P_1(g_1 \mid O) > P_1(g_2 \mid O)$ iff $P_2(g_1 \mid n) > P_2(g_2 \mid n)$. □

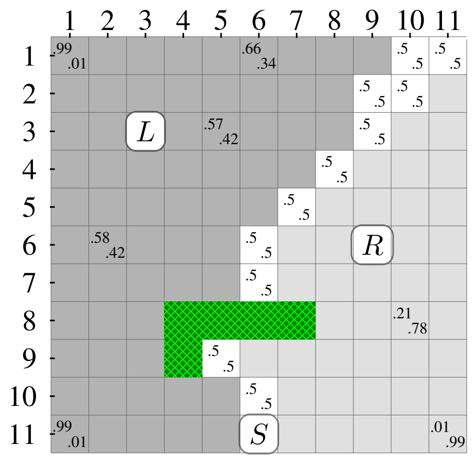
The finding is useful and unexpected. It tells us that we can achieve the same recognition, at the qualitative level, by only considering the "last" observation (rather than a complete observation sequence). This is important because it allows us to judge every location node *as if* an agent were observed 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 obtained 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 the starting location and all candidate goal locations are known—as would typically be the case in most domains—formula $\mathit{costdif}_2(s, g, n)$ can be fully *pre-computed offline* for any node $n \in N$ in the domain.

The implications are significant. This means not only can we perform *online* goal recognition without having to track the agent's movements (and therefore without incremental reasoning) but, as an alternative strategy, we could create a sort of "heatmap" of the domain (see Figure 1), showing the probability of each goal at each location, according to where the agent entered. Armed with such a heatmap, we can use it as a "lookup" for online recognition or, 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, using this approach we can just monitor the "hot" spots and only start tracking the agent if it arrives at one of them.

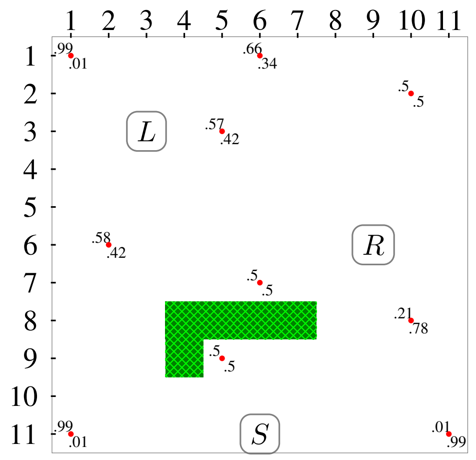
We note that the above result is actually not dependent on Equation (RG2) itself. Rather, it is applicable whatever manipulation is used to derive the probability distribution, provided that the posterior probability function satisfies the property that the lower the cost difference, the higher the probability,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

4.3 "Single-Observation" Recognition in the Continuous Domain

As we saw in Section 3, the path-planning framework for the continuous domain is significantly different from that of the discrete domain in its representation of space, paths, cost model, and observations. It turns out, however, that the results presented above in relation to the discrete domain,



(a) In a discrete domain, probabilities can be pre-calculated at every cell (or node). In dark shaded areas, the probability of L dominates; in lightly shaded areas, R dominates.



(b) In a continuous domain, probabilities can be calculated for all points of interest. To achieve a full heatmap, however, the domain needs to be discretised, e.g., to a grid as at (a).

Figure 1: Heatmap. Probabilities depend on location and can be pre-calculated if the start is known. Although, for legibility, only two goals are shown, $costdif_2(\cdot)$ can be substituted for $costdif_{RG}(\cdot)$ in Equation (RG2) to generate a complete probability distribution for any number of goals.

also hold for the continuous model, as we now show. (Definitions below are given relative to a GR-C problem $\mathcal{R}_c = \langle \mathcal{D}_c, \mathcal{G}, x_s, T, \mathcal{O}, Prob \rangle$.)

We saw in Section 3.2 that we can restate Ramirez and Geffner's cost difference formula (RG1) in the continuous setting as $costdif_{RG}(x_s, x_g, \mathcal{O})$. Making similar substitutions, we now restate our simpler formula (1) and the single-observation formula (2) for the continuous setting as follows (here $x_g \in \mathcal{G}$ is a possible goal, and $x_n \in X_{free}$):

$$costdif_1(x_s, x_g, \mathcal{O}) = optc(x_s, \mathcal{O}, x_g) - optc(x_s, x_g) \tag{3}$$
$$costdif_2(x_s, x_g, x_n) = optc(x_n, x_g) - optc(x_s, x_g) \tag{4}$$

With that reformulation in place, the behaviour of the simple formula is unchanged, as follows.

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

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

This theorem states that, as in the discrete domain, $costdif_{RG}(x_s, x_g, \mathcal{O})$ and $costdif_1(x_s, x_g, \mathcal{O})$ for the continuous setting yield different results *only* in the case of "exclusive optimality" (i.e., when the *only* way to achieve an optimal path is via the observations).

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-C 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(π). 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, $\textit{optc}(x_s, \mathcal{O}, x_{g_i}) = \textit{optc}(x_s, \pi^*(0)) + \textit{cost}(\pi^*) + \textit{optc}(\pi^*(1), x_{g_i})$. Similarly, therefore:

$$\begin{aligned}
&\textit{costdif}_1(x_s, x_{g_1}, \mathcal{O}) - \textit{costdif}_1(x_s, x_{g_2}, \mathcal{O}) \\
&\quad = [\textit{optc}(x_s, \pi^*(0)) + \textit{cost}(\pi^*) + \textit{optc}(\pi^*(1), x_{g_1}) - \textit{optc}(x_s, x_{g_1})] - \\
&\qquad\quad [\textit{optc}(x_s, \pi^*(0)) + \textit{cost}(\pi^*) + \textit{optc}(\pi^*(1), x_{g_2}) - \textit{optc}(x_s, x_{g_2})] \\
&\quad = [\textit{optc}(\pi^*(1), x_{g_1}) - \textit{optc}(x_s, x_{g_1})] - [\textit{optc}(\pi^*(1), x_{g_2}) - \textit{optc}(x_s, x_{g_2})] \\
&\quad = \textit{costdif}_2(x_s, x_{g_1}, x_n) - \textit{costdif}_2(x_s, x_{g_2}, x_n).
\end{aligned}$$

Since $\textit{costdif}_1(x_s, x_{g_1}, \mathcal{O}) > \textit{costdif}_1(x_s, x_{g_2}, \mathcal{O})$ iff $\textit{costdif}_2(x_s, x_{g_1}, x_n) > \textit{costdif}_2(x_s, x_{g_2}, x_n)$, we get $P_1(x_{g_1} \mid \mathcal{O}) > P_1(x_{g_2} \mid \mathcal{O})$ iff $P_2(x_{g_1} \mid n) > P_2(x_{g_2} \mid x_n)$. $\square$

4.4 Properties and Constraints of Single-Observation Recognition

We have shown (by Theorems 5 and 7) that, under Ramirez and Geffner's model for probabilistic goal recognition as specialised to path-planning, unless an agent must pass through all observations to behave optimally (i.e., the special case of exclusive optimality excepted by Theorem 3), if we know where it started from, the path it takes has *no effect whatsoever* on the probability rankings of its possible destinations. Even on reflection, the finding is counter-intuitive: how is it that we can predict where an agent is going whether or not we know where it has been? On careful consideration, however, the Markovian nature of our solution is not quite so unlikely as it seems. There are three sound bases, as follows.

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, i.e., its location (a contraction, as discussed, of $go(x)$).

2. The agent's starting location is known, supplied as input to the GR-D or GR-C problem. Thus, given its current location (and given that, in this environment, location implies full observability), we are implicitly aware of everything relevant that has *changed* since it entered the domain. Hence, the optimal path to its 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.

If any of these conditions were not met, single-observation goal recognition would be unlikely to succeed, as discussed further in Section 7. Provided they *are* met, however (as they *always* are in the core navigational domains with which we are concerned), probability distribution $P_2(\cdot)$, which uses single-observation cost difference formula (2), yields exactly the same probability rankings across goals as $P_1(\cdot)$ using cost difference formula (1). We can therefore make the following specific claims for single-observation recognition by comparison with Ramirez and Geffner's original, more complex formulation.

- $P_2(\cdot)$ results in the *same goal rankings* as $P_{RG}(\cdot)$ whenever observations conform to a suboptimal path; when they conform to an optimal path that is non-exclusive (i.e., it is not the only optimal path); and even if they conform to a path that is exclusively optimal for one goal but would be a suboptimal way of reaching the others.

- $P_2(\cdot)$ only results in different rankings from $P_{RG}(\cdot)$ if observations conform to an *optimal* path for multiple goals and are *exclusively optimal* for at least one. We discuss this case in Section 7 but, for now, note that this represents precisely that set of goals which are *not* the motivation of the probabilistic GR model.

- $P_2(\cdot)$ is computationally advantageous. Like $P_1(\cdot)$, it does not require any negative reasoning about observations or, in fact—and here it differs from $P_1(\cdot)$—any reasoning about observations at all. It requires just two optimal paths from point to point per goal. Thus, it can be solved by an even more rudimentary path-planning algorithm than can $P_1(\cdot)$, which must plan an optimal path *through waypoints*.

To elaborate more formally, calculation of $P_{RG}(\cdot)$ requires $2|\mathcal{G}|$ calls to the path-planner, whereas $P_1(\cdot)$—and by extension $P_2(\cdot)$—require only $|\mathcal{G}|$ calls. In practice, the time-saving is more emphatic. Time taken is *more* than halved using $P_1(\cdot)$ and $P_2(\cdot)$ because the call to the planner that does *not* now occur 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 which, depending on their distance apart and the intricacy of the underlying terrain, can take considerably longer than finding a single path from point to point.[13]

The above discussion highlights points of *comparison* between GR using the single-observation formula (2) and the original, more complex formula (RG1). Note, however, that 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 that directly advances towards a gate should be assigned a higher probability with respect 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.

In this section, we demonstrated a fast and computationally undemanding method of goal recognition. Using this method, we can generate a probability distribution for any location 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 point at which a "goal of interest" becomes the most probable goal; and we do this, not only independently of the observation sequence and without negative reasoning, but without even calculating any probabilities.

5. Radius of Maximum Probability

We demonstrated in Section 4 that, given an agent's starting location, probabilities consistent with $P_{RG}(\cdot)$ can be generated for any point in the domain and used to create a probabilistic heatmap; we

13. In the worst case, path-planning through $|O|$ waypoints may involve $|O|$ path-planning tasks, using subgoaling.

now show that the radius within which a goal is guaranteed to be the most probable can be obtained using the (typically available) continuous measurements between a starting point for the domain and a set of candidate goal locations.[14]

Observe that this approach is not a substitute for single-observation recognition. With the single-observation formula, we can calculate a full probability distribution for a set of goals at any one node or point. Using the RMP—and based on very similar parameters (i.e., optimal costs between goals and the start location)—we can calculate a radius within which any one particular goal is the most probable; and that radius might encompass hundreds of nodes or infinitely many points. There is a trade-off, of course, and it is important to state right away that, although the RMP guarantees maximum probability of a goal *within the calculated radius* it says *nothing* about probabilities outside that radius.

We first demonstrate calculation of the RMP in the context of a continuous domain, then show how it may be applied in the discrete domain.

5.1 RMP in the Continuous Domain

Our method depends on the relationship, implicit in Equations (2) and (4), between a goal's probability and an agent's precise location. As discussed, a key implication of the single-observation formula is that, provided we use a probability distribution formula, such as Equation (RG2), that satisfies the property that the lower the cost difference, the higher the probability, the ranking of goals (with respect to their probabilities) at every point in the domain remains constant, regardless of the path taken to get there. We have also noted that the single-observation account in the discrete domain allows for probabilities to be pre-calculated node by node to create a "heatmap" (as at Figure 1) which reveals the perimeter within which a goal becomes "most probable".

We recognise that, even in a discrete domain, such calculations might be computationally demanding; in a continuous domain, it would clearly be impossible to pre-calculate probabilities for infinitely many points! Building on Equation (4), however—and provided that we know the relative location of the starting point and candidate goals (information typically available in a known domain)—an alternative approach is available, as we now show.

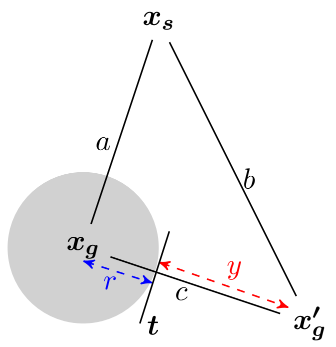
Let us first precisely define the cost-distance that we propose to measure.

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

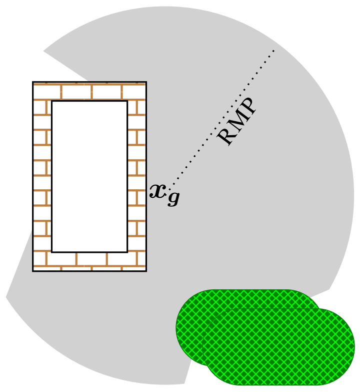
1. *for all $x \in X_{free}$ such that $optc(x, x_g) < r$, it is the case that $P(x_g \mid x) > P(x'_g \mid x)$, for all $x'_g \in \mathcal{G} \setminus \{x_g\}$; and*
2. *there exists a point $x' \in X_{free}$ such that $optc(x', x_g) = r$ and $P(x_g \mid x') = P(x'_g \mid x')$, for some $x'_g \in \mathcal{G}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

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. 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. Note that, since

14. We have previously presented this result in the context of discrete domains (Masters & Sardina, 2017b); the continuous setting provides a more adequate framework in which it is more readily understood and more applicable.



(a) Labels a, b, and c stand for optimal cost-distances. We show only two goals; if we find t between x_g and *multiple* goals, then RMP_{x_g} will be the *minimum* r.



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

Figure 2: The Radius of Maximum Probability for possible goal x_g.

probabilities are based on cost difference, a point $x \in X_{\mathit{free}}$ at the tipping point will have the property (by Observation 1) that $\mathit{costdif}_2(x_s, x_g, x) = \mathit{costdif}_2(x_s, x'_g, x)$.

A Closed Formula to Calculate the RMP. Consider the simple two-goal domain depicted in Figure 2a. We wish to construct a formula for RMP_g, denoted r. So, take $a = \mathit{optc}(x_s, x_g)$, $b = \mathit{optc}(x_s, x'_g)$, $c = \mathit{optc}(x_g, x'_g)$, 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 $\mathit{optc}(t, x_g) = r$ and $\mathit{optc}(t, x'_g) = y$. Using Equation (4):

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

Thus we obtain r, the optimal cost from goal location x_g to the point at which t must occur. Since there may be *many* potential goals for consideration, to construct a general formula for RMP_g, we take the minimum (recall x_s is the starting point and $\mathcal{G}$ is the set of possible goals):

$$\text{RMP}_g(x_s, \mathcal{G}) = \min_{x'_g \in \mathcal{G} \setminus \{x_g\}} \frac{\mathit{optc}(x_g, x'_g) + \mathit{optc}(x_s, x_g) - \mathit{optc}(x_s, x'_g)}{2}. \tag{5}$$

Calculation of RMP_g (as above) requires $2|\mathcal{G}| - 1$ calls to a path-planner. If we wished to calculate RMPs for every goal in the domain, it would take $\frac{|\mathcal{G}|^2 + |\mathcal{G}|}{2}$ calls. As an 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. This is approximately twice the computational cost of calculating *a single probability distribution* using $P_2(\cdot)$, yet the area that the RMP describes may contain very 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*.

Equation (5) is significant because it enables us to identify a clear boundary—a target area—*within which a goal is guaranteed to be the most probable*, whilst freeing us from calculating the probabilities at any particular location (or any location at all). Note, again, that this closed equation arises as a direct consequence of the single-observation formula for cost difference, Equation (4).

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 2b. The green "forest" represents an obstacle: 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 2b 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. (Observe also that the impossibility of traversing those inaccessible areas means that other points (unshaded) become more distant from x_g and are forced outside RMP$_g$.)

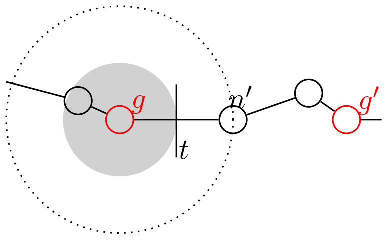
2. The fact that a point x lies *beyond* the cost-distance represented by RMP$_g$, does not imply that the probability of x_g at x is *less* than the probability of some other goal. Recall that the guarantee concerns only those points *within* RMP$_g$; it says *nothing* about points outside that radius. Indeed, there may many be points *outside* 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 3b conveniently illustrates the case. If we calculate probabilities for an agent at cell (9,11), its most probable goal is g, approximately 10 units away. An agent at (4,5), on the other hand—only 4 units from g—is most probably heading for goal g' (not g). 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, RMP$_g < 4$ even though g is the most probable goal at other cells 8, 9 and 10 units away.

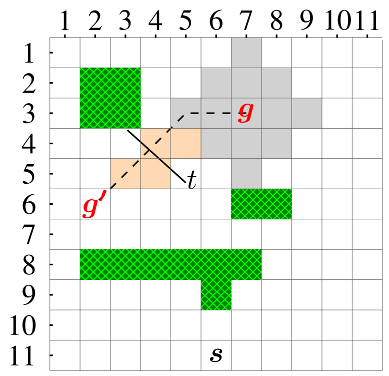
5.2 RMP in the Discrete Domain

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 goal where the tipping point between goals ought to occur.

Graph-based representation. Referring to Figure 3a, 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



(a) In a graph, the tipping point may fall between nodes: there is no node where $P(g) = P(g')$.



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

Figure 3: The RMP in discrete domains.

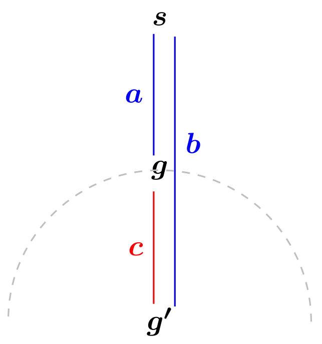
candidate goals, even in the presence of no obstacles (i.e., $X_{obst} = \emptyset$). Thus, in the discrete domain, the RMP represents a *theoretical* minimum distance; in practice, a no less useful lower bound.

Referring again to Figure 3, we see that, if there *were* a node at t we could say that $P(g \mid t) = P(g' \mid t)$ and that at all nodes closer to goal than t (within the shaded area), the probability of g dominates. In this example, however, the first existing node where $P(g) \leq P(g')$ is the node marked n', so the shaded area could be extended up to the radius marked by the dotted line. Observe that the cost-distance $optc(n', g)$ is in fact *greater* than the value returned by Equation (5). Thus, in a domain discretised into a graph, the RMP provides a lower bound for the distance of interest.

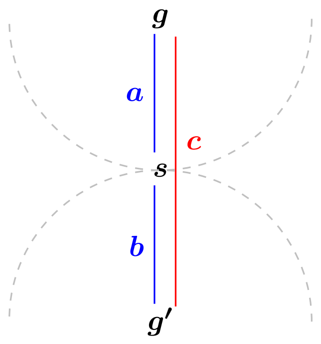
Gridworld representation. When a path-planning space is discretised into a grid, the situation is slightly different. In a gridworld environment, nodes are represented as cells and every cell is immediately adjacent to some other cell. Therefore, it can never happen that the tipping point t occurs at 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 3b, where cost is 1 for straight moves and $\sqrt{2}$ for diagonal moves, Equation (5) returns a value of 3.53 for RMP_g, but the cost of cell traversal on an optimal path from g to g' 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, one cannot know whether probabilities at n_t favour g or (some other goal) g' or are equal. Nonetheless, what we *do* know is that for all cells n' such that $optc(n', g) < optc(n_t, g)$, it is the case that $P(g \mid n') > P(g' \mid n')$, for all $g' \in \mathcal{G} \setminus \{g\}$.

Heatmap example. We conclude this subsection by reminding the reader of the heatmap example discussed in Subsection 4.2. We explained there that one could use Equation (2) to pre-calculate a complete heatmap of probabilities (Figure 1) from which to identify the perimeter within which goal L or R was the most probable. Using Equation (5), we can now instead *instantly* calculate the



(a) The optimal path to g' goes through g, so $\text{RMP}_g = 0$.



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

Figure 4: Properties of RMPs. Dotted grey lines show RMPs for g and g'.

minimum distance of that perimeter from either goal without pre-calculating any probabilities:

$$\text{RMP}_R = \frac{9+8-11}{2} = \frac{6}{2} = 3$$
$$\text{RMP}_L = \frac{9+11-8}{2} = \frac{12}{2} = 6$$

Using this calculation—and nothing else—we now have a clear target area within which to focus our attention. If we have a particular interest in monitoring or protecting goal L or R, 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.

To clarify: the heatmap is a hypothetical artefact that can be created by the repeated application of cost difference formula (2) for every node in the domain (or every location of interest) at a computational cost of $|\mathcal{G}| \cdot k$ 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 but, once calculated in an offline process, can also be used to access a complete probability distribution for every point that has been evaluated; and this 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 is similar to "worst case distinctiveness", or in place of landmarks for goal recognition and/or goal pruning (Vered et al., 2018). Furthermore, being quickly calculable (unlike a heatmap), the RMP can be used in online scenarios where the terrain changes dynamically (e.g., Raffe, Zambetta, & Li, 2012) or if candidate goals were previously *unknown* or agent-specific, making pre-calculation impossible.

5.3 Properties of the RMP

The RMP measure identifies a radius around 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, that the radius within which its probability dominates that of all other goals may be precisely zero.

Consider the two goals at Figure 4a. Here, the goals and starting point are in a direct line, that is, the optimal path to g' goes directly through g. Calculating the RMPs for each goal, we get:

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

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

Consider now the two goals at Figure 4b. Again, the goals and starting point are in a direct line but this time the relationship is different in that the optimal path from g to g' goes directly through the starting point s. Calculating the RMPs for each goal, we obtain:

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

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. This could have the effect of placing the starting point s *inside* a goal's RMP: that is, one goal may be "most probable" from the start.

6. Experimental Evaluation

In this section, we report our results on the performance of goal recognition in path-planning when using the original (complex) cost difference (RG1), the simpler version (1) that does not reason negatively about observations, and the single-observation version (2). Tests were conducted in a discrete (gridworld) domain using problems adapted from the well-known MOVING-AI[15] path-planning benchmarks (Sturtevant, 2012), which discretise the underlying maps and groundplans to a 512×512 grid. The aim was to develop an experimental framework for the problem of goal recognition in path-planning to empirically confirm that *(i)* the case of observations conforming to the *only* optimal path to a goal (as in Theorem 3) is rare and, otherwise, the simpler formula (1) yields identical posterior probability distributions to formula (RG1); *(ii)* all three accounts return posterior probability distributions that rank goals the same; and *(iii)* use of either formula (1) or (2) cuts processing time by more than half. Note that we have relied on our theoretical conclusions to support the applicability of our formulas in the continuous domain.

15. http://movingai.com/

6.1 Experimental Setup

We generated over 2500 individual probability distributions from a new problem set of 774, built on 43 scenarios selected at random from two sets of MOVING-AI benchmarks (Sturtevant, 2012):[16] 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 goal recognition 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 a full continuous path from the start location to the real goal. We then extracted observation sequences varying three dimensions: path quality (optimal, suboptimal, greedy), observation density, that is, the proportion of the continuous path extracted to represent the observation sequence (sparse 20%, medium 50%, dense 80%) and two observation strategies (*random* extracts observations from random locations along the path, *prefix* extracts a consecutive sequence of location nodes from the start location).

In previous tests, we had found that the use of exponential values to generate posterior probabilities made formula (RG2) particularly sensitive to any small variation in cost difference. This was especially noticeable with the large negative values often returned by formula (2) (where optimal cost of a complete path is subtracted from a much smaller cost from the most recent observation). For this article, we therefore compared four probability distributions (rather than three). These are derived using formula (RG2) with the original (complex) formula (P_{RG}), the simpler formula (P_1), the single-observation formula (P_2), plus a variation obtained by adding a large constant value (which we set at 800) to the cost difference returned by the single-observation formula (P_2*). We had also found the probability formula very sensitive to the value of the β constant. For the automated tests, we adopted a value of 0.1 throughout. (With $\beta = 1$, P_{RG} and P_1 tend to return 1 or 0, while with $\beta = 0.01$, the distribution tends to even out and, with loss of precision on test equipment, returns e.g., 0.33 for each of three goals, 0.25 for each of four, and so on).

Other results discussed inline are drawn from previous tests. 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)[17]. To obtain the cost of an optimal path given *not* the observations, inspired by the technique used by Ramirez and Geffner (2010), 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 node—and so the associated path that embedded the observations—was pruned.

We used the usual uniform-cost approach for grids, with horizontal and vertical moves costed at 1, and diagonal moves at 1.414; and made the simplifying assumption that priors were equal.

16. New experiments were conducted on a i7 3.4GHz dual core with 10GB RAM in a virtual Linux environment; previous experiments on similar 1.8GHz machine.
17. We used a Python-based infrastructure, originally designed as a simulator and testbed for path-planning algorithms, which already included implementations of A* and Weighted A* in its library (https://tinyurl.com/p4sim).

6.2 Results

Our theoretical results were confirmed. In hand-crafted problems using maps with open landscapes, the formulas performed exactly as predicted: formulas (RG1) and (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 (1) and (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 (1) and (2) returned in 0.005 and 0.002 seconds, respectively, and successfully identified the real goal.

Unsurprisingly, given the symmetries found in a two-dimensional grid, the corner case in which observations conform to the only optimal path to goal did not arise in any of the randomly generated scenarios. We were only 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 5, respectively), whereas formulas (1) and (2) returned 0.333 for all three goals.

Tables 1 and 2 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 goal recognition 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.

The implementation using cost-difference formula (RG1) performed even more slowly than we expected. In the room layouts, as shown in Table 1, it frequently exceeded our three minute timeout; the longer the paths (i.e., the larger the set of observations) and the more optimal, 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). Our results also showed that 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 algorithm backtracks and, if observations are consecutive, repeatedly reaches the final observation via multiple different routes. Ultimately, however, 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 an important navigational element, though in the more demanding context of general task-planning)[18] 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

18. 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.

	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	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**

Table 1: Rooms
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).

suboptimal—and much faster—algorithm but, although it returned approximately equivalent probability distributions, it failed to preserve the corner cases, which were of interest to us.

As can be seen from the tables, use of the simple cost difference formula (1) cut processing time even from landscapes (Table 2) 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 (2) were consistently lower than those for the simple formula (1).

Whereas the probabilities based on the simple cost difference formula (1) always exactly matched those based on the original more complex formula (RG1), probabilities generated using single-observation cost difference (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 effect we noted in previous testing and the reason why, in this version of the paper, we include a full set of results using the modified probability distribution $P_2{}^*$. Delta values for P_2 were sometimes quite high. Whereas formula (1) subtracts optimal path cost from the cost of a (typically suboptimal) path that embeds observations, always resulting in a cost ≥ 0, formula (2) subtracts optimal path cost

	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**

Table 2: Landscapes
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 1. 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.

from the cost of a truncated path (only from the most recent observation). This inevitably results in negative cost difference for the "most likely" goal, which translates to negative exponentials when used as input to formula (RG2) (i.e., tiny fractions). Consequently, in implementation, the output for P_2 seems to lose precision, making the overall distribution flatten out. For $P_2{}^*$ 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.

7. Discussion

In this section, we discuss three 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, 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.

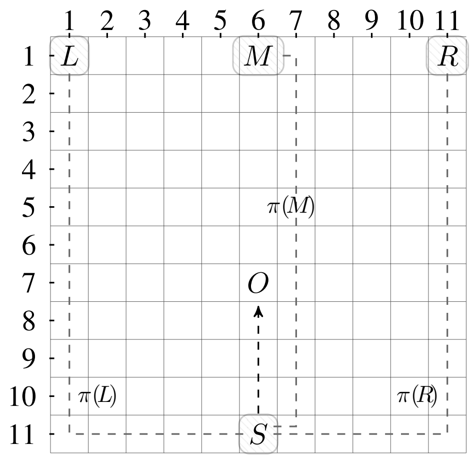


Figure 5: O is on an optimal path to all three goals but, by Equation (RG1), M is the most probable.

7.1 The Corner Case: Exclusive Optimality and Negative Reasoning

Formulas (1) and (2) rank goals differently from the original, more complex, cost difference formula (RG1) only where observations conform to the optimal path for *multiple* goals and are *exclusively* optimal for at least one of them. Using formulas (1) or (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.

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

Example 1 *Consider the situation depicted in Figure 5. An agent operates in a gridworld environment where the only legal moves are horizontal or vertical and all steps cost* 1*. There are three possible goals, L,* M *and* R*, all north of the start location,* S*. Observations track directly north through* O *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*$(S, L, O) = 0$ *and costdif*$(S, R, O) = 0$ *(see formula RG1). In the case of* M*, however, which lies directly north of* S *and* O*, 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*$(S, O, M) = 10$*, whereas optc*$^{\neg}(S, O, M) = 12$*. Thus, costdif*$(S, M, O) = -2$*. The lower the cost difference, the higher the probability, making* M *the most probable goal.*

Although cited as an "example" of the distinction between cost difference formulas (RG1) and (1), this scenario, in fact, depicts the case of exclusive optimality which represents the *only* distinction, as we proved in Theorem 3. Furthermore, it concerns the very set of goals in which the probabilistic account is least interested because goals on the *optimal* path were already handled in the non-probabilistic account (Ramirez & Geffner, 2009). Nevertheless, let us consider what is lost (and/or what is gained) by substitution of the simpler or single-observation formula for (RG1).

To arrive at a probability distribution, Ramirez and Geffner (2010) appeal to Bayes' Rule, $P(G \mid O) = \alpha P(O \mid G)P(G)$, where α is a normalising constant. Assuming that prior probabilities in

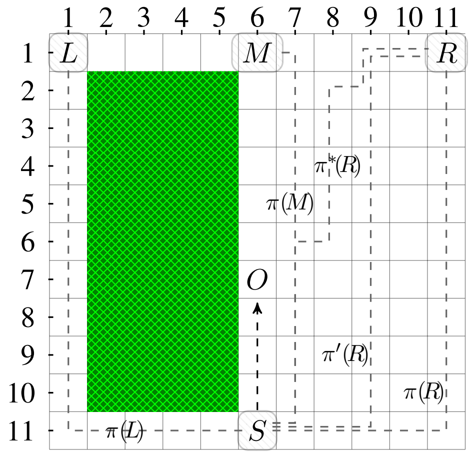


Figure 6: $P(L \mid O)$ is the same as $P(R \mid O)$.

$P(G)$ are given, the challenge is to account for $P(O \mid G)$ and it is correct, the authors say, that $P(O \mid M)$ in Example 1 above should exceed $P(O \mid L)$ and $P(O \mid R)$ because "goal M predicts the observations better than either L or R" (p.1123). The intuition is that, in order to reach M optimally, an agent from S *must* pass through the observation O; whereas, to reach L or R optimally, the agent *might* (or might not) pass through O.

The 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 6. The rectangular block of patterned green cells, $(2, 2)$ to $(5, 10)$, is not traversable. Again there are three goals, L, M and R but now L has been made to be very like M. There are just two optimal path to L, only one more than to M. Meanwhile, (owing to the notorious symmetry of gridworlds) there are 3003 optimal paths to R, as before, and yet Equation (RG1) can no more distinguish between L and R (which are "non-exclusively optimal") than can the simpler formula (1). In this scenario, the probability of L should—based on how well the goal predicts the observations—be (very much) greater than R (only a little less likely than M) but, for both goals L and R, both formulas return zero and, by the posterior probability calculation (RG2), both goals appear to be equally 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 L and R are equally likely; rather that it would have been just as reasonable to assume that L, M and R are *all* equally likely.

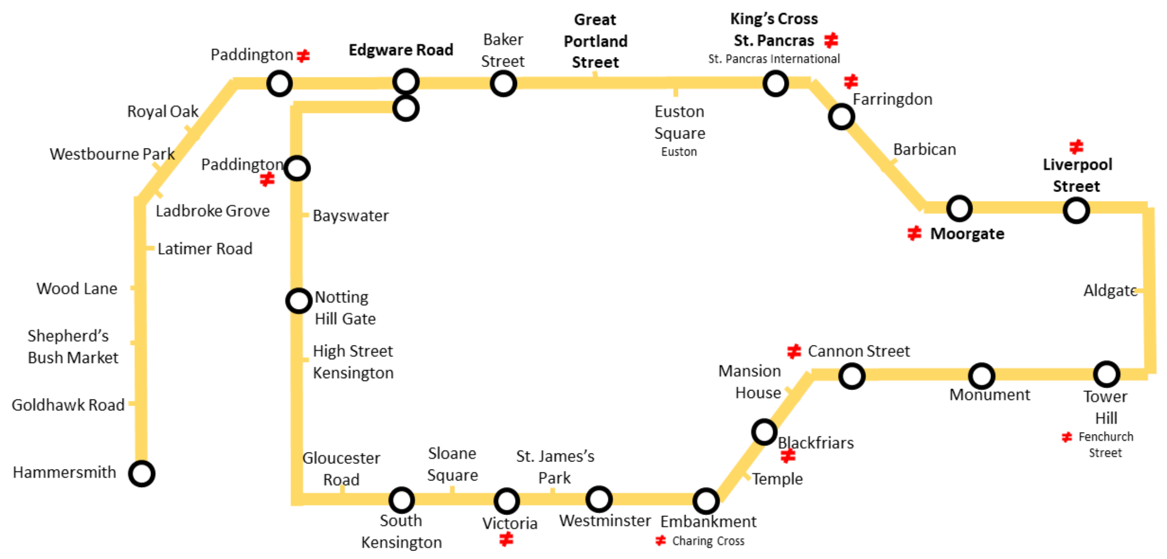


Figure 7: Using negative reasoning, an agent who boarded a train at Edgware Road and was observed at Great Portland Street is more likely to be travelling to Moorgate than Liverpool Street.

A more extreme—and more commonplace—example of exclusive optimality arises in the context of a transport network, such as a road network or the London Underground system. It is an interesting situation where we have an apparently continuous domain (the train moves continuously through the real world) even though, for practical purposes, the domain is discrete (a vehicle/passenger can only exit or change direction at an intersection/station). Exclusive optimality occurs routinely in this environment because there are frequently situations where the agent (having decided which "line" they are travelling on and in which direction) may be 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 simple formulas—in common, in fact, with the many other approaches that also 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.

Example 3 *Consider Figure 7, 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* (1) *or* (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).*

Whilst we must agree, 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, the question must be asked: in the case of Example 3, is it 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 Ramirez and Geffner's 2010 approach, which strictly applies Bayes' Rule and so suggests that King's Cross is more likely than Moorgate and Moorgate more likely than Liverpool Street?

7.2 Application of our Model in a Task-Planning Domain

There are three main components of our model: *(i)* a *simpler formula* (1), which eliminates negative reasoning from Ramirez and Geffner's original approach; *(ii)* a *single-observation formula* (2), which further eliminates an agent's observation history (between the initial and final locations); and *(iii)* the *RMP measure* (5), 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".

7.2.1 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 1 and 3) 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, $\mathit{optc}^{\neg}(s, O, g) = \mathit{optc}(s, g)$.

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 7.1 above (i.e., where—for at least one of the goals—there is no optimal plan 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 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 RMP, however, the situation is somewhat different.

7.2.2 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 inherited requirements (which also apply to Equations 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.[19]

If either one of these conditions can be met, the single-observation formula is also "plug and play". The conditions are necessary because 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 five metres; turn right," would not reveal the current location (e.g., other actions may have occurred after the agent turned left and before they walked five 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*.

19. 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).

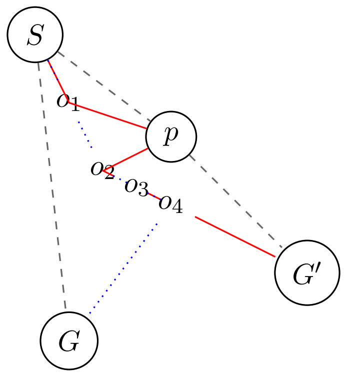


Figure 8: The optimal paths from S to o_4 that embed observations are different for each goal.

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.*

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 can be unreliable.

Example 6 *In the scenario depicted by Figure 8, 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 initial state to most recently observed action—and so whittle the family of possible states back down to one[20]—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 and/or whether they have 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 non-trivial task.

20. 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 article but suggests a most interesting avenue for future work.

7.3 Implications for Plan Recognition

In this paper, we have treated goal recognition 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; and our formalism is based on Ramirez and Geffner's account, which refers to the framework with the expression "plan recognition as planning" when it too focuses on goals.

Strictly, though, goal recognition 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 goal recognition 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, goal recognition 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 builds 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 goal recognition 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 "a 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 goal recognition 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 goal recognition 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’ and so on).

Either option would be a nice refinement. To our knowledge, however, no contemporary plan recognition system takes “all” observations into account in this way, making 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) a practical and competitive solution.

8. Conclusion

In this article, we have examined the “plan recognition as planning” approach to probabilistic goal recognition, first introduced in the principled work of Ramirez and Geffner (2010), and applied it in the context of path-planning. In particular, we have focused on how a cost-based approach to goal recognition, which they pioneered, can be exploited in core navigational domains to minimise the computational effort required to determine an agent’s most probable destination.

Following preliminaries, we showed, in Section 4.1, that the simpler cost difference formula (1) used by Ramirez and Geffner (2009)—which does not require reasoning negatively about observations (a demanding task) and is achievable by calls to a standard path-planner—yields an *identical result* to their 2010 formula (RG1) (which does reason negatively) in *all but one specific case*, which we characterised. We argued, in line with intuitions expressed by Ramirez and Geffner (2010), that this is a case of little interest. Then, in Section 4.2, we demonstrated that an even simpler cost difference formula that *does not even depend on the observation sequence*, namely, the single-observation formula (2), nevertheless generates a posterior probability distribution that *exactly preserves* the ranking of goals from the simplified account and, by extension, results in an *identical ordering* to formula (RG1) in all cases bar one. Besides confirming our theoretical results, the experiments reported in Section 6 provide evidence that both proposed alternative formulas reduce computation time significantly. Importantly, because the single-observation formula is independent of the observation sequence, it confers additional benefits. Not only can it be used *online* for on-the-spot checks of agents that 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 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.

In this extension to previous work (Masters & Sardina, 2017a), we generalised the problem definition (and our solution) to the continuous domain and, in Section 5, we used this generalised framework to develop the notion of a radius of maximum probability (or RMP)—introduced elsewhere (Masters & Sardina, 2017b) but fully realised and explored for the first time here. The RMP formula (5) is significant in that it formalises the relationship between precise location and probability (implicit in cost difference formula (2)) enabling calculation of 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. Thus, the RMP formula can be used offline, for example as a tool complementary to goal recognition design (Keren et al., 2014). Alternatively, since it is quickly calculable, it can be used online as well, for example to find RMPs in a game that automatically generates new (previously unknown) terrains (e.g., Raffe et al., 2012).

We conclude by reminding the reader that, in Subsection 7.2, we discussed the formal development of our model for GR in the context of task-planning domains. Though outside the scope of this article, we believe this presents a useful and interesting avenue for future work. In particular, we see the possibility of generating probabilistic heatmaps in the context of general goal recognition for task-planning as an appealing concept.

Acknowledgements

This article is an extension of our conference paper (Masters & Sardina, 2017a) incorporating the notion of a "radius of maximum probability" (Masters & Sardina, 2017b). We acknowledge the support of the Australian Research Council (DP120100332) and Australian Government Research Training Program Scholarship and thank the anonymous reviewers for their comments and insights, which have significantly helped to improve the quality of this work.

References

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).

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.

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.

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.

Charniak, E., & Goldman, R. P. (1991). *Probabilistic abduction for plan recognition* (Tech. Rep. Nos. CS-91–12). Brown University. Department of Computer Science.

Demeester, E., Nuttin, M., Vanhooydonck, D., & Van Brussel, H. (2003). A model-based, probabilistic framework for plan recognition in shared wheelchair control: Experiments and evaluation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vol. 2, pp. 1456–1461).

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).

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271.

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).

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).

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).

Geib, C., & Goldman, R. (2009). A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, *173*(11), 1101–1132.

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).

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.

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.

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).

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. (2006). *Adversarial reasoning: Computational approaches to reading the opponent's mind*. CRC Press.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge, U.K.: Cambridge University Press. (Available from http://planning.cs.uiuc.edu/)

Lefèvre, S., Vasquez, D., & Laugier, C. (2014). A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal*, *1*(1), 1–14.

Lesh, N., Rich, C., & Sidner, C. L. (1999). Using plan recognition in human-computer collaboration. In *UM99 User Modeling* (pp. 23–32).

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).

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).

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).

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. (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).

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).

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.

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.

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).

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., & Rosenbloom, P. (1995). Resc: An approach to agent tracking in a real-time, dynamic environment. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 103–110).

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).

Wiest, J., Höffken, M., Kreßel, U., & Dietmayer, K. (2012). Probabilistic trajectory prediction with gaussian mixture models. In *IEEE Intelligent Vehicles Symposium* (pp. 141–146).