Cost-Based Goal Recognition for Path-Planning

Peta Masters
RMIT University
Melbourne, Australia
peta.masters@rmit.edu.au

Sebastian Sardina
RMIT University
Melbourne, Australia
sebastian.sardina@rmit.edu.au

ABSTRACT

Recent developments in plan recognition have established a method of identifying an agent's most probable plan or goal by use of a classical planner and an appeal to the principle of rational action. We examine this technique in the strict context of path-planning. We show that a simpler formula provides an identical result in less than half the time under all but one set of conditions, which we identify. Further, we show that the probability distribution based on this technique is observation-independent and present a revised formula that achieves goal-recognition by reference to the agent's current location only.

1. INTRODUCTION

In this work, we investigate goal recognition in navigation. That is, we seek to determine where an agent may be travelling, given a set of potential destinations and some (partial) observed locations that she has already visited. Our work draws from (and further refines) Ramirez and Geffner's [16, 17] novel and principled work on goal recognition using automated planning.

Plan recognition (PR)[1] is the problem of identifying an agent's intent by observing her behaviour. Its growing number of applications include language understanding and response generation [1], adversarial reasoning for games and the military [12], smart homes for the cognitively impaired [18] and human-machine interaction [13]. Traditionally, PR has involved matching a sequence of observations to some plan in a given *plan library*, the "winning" plan being the one that best matches the observations [10, 4]. Given that each plan presumably sets out to achieve a goal, having identified the agent's plan, the observer has implicitly identified her goal [5].

Recent developments in PR dispense with the overhead of a plan library by treating the problem as one of "planning in reverse" [16, p.1778]. This innovation has enabled plan recognition to leverage advances made by the planning community and relies on a key insight (independently arrived at by multiple authors [16, 2]) that the probability of a plan can be linked directly 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* [17] path to goal.

[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.

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

Given that perfectly rational agents pursue their goals via perfectly optimal paths, Ramirez and Geffner first devised a solution that used classical planners to identify a crisp subset of "most-likely goals", consisting solely of those goals for which observations conformed to the optimal path [16]. The inability of the system to accommodate even marginally suboptimal plans represented a significant limitation and, in [17], they presented a more flexible model which, again using classical planners, this time generated a probability distribution across the set of possible goals based on the cost difference between the cheapest available path that conforms to observations and the cheapest path that does not.

In this paper, we examine the probabilistic framework developed in [17] in the context of path-planning. The work is significant because of the many applications where it is useful to reason about an agent's most likely destination: a real-world security detail at an airport observes an agent who may or may not be heading for a particular boarding gate [11]; an intelligent NPC in a real-time strategy game—or the game engine itself, unfolding the game narrative—continually assesses other players' movements as they navigate the game space [9]. In both scenarios, the observer must assess the agent's movements not once but multiple times; in both scenarios, the underlying terrain, or groundplan, is known in advance.

Having imported Ramirez and Geffner's account into the realm of path-planning, we prove formally that a simpler account yields an almost identical result with less computational effort. This is beneficial to the RTS gaming community, for example, where any speed gain is welcome. We set out the case in which the two accounts do differ, which happens to be a case of little relevance.

More surprisingly, we also show that a probability distribution that ranks candidate goals in the same order can be obtained *without referencing any observations* (other than the agent's current location). Being observation-independent—and provided possible entry points (e.g., entries to the terminal) and destinations of interest (e.g., locations to monitor and protect) are known in advance—the rankings can be pre-computed offline and retrieved in constant time. So for example, this finding implies that an airport security system, instead of tracking agents throughout the terminal, can focus resources on those locations where, should an agent appear, there is the greatest probability that she is making for the boarding gate of interest.

The rest of this paper is structured as follows. First, we describe the PR problem in terms of planning, as developed in [17]. We then ground the problem and its solution in the context of path-planning. After that, we present our main technical contributions: a simpler, equivalent formula; and an observation-independent formula. Finally, we provide a brief empirical evaluation using the well-known Moving-AI benchmarks [19] and present our conclusion.

2. GOAL RECOGNITION AS PLANNING

In this section, we review the "domain theory" approach to PR pioneered by Ramirez and Geffner.

Goal recognition is a well-established problem and approaches to its solution have ranged from Bayesian networks [4] through hidden Markov models [3] to tree grammars [6]. In most cases, observations are matched against potential sequences of actions stored as a plan library. Despite the obvious relationship between plan recognition and planning, Ramirez and Geffner were among the first to make an explicit link [16]. Instead of matching observations to pre-existing plans, they propose the use of a planner to generate new plans that incorporate the observations: by creating and costing two plans for each in a set of potential goals, it is possible—assuming the agent is rational—to assess which of the goals she has set out to achieve.

We now take a more detailed look at their approach. Using STRIPS notation, a ***planning domain*** is a tuple $D = \langle F, A\rangle$, where F is a set of fluents and A is a set of actions a, each of which has a precondition, add and delete list $Pre(a)$, $Add(a)$ and $Del(a)$, all subsets of F. An action a can occur in a state s if $Pre(a) \subseteq s$. The initial state is assumed to be fully observable and the domain is deterministic. If action a occurs in state s, a new state s' results such that $s' = (s \cup Add(a)) \setminus Del(a)$.

A ***planning problem*** $\langle F, I, A, G\rangle$ is a planning domain with a specified initial state $I \subseteq F$ and goal state $G \subseteq F$, and its solution is a plan $\pi = a_1, \ldots, a_k$ that maps I to G. Typically, each action has a cost $c(a)$. The cost of a plan is defined as $cost(\pi) = \sum c(a_i)$ and an ***optimal plan*** is a solution with the lowest cost.

Building on this framework, Ramirez and Geffner [16] articulate the ***PR problem*** in planning terms as a tuple $T = \langle D, \mathcal{G}, I, O\rangle$, where:

- $D = \langle F, A\rangle$ is a planning domain;
- $\mathcal{G} \neq \emptyset$ is a set of possible goal states;
- I is the initial state; and
- $O = o_1, \ldots, o_k$, where $k \geq 0$ and $o_i \in A$, is a sequence of observations.

The solution to T is a set of goals, the optimal plans for which satisfy the observation sequence. A plan $\pi = a_1, \ldots, a_n$, they say, satisfies the observations $o_1, \ldots, o_m$ if it ***embeds*** them in such a way that the order of actions is preserved; that is, there must be a monotonic function f that maps the observation indices into the action indices such that $a_{f(j)} = o_j$. Any goal for which there is an optimal plan that meets this criterion is part of the solution set.

A major drawback of the PR framework proposed in [16] is that it can only identify a goal if the *observations conform to an optimal path*. Realistically though, agents behave suboptimaly, so rational behavior should only be assumed as a guiding principle and, in [17], Ramirez and Geffner present a modified framework that accommodates suboptimality. This is a *probabilistic* approach to the PR problem and the one upon which we will focus.

A ***probabilistic PR problem (PPR)*** $\langle P, \mathcal{G}, I, O, Prob\rangle$ is a PR problem, as above, plus a prior probability distribution *Prob*. The solution amounts now to a *posterior probability distribution* which prefers those goals whose plans "best satisfy" the observations [17], as determined by the principle of rationality.

To capture this idea, one of the main insights of Ramirez and Geffner is to base the probability of a goal on the ***cost difference*** between the cheapest plan that can reach the goal, given the observed actions already taken, and the cheapest plan that could have reached the goal, had the agent behaved differently (i.e., had the agent *not* displayed the observations already seen). The costs of these optimal plans are given as $optc(G, O)$ and $optc^{\neg}(G, O)$, respectively.[2] Formally, cost difference is a function $costdif_{RGT} : 2^F \times A^* \mapsto \mathbb{R}$ defined as follows:[3]

$$costdif_{RGT}(G, O) = optc(G, O) - optc^{\neg}(G, O). \qquad \text{(RG1)}$$

Next, by comparing cost differences for all $G \in \mathcal{G}$, the authors propose generation of a probability distribution across $\mathcal{G}$—a solution to the PPR task—with the following important property: *the lower the cost difference for a particular goal, the higher its probability*. Concretely, they propose the assumption of a Boltzmann distribution and take $P(O|G) = sigmoid(\beta(optc^{\neg}(G, O) - optc(G, O)))$ [15], thus arriving at the following distribution:

$$P_X(G|O) = \alpha \frac{e^{-\beta X}}{1 + e^{-\beta X}}, \qquad \text{(RG2)}$$

where $X = costdif_{RGT}(G, O)$, α is a normalising constant across all goals, and β a positive constant.[4] The resulting distribution is an "order of magnitude approximation" that conforms to the authors' intuition, stated above, that goals with the lowest cost difference have the highest probability. The β parameter 'modulates' the assumption that the observed agent is pursuing plans sensitive to the same cost function used by the observer: if $\beta > 1$, slight deviations from optimality are penalized; for β between 0 and 1, the observer is more skeptical w.r.t. the rationality assumption.[5] Additional justifications for this choice can be found in [17].

As one can observe, this framework moves the focus from plan libraries to declarative goals (and a model of the environment). The key is that Equation (RG1) above can actually 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 can be encoded back into the planning task. Roughly speaking, if a_i is the i-th observation, a new fluent p_{a_i} is added which becomes true only when action a is executed in a state where $p_{a_{i-1}}$ holds true (i.e., a_i is observed after a_{i-1} has been observed). Then, taking a_k as the last observation, one can plan (optimally) for $G \cup \{p_{a_k}\}$ and $G \cup \{\neg p_{a_k}\}$ to extract $optc(G, O)$ and $optc^{\neg}(G, O)$, respectively.

Eg for this

Using this approach, an agent reasoning about another agent's intention must perform two planning tasks for each potential goal in order to compute the probability distribution described in Equation (RG2). These planning tasks are arguably more complex than merely planning for each goal G, as they embed the behaviour observed so far. In addition, because observations change over time as the observed agent acts, all such planning tasks need to be re-done, every time a new observation is obtained.

In what follows, we investigate this principled account for PPR in the context of path-planning; and we show that simpler and less demanding formulas yield the same outcome, at least in the path-planning context.

---

[2] We have modified the notation from [17] to improve legibility. In [17], $cost(G, O)$ or $c(G, O)$ was used to denote the optimal cost compatible with observations O (we use $optc(G, O)$), and $cost(G, \overline{O})$ or $c(G, \overline{O})$ was used to denote the optimal cost provided *without observing* O. Technically, the latter concepts do not amount to applying the same function to different arguments, but rather a different function to the same arguments, which we denote as $optc^{\neg}(G, O)$.

[3] *RGT* denotes **R**amirez and **G**effner account for **T**ask planning.

[4] This formulation appeared in the code referenced from [17] and is provably equivalent to the account given in the paper and in [15].

[5] Personal communication and [15, p.63].

3. THE PATH-PLANNING CASE

In this section, we import Ramirez and Geffner's approach to goal recognition, as described above, to the particular context of the path-planning (or path-finding) problem, that is, the problem of navigating from an initial location to a final destination in some map or "model" of the world, often expressed as a graph [8].

In a path-planning domain, fluents are tied to locations in a functional manner (e.g., $at(x)$ denotes the agent being in location x) and actions relate to (legal) movements. Typically, the domain is deterministic. Whereas in task-planning, as we have seen, a solution plan is described as a sequence of actions, in path-planning, the solution can be specified as a *path*: a sequence of connected nodes in a graph.

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

- N is a non-empty set of nodes (or locations);
- $E \subseteq N \times N$ is a set of edges between 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 $\pi = n_0, n_1, \dots, n_k$ such that $(n_i, n_{i+1}) \in E$, for each $i \in \{0, 1, \dots, k-1\}$. 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 $\pi(i, j) = \pi^i, \pi^{i+1}, \dots \pi^j$ 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 $\Pi(n_1, n_2)$.

As in task-planning, an instance of a path-planning problem operates in a domain and includes initial and goal states.

Definition 2. A ***path-planning problem*** is a tuple $\langle \mathcal{D}, s, g\rangle$, 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 $\pi^{|\pi|} = g$; the set of all of them being $\Pi(s, g)$. An ***optimal path*** is a solution path with the lowest cost among all solution paths. We use $\Pi^*(s, g)$ to represent the set of all such optimal paths. The ***optimal cost*** between two location nodes is the cost of an optimal path between them.

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, just as we have described the embedding of observations in the action sequence of a PR problem. That is, given a path π and a sequence of waypoints $W = w_0, w_1, \dots, w_k$, where $w_i \in N$, we say that π ***embeds waypoints*** W, if there exists a monotonic function f mapping waypoint indices into path indices such that $\pi^{f(i)} = w_i$. The ***optimal cost*** of a path from n_i to n_j ***via waypoints*** W—that is, the cheapest path possible, given that the waypoints must be embedded—is denoted by $optc(n_i, W, n_j)$. If $W = \emptyset$, we just write $optc(n_i, n_j)$, and if $\pi^0 = w_0$ and $\pi^{|\pi|} = w_k$, we write $optc(W)$, that is, the optimal cost through the waypoints themselves. We generalize the set of all solution paths $\Pi(s, g)$ to those embedding waypoints W as $\Pi(s, W, g)$. Similarly, $\Pi^*(s, W, g)$ will denote those paths that are, moreover, optimal w.r.t. cost among paths in $\Pi(s, W, g)$.

3.1 Cost-based goal recognition

With the path-planning framework in place, let us now reframe Ramirez and Geffner's probabilistic PR problem in a path-planning context. Whereas in task-planning, we seek to determine from observations of the agent "what is she doing?" in path-planning, we are trying to discover "where is she going?"

Definition 3. A ***path-planning goal recognition (PPGR) problem*** is a tuple $\mathcal{P} = \langle \mathcal{D}, \mathcal{G}, s, O, \mathit{Prob}\rangle$, 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;
- $O = o_1, \dots, o_k$, where $k \geq 0$ and $o_i \in N$, is the sequence of observations, with $o_1 \neq s$; and
- *Prob* represents the prior probabilities of the goals (though, in common with [17], we assume in this discussion that priors for all goals are equal).

The solution to a PPGR problem $\mathcal{P}$ is a probability distribution across $\mathcal{G}$ which we obtain by comparing, for each goal, a re-working of the cost difference formula (RG1), grounded in path-planning as the function $costdif_{RGP} : N \times N \times N^* \mapsto \mathbb{R}$, defined as:[6][7]

$$costdif_{RGP}(n_1, n_2, O) = optc(n_1, O, n_2) - optc^{\neg}(n_1, O, n_2), \tag{RG3}$$

where:

- n_1 and n_2 are the start and goal locations of interest;
- $O = o_1, \dots, o_k$ is the sequence of waypoint locations, not actions, representing where the agent has been observed (note that O is not a path; nodes need not be adjacent);
- $optc(n_1, O, n_2)$ is, as defined above, the optimal cost of navigating from location n_1 to location n_2 via waypoints O; and
- $optc^{\neg}(n_1, O, n_2)$ denotes the optimal cost of navigating from location n_1 to location n_2 without embedding the observed waypoints, which is defined as follows:

$$optc^{\neg}(n_1, O, n_2) = \min_{\pi \in \Pi(n_1, n_2) \setminus \Pi(n_1, W, n_2)} cost(\pi).$$

Finally, given a PPGR problem $\mathcal{P} = \langle D, \mathcal{G}, s, O, \mathit{Prob}\rangle$, a probability distribution can be derived exactly as for task-planning, by taking $X = costdif_{RGP}(s, g, O)$ into Equation (RG2). For legibility, we shall call this resulting function P_{RGP}, namely:

$$P_{RGP}(g|O, s) = \alpha \frac{e^{-\beta\, costdif_{RGP}(s,g,O)}}{(1 + e^{-\beta\, costdif_{RGP}(s,g,O)})}. \tag{RG4}$$

In words, $P_{RGP}(g|O, s)$ stands for the probability that the observed agent is travelling to goal $g \in \mathcal{G}$, relative to PPGR problem $\mathcal{P}$, when the observed waypoint sequence is O. Note that the initial node s comes from the problem $\mathcal{P}$ itself. In common with Equation RG2, this formula has the property that the lower the cost difference, the higher the probability.

This concludes the reformulation of Ramirez and Geffner's account for goal recognition into a strict path-planning context.

[6]Observe that we now make explicit the initial starting point n_1, which was taken as implicit in Equation (RG1).

[7]*RGP* denotes **R**amirez and **G**effner account for **P**ath-planning.

4. A SIMPLER COST DIFFERENCE

In this section we take a closer look at the cost difference formula (RG3) with the aim of making it simpler and faster to calculate. To that end, consider an agent using $P_{RGP}(\cdot)$, as above, to reason about another agent's travel. Given Equation (RG3), the first agent must perform two planning tasks for each goal g in $\mathcal{G}$: one to extract $optc(s, O, g)$ and one to extract $optc^{\neg}(s, O, g)$. Furthermore, in a typical application, such as during an RTS game or while conducting surveillance, the cost is not incurred once only: the calculation must be repeated, and the accompanying time-hit sustained, for every goal, every time the observed waypoint sequence O is extended by the addition of a new observation.

As with task-planners, path-planning systems have no off-the-shelf means of handling observation requirements. Furthermore, having a less general representation, it is not possible to encode such (positive or negative) requirements back into the input of the problem as occurs in [17] for goal recognition in task-planning. We can only 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 therefore propose an alternative formulation whereby, instead of calculating and deducting optimal cost "given not the observations", we simply deduct the more readily available "optimal path cost". Formally:

optc(G,O)

$$costdif_1(s, g, O) = optc(s, O, g) - optc(s, g). \tag{1}$$

This alternative formulation is conceptually simpler *and* computationally less demanding, in that there is no requirement to reason negatively about the observations. What is more, since the optimal path cost to each potential goal in $\mathcal{G}$ is not dependent on the observations, it can be pre-computed once at the outset. Note that 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 $optc(s, g)$ 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 [17, p.1123] for general goal recognition. It turns out though that the differences between Equations (RG3) and (1) (and their actual impact) were not well understood. In what follows then, we demonstrate not only that (1) is simpler and easier to compute, but that it provides an identical result in all cases bar one in the context of navigation; 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, (1) actually enables calculation of the posterior probability distribution when the original, more involved, formula (RG3) fails.

From now on, we assume a PPGR problem of the form $\mathcal{P} = \langle \mathcal{D}, \mathcal{G}, s, O, Prob \rangle$, as per Definition 3.

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 PR framework* [17] (as compared to the previous framework in [16]). We argue, in fact, that this case represents the most common case: it is usual to expect some noise and, in the presence of any noise at all, there is at once some cheaper way that the agent might have proceeded so the observed path is usually suboptimal.

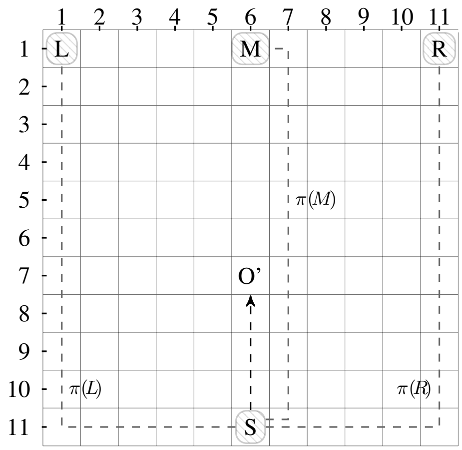

Figure 1: O' is on an optimal path to all three goals but, by the more complex formula, M is the most probable.

THEOREM 1. *Let O be an observation sequence such that $optc(s, O, g) > optc(s, g)$ (i.e., the observed behaviour is not optimal). Then, $costdif_{RGP}(s, g, O) = costdif_1(s, g, O)$.*

PROOF. Let π^* be an optimal solution path from s to g, that is, $\pi^* \in \Pi^*(s, g)$. Then, $cost(\pi^*) = optc(s, g)$. We can then conclude that path π^* does *not* embed O, otherwise, we would have $optc(s, O, g) = optc(s, g)$. Hence, since π^* does not embed O and is optimal among all solution paths, we get $optc^{\neg}(s, O, g) = cost(\pi^*) = optc(s, g)$, and since $optc^{\neg}(s, O, g) = optc(s, g)$, $costdif_{RGP}(s, g, O) = costdif_1(s, g, O)$ follows. $\square$

In words, if the observed path is suboptimal—as, arguably, it would be most of the time—the simpler formula (1) yields *exactly the same* value as the original formula (RG3).

Optimal paths (non-exclusive). We now consider the case in which observations 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 [7] (see Example 2). 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 such that $optc(s, O, g) = optc(s, g)$ (i.e., the observed behaviour is optimal). If $\Pi^*(s, g) \setminus \Pi^*(s, O, g) \neq \emptyset$, then $costdif_{RGP}(s, g, O) = costdif_1(s, g, O)$.*

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, $cost(\pi') = optc(s, g)$, and since it does not embed O we can conclude that $optc^{\neg}(s, O, g) = cost(\pi')$. Thus, $optc^{\neg}(s, O, g) = optc(s, g)$, and $costdif_{RGP}(s, g, O) = costdif_1(s, g, O)$ follows. $\square$

In words, even if the observed behavior is fully rational, if there are other ways of behaving rationally, again the simpler formula (1) is *exactly equivalent* to the original formula (RG3).

Optimal paths (exclusive). We now consider the only situation in which cost difference Equations (RG3) and (1) return *different* results. Ramirez and Geffner provided the following example to illustrate the case.

EXAMPLE 1. *Consider the situation depicted in Figure 1. 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 O track directly north through O' and satisfy an optimal path to all three goals. In the case of L and R, there are multiple 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, $costdif_{RGT}(L, O) = 0$ and $costdif_{RGT}(R, O) = 0$ (see Equation 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, $optc(M, O) = 10$, whereas $optc^{\neg}(M, O) = 12$. Thus, $costdif_{RGT}(M, O) = -2$. The lower the cost difference, the higher the probability, making M the most probable goal.*

Although cited as an "example" of the distinction between the two cost difference formulas, this scenario, in fact, represents the *only* distinction. We therefore now strengthen Theorem 2 by proving that the two cost difference formulations yield identical results in all cases bar one: when observations are not only sufficient for optimal behavior, but also *necessary*. In this case we say that the observations are ***exclusively optimal***: there is no other way of acting fully rationally.

THEOREM 3. *Let O be an observation sequence and $g \in \mathcal{G}$. Then, $costdif_{RGP}(s, g, O) \neq costdif_1(s, g, O)$ iff $\Pi^*(s, O, g) = \Pi^*(s, g)$ (i.e., all optimal paths embed the observations).*

PROOF. The (ONLY-IF) follows directly from Theorem 2. For the (IF) direction, suppose that $\Pi^*(s, O, g) = \Pi^*(s, g)$, that is, the optimal paths and the optimal paths embedding the observations coincide. 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 $\Pi^*(s, O, g) = \Pi^*(s, g)$, it follows that $\pi \notin \Pi^*(s, g)$. Given that $\pi \in \Pi(s, g)$, we get that $cost(\pi) > optc(s, g)$. Since path π was arbitrarily chosen, $optc^{\neg}(s, O, g) > optc(s, g)$, and $costdif_{RGP}(s, g, O) \neq costdif_1(s, g, O)$ follows. □

As already noted, this is a corner case. Furthermore, it is arguably irrelevant to the expected suboptimal behavior that the probabilistic PR framework [17] was designed to handle.

Regardless of how relevant or interesting this corner case may be, let us further investigate its implications. We remind the reader that, following [17], 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) 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 order—or *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 $f(g, O)$ be some (cost difference) function and let P_X be a template of the probability distribution defined in* (RG2). *If $f(g_1, O) < f(g_2, O)$ then $P_f(g_1|O) > P_f(g_2|O)$.*

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 in [15] and [17].

Let $P_{RGP}(\cdot)$ be the probability distribution obtained from (RG2) when $X = costdif_{RGP}(s, g, O)$ (Equation (RG3)) and $P_1(\cdot)$ the distribution obtained when $X = costdif_1(s, g, O)$ (Equation (1)).

THEOREM 4. *Let O be an observation sequence and suppose that, for some potential goal $g \in G$, it is the case that:*

1. *$\Pi^*(s, O, g) = \Pi^*(s, g)$, that is, observations are exclusively optimal; and*
2. *for every $g' \in \mathcal{G} \setminus \{g\}$, $optc(s, O, g') > optc(s, g')$, that is, observations would result in suboptimal paths to all the other possible goals.*

Then, for all $g_1, g_2 \in \mathcal{G}$ and $g_1 \neq g_2$, it is the case that $P_1(g_1|O) > P_1(g_2|O)$ if and only if $P_{RGP}(g_1|O) > P_{RGP}(g_2|O)$.

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

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

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 O is necessary to travel from s to g in any optimal way (first assumption), 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 (RG3) we get that $costdif_{RGP}(s, g, O) < 0$. 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
- $costdif_{RGP}(s, g, O) < 0 < costdif_{RGP}(s, g', O)$.

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

So, even in this corner case, using the simpler cost difference formula lets us determine the same ranking among potential goals and hence is sufficient to identify the "most probable" goal.

There remains one variation of exclusive optimality that we have so far excepted. It is the case where observations coincide with the only optimal paths to *multiple* goals (rather than one). We have argued that exclusive optimality for one goal is unusual; clearly, for multiple goals, it is even more so. Should the situation arise, however, the complex formula (RG3) would return multiple (negative) cost differences, which could be ranked, whereas the simple formula (1) ranks all goals for which observations match the optimal path equally. Arguably, however, this situation is not only extremely unlikely, it also concerns *the very set of goals in which this probabilistic account [17] is least interested.*

Why use the more complex formula?

Our discussion so far supports the use of $P_1(\cdot)$ rather than $P_{RGP}(\cdot)$ for reasoning about the goal of an observed agent: it is conceptually simpler, computationally less demanding and, except in one unlikely circumstance, yields the same (practical) outcomes. We note, however, that Ramirez and Geffner make a case for preferring the more demanding formula (RG1) [17], which we now consider.

A better predictor. To arrive at a probability distribution, their account appeals to Bayes' Rule, $P(G|O) = \alpha P(O|G)P(G)$ where

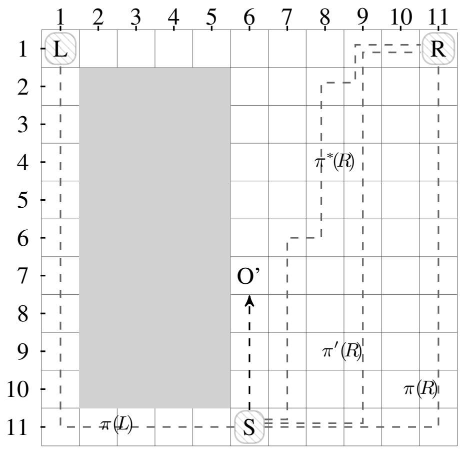

Figure 2: $P(L|O')$ is the same as $P(R|O')$.

α is a normalising constant. Prior probabilities in $P(G)$ are given, so the challenge is to account for $P(O|G)$. Example 1 demonstrated that $costdif_{RGP}(s, g, O)$ only differs from $costdif_1(s, g, O)$ when observations match the *only* optimal way to goal and it is correct, the authors say, that $P(O|M)$ should exceed $P(O|L)$ and $P(O|R)$ because goal M better predicts the observations.

The intuition behind this account seems to be that probabilities are somehow linked to the number of available paths to goal; that, if there had been four optimal paths and the agent was seen on one of them, the probability of the goal would be correspondingly lower; 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 that intuition.

EXAMPLE 2. *Consider a domain with two goals, L and R as depicted in Figure 2. The rectangular block of cells, $(2, 2)$ to $(5, 10)$, is not navigable. Now, instead of one optimal path to goal M (which has been removed), there are two optimal paths to goal L. Meanwhile, (owing to the notorious symmetry of gridworlds) there are 3003 optimal paths to R, as before. Equation (RG3), however, can no more distinguish between L and R than can the simpler formula (1). In this scenario, $P(O|L)$ should, if we are counting paths, be very much greater than $P(O|R)$ but, for both goals, both formulas return zero and, by the posterior probability calculation (RG4), both goals appear to be equally probable.*

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) and Equation (RG3), as Ramirez and Geffner [17] point out, returns $-\infty$. Given that the model hinges on the notion that the lower the cost difference, the higher the probability, this result should give g the highest possible probability within the distribution. However, in order to normalise the probabilities, each must be divided by their *sum*. Where one probability is infinite, the sum is infinite and therefore, using (RG3), normalisation fails (division by infinity) and probabilities cannot be calculated for any goal in the domain.

Thus, what seemed to be part of the case for the (RG3) formula turns out to be part of the case against after all. In an identical situation, Equation (1) (based on optimal cost from start to goal rather than "optimal cost given not the observations") returns zero and a complete distribution, clearly favouring g, is successfully returned.

Summary

To conclude this section, we summarise the case for formula (1):

- the simpler formula $costdif_1(s, g, O)$ returns the same result as the more complex $costdif_{RGP}(s, g, O)$ in all cases bar one;
- the one case where the formulas do not return identical results relates to fully rational agent behavior, which is precisely *not* the motivation of the probabilistic PR model;
- even when $costdif_1(s, g, O)$ returns a different result from $costdif_{RGP}(s, g, O)$, it is unlikely to impact the overall probability distribution;
- in the most extreme case when $costdif_1(s, g, O)$ returns a different result from $costdif_{RGP}(s, g, O)$—because there is no alternative path to the goal without the observations—it may return *no* result at all; and
- $costdif_1(s, g, O)$ is computationally advantageous: it does not require 'negative' reasoning about observations (which implies any standard path-planner can be used off-the-shelf); and in certain domains, its second term may be pre-computed.

5. OBSERVATION-FREE RECOGNITION

We have seen above that, in all but one extreme corner case, the simpler formula (1) can be used interchangeably with the more complex formula (RG3). In this section, we go further and prove that the ranking among potential goals, as judged by the probability distribution $P_1(\cdot)$ generated using cost difference (1), can be achieved *without reference to the observation sequence*.

At first sight, the finding is counter-intuitive. Indeed, it implies that we can perform goal recognition without observing how the agent behaves over time! Nevertheless, if we know an agent's start location (e.g., because it is one of a finite number of building entrances) and the location of each candidate goal, we require only the agent's *current* location in order to calculate a probability distribution within which the goals will be ranked in *exactly the same order* as if we had used formula (1).

Our observation-independent cost difference formula, $costdif_2$: $N \times N \times N \mapsto \mathbb{R}$ is defined as:

$$costdif_2(s, g, n) = optc(n, g) - optc(s, g), \qquad (2)$$

where n stands for the location last observed (i.e., $n = O^{|O|}$), that is, the current or most recently observed location of the agent whose destination we are trying to determine. Let $P_2(\cdot)$ be the probability function obtained by taking $X = costdif_2(s, g, n)$ in (RG3).

THEOREM 5. *Let O be an observation sequence. For all $g_1, g_2 \in \mathcal{G}$, $P_1(g_1|O) > P_1(g_2|O)$ iff $P_2(g_1|O) > P_2(g_2)$.*

PROOF. From Observation 1, $P_1(g_1) > P_1(g_2)$ if and only if $costdif_1(s, g_1, O) < costdif_1(s, g_2, O)$. Recall, from Equation (1), that for each $i \in \{1, 2\}$:

$$costdif_1(s, g_i, O) = optc(s, O, g_i) - optc(s, g_i),$$

where the first term can be written as:

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

Let $n_l = O^{|O|}$ be the last observation in O. From Observation 1, recall that the relative ranking between g_1 and g_2 w.r.t. their posterior probabilities can be deduced directly from the relative value of their cost difference formulas. So, let us expand that value:

$$\begin{aligned}
&costdif_1(s, g_1, O) - costdif_1(s, g_2, O) \\
&\quad = [optc(s, O^0) + optc(O) + optc(n_l, g_1) - optc(s, g_1)] - \\
&\qquad [optc(s, O^0) + optc(O) + optc(n_l, g_2) - optc(s, g_2)] \\
&\quad = optc(s, O^0) + optc(O) + optc(n_l, g_1) - optc(s, g_1) - \\
&\qquad optc(s, O^0) - optc(O) - optc(n_l, g_2) + optc(s, g_2) \\
&\quad = optc(n_l, g_1) - optc(s, g_1) - optc(n_l, g_2) + optc(s, g_2) \\
&\quad = [optc(n_l, g_1) - optc(s, g_1)] - [optc(n_l, g_2) - optc(s, g_2)] \\
&\quad = costdif_2(s, g_1, n_l) - costdif_2(s, g_2, n_l).
\end{aligned}$$

It follows then that $costdif_1(s, g_1, O) > costdif_1(s, g_2, O)$ iff $costdif_2(s, g_1, O) > costdif_2(s, g_2, O)$. Thus, from Observation 1, $P_1(g_1|O) > P_1(g_2|O)$ iff $P_2(g_1|O) > P_2(g_2)$. □

The finding is useful and unexpected. All parameters are independent of the observation sequence (modulo wherever the agent is "*now*") and can be obtained by calls to any *standard* path-planner: no specialised path-finding system is needed to reason about observations. Furthermore, if all the start and candidate goal locations are known—as would typically be the case in most domains—formula $costdif_2(s, g, n)$ can be fully *pre-computed offline* for any node $n \in N$ in the domain.

The implications are significant. It means that we can create a sort of "heat map" of the domain, showing the probability of each goal according to where the agent entered. If we have a particular goal of interest (e.g., a valuable location to monitor and protect), we can focus our attention fully on the *locations where that goal becomes the most probable*. That is, rather than tracking an agent's movements all over the terrain, we can just monitor the "hot" spots and only start tracking her in earnest if she arrives at one of them.

We close the section by noting that the above result is actually not dependent on formula (RG4) 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.

6. EXPERIMENTAL SETUP AND RESULTS

In this section, we report our results on the performance of goal recognition in path-planning when using the original (complex) cost difference (RG3), the simpler version (1) that does not reason negatively about observations, and the observation-free version (2) in problems adapted from the well-known Moving-AI[8] path-planning benchmarks [19]. The aim was to develop an experimental framework for the problem of goal recognition in path-planning to empirically confirm that *(a)* the case of observations conforming to the *only* optimal path to a goal (as in Theorem 3) is rare and, otherwise, the simpler formula yields identical posterior probability distributions; *(b)* all three accounts return posterior probability distributions that rank goals the same; and *(c)* use of either of the latter two formulas cuts processing time by more than half.

6.1 Experimental setup

We generated an initial problem set of 990 individual problems from a base set of 60 scenarios selected at random from two sets of Moving-AI benchmarks [19]:[9] 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* [14] 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 (sparse 20%, medium 50%, dense 80%) and two observation distributions (*random* selects random locations along the path, *prefix* selects a consecutive sequence of location nodes from the start location).

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 [8][10]. To obtain the cost of an optimal path given *not* the observations, inspired by the technique in [17], 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 grid-type settings, where all horizontal and vertical moves were costed at 1, and all diagonal moves at 1.414. Like [17], we assumed equal goal priors and calculated the probability of each goal using Equation (RG4).

6.2 Results

Our results confirmed the hypotheses. In maps representing relatively open landscapes, the formulas performed exactly as predicted: formulas (RG3) and (1) returned identical results, all three formulas ranked goals identically by probability and (after the first iteration in a domain with the same start location and goals) formula (RG3) was half the speed of formulas (1) and (2).

In the single-pixel maze, again as predicted, the implementation based on formula (RG3) 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.

Importantly, 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 (RG3) returned 0.329, 0.342, 0.329 (for L, M and N in Figure 1, respectively), whereas formulas (1) and (2) returned 0.333 for all three goals.

Tables 1 and 2 summarise our results for suboptimal observation sequences.[11] Column O displays the percentage of nodes from the full path that were included in the observation sequence: P indicates that the observations were presented as a continuous path prefix, and R that they were randomly drawn from the length of the path. Column T displays the average time-taken per goal recognition problem. Column M shows the percentage of probability

[8] http://movingai.com/

[9] Experiments were conducted on a i7 1.8GHz dual core with 8GB RAM in a Linux environment.

[10] 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).

[11] To save space—and to avoid repetition—we have not tabulated the greedy or optimal results but have discussed them inline.

Table 1: Rooms - suboptimal

O	P_{RGP} T	P_1 T	P_1 M	P_1 Δ	P_2 T	P_2 M	P_2 Δ
20P	30.915	6.795	100%	0	3.182	37.0%	0.024
20R	13.916	2.749	100%	0	2.688	27.5%	0.313
50P	32.001	2.439	100%	0	2.428	27.5%	0.097
50R	21.077	2.755	100%	0	2.698	27.5%	0.337
80P	34.970	2.465	100%	0	2.425	24.1%	0.242
80R	29.980	2.775	100%	0	2.712	27.6%	0.340

180 problems. Average goals: 4.8. Average path cost: 362.

Table 2: Landscapes - suboptimal

O	P_{RGP} T	P_1 T	P_1 M	P_1 Δ	P_2^* T	P_2^* M	P_2^* Δ
20P	24.596	3.835	100%	0	1.780	34.6%	0.040
20R	10.346	2.112	100%	0	2.062	53.8%	0.025
50P	31.934	1.706	100%	0	1.681	38.5%	0.034
50R	18.962	2.110	100%	0	2.073	57.7%	0.024
80P	33.274	1.876	100%	0	1.848	42.3%	0.035
80R	26.073	2.111	100%	0	2.062	57.7%	0.023

156 problems. Average goals: 4.35. Average path cost: 280.74. We obtain P_2^* as P_2 but adding a constant to the corresponding cost difference (see discussion inline).

distributions that exactly matched those generated using cost difference formula (RG3). Where a difference was recorded, column Δ displays the average difference for those cases. On average, the implementation using cost-difference (RG3) performed even more slowly than we expected. In the room layouts, as shown in Table 1, it was four to 12 times slower than using the simple formula. This is, in part, explained by outliers. For example, we found the complex implementation slow to identify an alternative optimal path (i.e., where cost difference returned zero). 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 is 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 in [17] (which most closely resemble our experiments, though in the context of planning which, we appreciate, is necessarily more demanding) took, on average, over three minutes to complete problems with observation densities of 50% using an optimal planner comparable to A*, on problems with average optimal path lengths of just 17 steps. Ramirez and Geffner improved performance by using a suboptimal planner. We did try a suboptimal—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 formula (1) cut processing time, in some cases returning a result in a twelfth of the time taken by equivalent calls to formula (RG3). 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 of the minimal formula for that problem.

Whereas the probabilities based on cost difference (1) always exactly matched those based on cost difference (RG3), probabilities generated using formula (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. Equation (RG4) uses exponential values to generate posterior probabilities and so is particularly sensitive to any small variation. Furthermore, although the relative cost differences always exactly match those of the simpler formula (as proved in Theorem 5) the actual cost differences are almost always negative and sometimes (on map problems involving paths of 1000 or more steps) vastly so. This translates, when used as a cost difference parameter in probability distribution formula (RG4) to negative exponentials, that is, tiny fractions. As a result, the output loses precision and the posterior probability distribution tends to equalise across multiple goals.

This accounts for the sometimes higher than expected delta values (see Table 1) but is easily rectified by adding a large constant to the function's output, raising it always above zero whilst preserving relative rank (see Table 2). Furthermore, whether or not the constant is added, in all cases, use of the observation-independent formula successfully identified the same goal as having the highest, or equal highest, posterior probability as either other formula.

7. CONCLUSION

In this work, we have examined techniques for performing probabilistic plan recognition, first introduced in the insightful work of Ramirez and Geffner [17], and applied them in the context of path-planning. In particular, we focused on how a cost-based approach, which they pioneered, translates to the path-planning realm.

We have shown that a simpler cost difference formula (1) (which does not require reasoning negatively about observations and is achieved by standard calls to a standard path-planner) returns an *identical result* to the original formula (RG3) (which reasons about optimal path cost "given not the observations") in *all but one specific case*, which we characterize. We argue, in line with intuitions expressed in [17], that this is a case of little interest and, in fact, it is one that did not even arise in our automated tests.

Further, we have presented an alternative cost difference formula (2) that *does not depend on the observation sequence* but 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 (RG3) in all cases bar one. This formula requires similar computational effort to the simpler formula (two calls to a standard path-planner for each goal but without having to consider waypoints) and has the benefit that it can be pre-computed in many realistic domains, namely, whenever the start and potential goals are known at the outset (e.g., doors to a building and rooms that require protection). This amounts to a finding that the relative probability of possible goals is a function of the domain. So one can create a sort of "heat map" of posterior goal probabilities from which to identify the perimeter that should be monitored around any goal of interest, knowing that, at all locations within the perimeter, that goal is ranked highest (i.e., it has become the most probable).

We close by noting that our ability to derive an observation-independent formula demonstrates (at least with regard to the *ordering* of goals) that the "cost difference" method of goal recognition is inherently Markovian. We note also that, while reasonable, our account inherits the limitation of assuming that the observed agent is rational.

Acknowledgements

We thank the reviewers for their suggestions and acknowledge the support of the Australian Research Council (DP120100332) and Australian Government Research Training Program Scholarship.

REFERENCES

[1] J. F. Allen, B. W. Miller, E. K. Ringger, and T. Sikorski. A robust system for natural spoken dialogue. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 62–70, 1996.

[2] C. Baker, R. Saxe, and J. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[3] H. H. Bui. A general model for online probabilistic plan recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 3, pages 1309–1315, 2003.

[4] E. Charniak and R. Goldman. Probabilistic abduction for plan recognition. Technical report, Brown University and Tulane University, 1991.

[5] R. Demolombe and E. Hamon. 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)*, pages 905–911. ACM Press, 2002.

[6] C. W. Geib and R. P. Goldman. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11):1101–1132, 2009.

[7] D. D. Harabor and A. Grastien. The JPS pathfinding system. In *Proceedings of the Annual Symposium on Combinatorial Search*, 2012.

[8] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions of Systems Science and Cybernetics*, 4(2):100–107, 1968.

[9] F. Kabanza, P. Bellefeuille, F. Bisson, A. R. Benaskeur, and H. Irandoust. Opponent behaviour recognition for real-time strategy games. *Plan, Activity, and Intent Recognition*, 10:05, 2011.

[10] H. Kautz and J. Allen. Generalized plan recognition. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 32–37, 1986.

[11] S. Keren, A. Gal, and E. Karpas. Goal recognition design. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2014.

[12] A. Kott and W. M. McEneaney. *Adversarial reasoning: computational approaches to reading the opponent's mind*. CRC Press, 2006.

[13] N. Lesh, C. Rich, and C. L. Sidner. Using plan recognition in human-computer collaboration. In *UM99 User Modeling*, pages 23–32. Springer, 1999.

[14] I. Pohl. Heuristic search viewed as path finding in a graph. *Artificial Intelligence*, 1(3-4):193–204, 1970.

[15] M. Ramirez. *Plan Recognition as Planning*. PhD thesis, Univeristat Pompeu Fabra, 2012.

[16] M. Ramirez and H. Geffner. Plan recognition as planning. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1778–1783, 2009.

[17] M. Ramirez and H. Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1121–1126, 2010.

[18] P. C. Roy, A. Bouzouane, S. Giroux, and B. Bouchard. Possibilistic activity recognition in smart homes for cognitively impaired people. *Applied Artificial Intelligence*, 25(10):883–926, 2011.

[19] N. R. Sturtevant. Benchmarks for grid-based pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(2):144–148, 2012.