

Active Goal Recognition Design

Kevin C. Gall¹, Wheeler Ruml¹ and Sarah Keren^{2,3}

¹Department of Computer Science, University of New Hampshire, USA

²School of Engineering and Applied Sciences, Harvard University, USA

³School of Computer Science and Engineering, Hebrew University of Jerusalem, Israel

kcg245@gmail.com, ruml@cs.unh.edu, skeren@seas.harvard.edu

Abstract

In Goal Recognition Design (GRD), the objective is to modify a domain to facilitate early detection of the goal of a subject agent. Most previous work studies this problem in the offline setting, in which the observing agent performs its interventions before the subject begins acting. In this paper, we generalize GRD to the online setting in which time passes and the observer’s actions are interleaved with those of the subject. We illustrate weaknesses of existing metrics for GRD and propose an alternative better suited to online settings. We provide a formal definition of this Active GRD (AGRD) problem and study an algorithm for solving it. AGRD occupies an interesting middle ground between passive goal recognition and strategic two-player game settings.

1 Introduction

Goal recognition is the problem of identifying the goal of an agent from observations of its actions [Sukthankar *et al.*, 2014]. Applications range from educational games [Ha *et al.*, 2011] to adversarial settings [Ang *et al.*, 2017]. In this paper, we study Goal Recognition Design (GRD) [Keren *et al.*, 2014], where the objective is to modify a domain to facilitate early detection of the goal of a subject agent acting to achieve one of a known finite set of goals. While most previous work studies this problem in the offline setting, we propose a variant in which the observing agent performs its interventions online while the subject is acting.

For example, consider a playground with an area that includes a sand pit for toddlers and an area with tall slides that are dangerous for younger children. Now consider an arriving toddler very excited to play and not listening to her caregiver. She might run toward *both* the sand pit and the tall slide, since they are in the same direction. As those with small children know, toddlers can have a strong independent streak. Her caregiver would want to know which was her goal as soon as possible so that they could intervene before she gets to the slide, but take no other action if she is going to the sand pit so as to avoid a tantrum. Conventional offline GRD would require the designers of the playground to plan in advance a wall separating the two areas. In this case, the toddler must

decide which direction she is going in plenty of time for the caregiver to see and react. In the absence of such specific foresight by the designer, the caregiver must step in the middle of the path, forcing the child to go left or right and thus revealing her goal. Clearly, real-world environments do not always have the luxury of offline design efforts to optimize the recognition of agents’ goals.

In this paper, we 1) illustrate limitations of the conventional approaches to GRD, 2) define a formal problem setting for the Active Goal Recognition Design (AGRD) problem, and 3) design and experimentally characterize an optimal algorithm for solving AGRD problems. We find that AGRD captures a useful range of multiagent interactions lying between passive goal recognition and strategic two-player games.

2 Previous Work

Goal Recognition Design (GRD) was first proposed by Keren *et al.* [2014], then extended for sub-optimal agents, partial observability, and partially-informed agents [Keren *et al.*, 2015; Keren *et al.*, 2016a; Keren *et al.*, 2016b; Keren *et al.*, 2020]. The objective of a GRD problem is to minimize worst-case distinctiveness (*wcd*), the length of the longest (or more generally, the cost of the most expensive) possible plan prefix before the subject’s true goal is the only possible goal given the prefix. Possible plan prefixes depend on the setting, but in the simplest case are limited to prefixes of optimal plans. A non-distinct plan prefix is a prefix compatible with multiple goals the agent could be pursuing. Solutions to GRD problems are sequences of interventions in the domain that change the set of plans available to agents and minimize *wcd*.

Wayllace *et al.* [2017] modify GRD to handle stochastic domains (S-GRD), where the subject agent’s actions’ outcomes are not deterministic. They propose a new objective, expected case distinctiveness (*ecd*), to deal with stochasticity. *ecd* is the expected time that an agent’s path will be non-distinct given that the agent is following an optimal policy to its goal. By using an expectation, their algorithm can select an intervention that is expected to reduce the non-distinctiveness of the agent’s path, even if the intervention does not necessarily reduce *wcd*.

There is a growing body of work on observers who use agency in the domain to help reveal the subject agent’s goal more quickly. Kabanza *et al.* [2010] and Bisson *et al.* [2011] explore ‘provoking actions’ to force opponents to reveal their

intentions through their reactions. These works depend on direct cause-effect relationships between provoking actions and opponent reactions. By contrast, our work allows observers to plan longer-term sequences of interventions without immediate effect. Mirsky *et al.* [2018] model plans as context-free grammars with nonterminals that can be refined via productions into primitive actions. They query the subject online about whether a coarse plan can be refined to the correct hypothesis and can thus prune significant portions of the observer’s hypothesis set. However, their observer does not have agency to act in the world. Masters and Sardina [2019] consider adversarial goal recognition, in which the subject attempts to defeat recognition. Their setting is offline and limited to path planning domains. We will discuss the work of Shvo and McIlraith [2020] and Amos-Binks and Cardona-Rivera [2020] in detail in Section 7.

3 GRD for Online Settings

Existing work on GRD optimizes distinctiveness in the form of wcd or ecd . As our first contribution, we analyze these metrics and identify problematic cases. We then discuss an alternative that has advantages in online settings. We also emphasize the importance of modeling the passage of time.

3.1 Objective Functions

One underlying problem with wcd is that its focus on the worst case can cause useful interventions to be overlooked. Consider Figure 1-left, which represents a deterministic domain where the wcd equals the optimal cost to goal A . There are two plan prefixes whose length equals wcd : one that leads to both B and C and becomes distinct at time step $wcd + 1$, and one that leads to goal B and passes through goal A at time step wcd , also becoming distinct at $wcd + 1$. The red circle indicates an intervention that eliminates the plan to B that passes through A . However, the standard GRD objective would ignore this intervention since it does not affect wcd . Clearly if the agent’s goal is A , it is preferable to distinguish it even if the distinctiveness of B and C cannot be reduced. To address this issue, we look to expected case distinctiveness (ecd) [Wayllace *et al.*, 2017]. The ecd of an MDP state s is the expected length of the agent’s non-distinct plan prefix. $ecd(s)$ is defined recursively for all successors of s , weighted on the transition probabilities of the actions leading to the successor and the actions’ likelihood w.r.t. a set of given prior probabilities over possible goals. The base case of ecd is a state for which the goal is uniquely identifiable. Though ecd was designed for stochastic domains, determinism is merely a special case. Let us weight each goal in Figure 1-left equally. Before executing the intervention, $ecd(Start) = \frac{wcd}{3} + \frac{wcd}{3} + \frac{wcd}{3} = wcd$ since all goals become distinct at $wcd + 1$. It is clear that by executing the intervention, $ecd(Start) = 0 + \frac{wcd}{3} + \frac{wcd}{3} = \frac{2}{3}wcd$ since goal A becomes distinct after taking a single step, thus identifying that the intervention is worth executing.

However, ecd still has key flaw: as a measure based on distinctiveness, it is insensitive to the time remaining after the goal is identified. Consider Figure 1-middle, in which there are three possible goals with a priori equal probability: A ,

B , and C . All arrows describe deterministic actions of cost 1. There are two possible interventions: removing a_1 or a_2 . It is possible that an agent pursuing goal A will not reveal its goal until the final action by taking the path $\langle r, s_2, s_4, A \rangle$. The only way to guarantee that all goals will be identified before the subject’s final action is to remove action a_1 . However, removing either a_1 or a_2 will have identical effects on ecd since their removal is symmetrical. A plan to goal A via s_4 is distinct after 3 steps, which is the same as a plan to C via s_5 , despite the fact that A is achieved in 3 steps where C is not. Therefore, a_1 will not be identified as the more urgent intervention despite our intuition that a_1 is better. For offline GRD, this may be acceptable if both can be executed before an agent enters the system. If intervening online, we need to know which intervention to pursue first since we may not have time to execute both prior to the subject transitioning beyond it.

In their work on Active Goal Recognition, Shvo and McIlraith [2020] compute a different objective: the fraction of the subject’s plan that is completed prior to the goal being recognized. Their approach does not directly minimize this metric, but its use is indicative of the properties of a useful online detection system, e.g. one that considers the amount of time the observer will have to utilize the information once the goal is identified. Pursuing this idea, let us define a function ψ that takes a state and a goal and returns the fraction of the plan that has been executed to achieve that goal relative to a start state. Let $ec\psi$ be a function that behaves exactly as ecd except, instead of returning the expected distinctiveness of a state, it returns the expectation of ψ for the first state where we can identify the subject’s true goal. As with ecd , lower numbers are better.

Returning to Figure 1-middle, if we replace ecd with $ec\psi$, it becomes clear that removing a_1 is the best intervention. We can decompose $ec\psi = p_A \times ec\psi_A + p_B \times ec\psi_B + p_C \times ec\psi_C$ where p_X is the probability of Goal X , and $ec\psi_X$ is the expectation of ψ for instances where the subject’s goal is actually revealed to be Goal X . Observe that for a subject pursuing B , the effect of removing a_1 or a_2 is identical. Since all goals are equally likely, we must simply prove that removing a_1 reduces $ec\psi_A$ more than removing a_2 reduces $ec\psi_C$. A subject pursuing A could transition to s_1 in which case their plan is distinct after 1 step, so $ec\psi_A(s_1) = \frac{1}{3}$. Prior to any interventions, the subject could also transition to s_2 in which case the only remaining plan to A is not distinct from a plan to B until the subject executes the action that achieves A , i.e. $ec\psi_A(s_2) = 1$. Assuming that each action that leads to the subject’s goal is equally likely, $ec\psi_A(r) = \frac{1}{2} \times \frac{1}{3} + \frac{1}{2} \times 1 = \frac{2}{3}$. If we remove a_1 , then the plan that is distinct after one step is the only one remaining, so $ec\psi_A(r) = \frac{1}{3}$, a reduction of $\frac{1}{3}$. Repeating this for a subject pursuing C , prior to any interventions we have $ec\psi_C(r) = \frac{1}{2} \times \frac{1}{4} + \frac{1}{2} \times \frac{3}{4} = \frac{1}{2}$. After removing a_2 , $ec\psi_C(r) = \frac{1}{4}$, a reduction of $\frac{1}{4}$. Since the reduction from removing a_1 ($\frac{1}{3}$) is greater than the reduction from removing a_2 ($\frac{1}{4}$), minimizing $ec\psi$ would prioritize a_1 for removal.

$ec\psi$ effectively normalizes both distinctiveness and reaction time across all goals so that the urgency of distinguishing a goal is inversely proportional to the length of the opti-

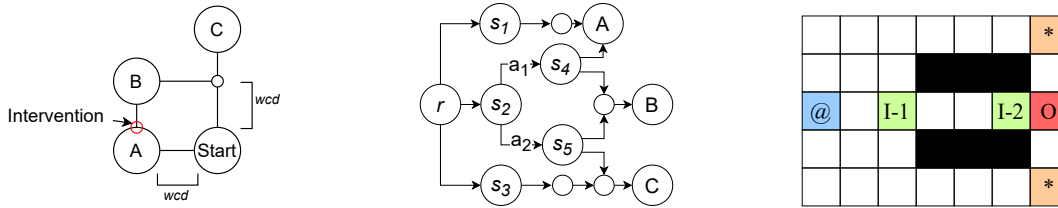


Figure 1: left: wcd misses the intervention; middle: ecd provides no guidance; right: temporal reasoning in AGRD.

mal plan to achieve it. In other words, close goals are prioritized over distant goals. The normalizing effect of ψ provides guidance when interventions have effects of similar magnitude. More weight is given to those that affect close goals because a reduction in time to identify the goal accounts for a larger fraction of the plan to a close goal than a distant one. This guidance is important in online settings with time pressure where we may not have time to execute all meaningful interventions and so must prioritize them.

Ultimately, the choice of objective is domain-specific. For example, if a wildlife photographer wants to know which stream their subject is going to drink from, they may wish to maximize reaction time directly so they have enough time to set up their equipment before the animal gets there. Distinguishing close goals may not be a priority in this case because, if they miss their chance with one animal, there will likely be another possible subject soon. We will define the objective function of our problem setting generically to accommodate varying requirements appropriate to different domains.

3.2 Temporal Reasoning

Our focus on reaction time illustrates a gap in the conventional Active Goal Recognition setting: a formal treatment of the passage of time. If we wish to minimize a metric based on ψ , we need to be able to execute interventions that are still relevant by the time they are carried out. For example, consider Figure 1-right. In this simple Grid-World with 4-way movement, the subject agent (blue) has one of two goals (orange). The observer agent (red) may block any green cell it is adjacent to. The observer follows the same movement rules as the subject, and they can occupy the same cell. We assume the subject follows optimal plans, the observer must not change the optimal cost to any goal, and all actions are immediately observable. By any of the discussed objectives, it is clear that blocking *I-1* would force the subject to reveal its goal more quickly than *I-2*, since the subject must turn left or right after one step. However, under a realistic online model where the subject’s and observer’s actions are interleaved, the observer agent does not have enough time to reach and block *I-1* before the subject enters that cell. An observer reasoning with temporal awareness will identify that *I-2* is actually the optimal intervention since the subject would then turn away from the corridor at the latest after they reach cell *I-1*.

4 Active Goal Recognition Design

As our second contribution, we propose a new problem setting that combines elements of prior models and applies them

to the online setting to explicitly model the passage of time and enable the consideration of sequences of interventions.

4.1 Preliminaries

We assume action outcomes are deterministic and that the subject and observer take turns executing actions. Both agents immediately observe the effects of the other’s action prior to selecting their next action. We assume the subject agent acts optimally with respect to the domain costs C^* (defined formally in Section 4.2) and that it is unaware of or agnostic to the observer. This means the subject is not antagonistic to the observer by attempting to disrupt or hinder the goal recognition process through purposeful choice of maximally non-distinct plan prefixes. It also means that the subject’s planning is not strategic, i.e. it is not affected by interventions the observer has not yet taken.

We assume that the observer is neither assisting nor impeding the subject, and thus it must not change the optimal cost to achieve any goal. The observer’s interventions can, however, remove optimal plans available to the subject as long as there exists at least one plan with the original optimal cost available for each goal that the subject may be pursuing. Accordingly, we assume that if the plan the subject was executing is invalidated by an intervention, it is able to select a new optimal plan to its goal.

4.2 Problem Formulation

Definition 1. An AGRD domain is a tuple $D = \langle S, A_{subj}, A_{obs} \rangle$ where S is a set of system states representing the subject, observer, and their environment and A_{subj} and A_{obs} are functions that, given state s , return the actions available in s to the subject and observer, respectively. An action $a \in A(s)$ is a function that maps a state to its successor and the cost $c(a(s))$ of that transition.

An action that maps a state to itself is called an identity action and can be used by the observer when it does not have a more useful action to execute. We refer to actions returned by A_{obs} , i.e. observer actions, as interventions.

Definition 2. An AGRD Problem Instance is a tuple $I = \langle D, s_{root}, \mathcal{G}, P_{\mathcal{G}}, \rho \rangle$ where D is an AGRD domain, s_{root} is the start state, $\mathcal{G} = \{g_1, g_2, \dots, g_n\}$ is the set of possible goals the subject could be pursuing, where a goal $g \in \mathcal{G}$ is a set of states, $P_{\mathcal{G}} = \{p_1, p_2, \dots, p_n\}$ where p_i is the prior probability for $g_i \in \mathcal{G}$, and ρ is a value function mapping a state to a real number.

Definition 3. An action sequence $\alpha_{s_0} = \langle a_{obs}^1, a_{subj}^1, \dots, a_{obs}^n, a_{subj}^n \rangle$ is a sequence of alternating

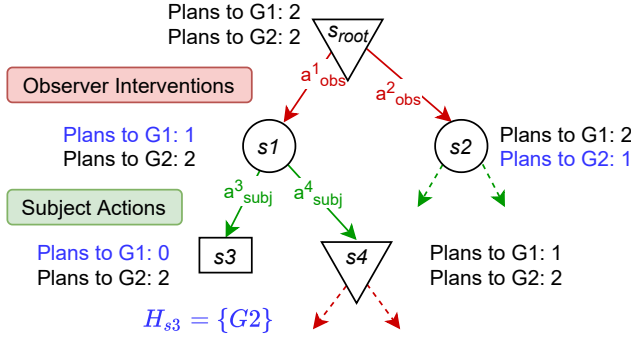


Figure 2: Example state space tree induced by $e_\rho(s_{root})$.

interventions and actions starting from state s_0 such that $s'_i = a^i_{obs}(s_{i-1})$, $s_i = a^i_{subj}(s'_i)$, $a^i_{obs} \in A_{obs}(s_{i-1})$, and $a^i_{subj} \in A_{subj}(s'_i)$.

α describes the interleaved action execution of both the observer and subject as the system evolves. In a slight abuse of notation, we will refer to the accumulated cost incurred by subject actions along an action sequence from s_{root} to s as $c_{subj}(s)$ and the number of subject actions in the sequence as $length_{subj}(s)$. Action sequences are different from the plans entertained by the subject, which do not include interventions.

Definition 4. A subject plan $\phi_{s_0,g} = \langle a^1, a^2, \dots, a^n \rangle$ is a sequence of actions starting from state s_0 such that $s_i = a^i(s_{i-1})$, $a^i \in A_{subj}(s_{i-1})$, and $s_n \in g$.

We will use this concept to define the behavior of an optimal subject which, according to our assumptions, is ignorant of the observer.

Definition 5. For a problem instance I with start state s_{root} , $C^*(g)$ is the minimum cost among all subject plans $\phi_{s_{root},g}$ and we notate a plan of this cost as $\phi_{s_{root},g}^*$.

Because we assume the observer cannot change the optimal cost to any goal, $C^*(g)$ always refers to the original optimal cost of g . The optimal cost to g from an intermediate state s along an optimal path to g is $C^*(g) - c_{subj}(s)$.

Definition 6. H_s is the set of goals g such that there exists at least one subject plan $\phi_{s,g}$ of cost $C^*(g) - c_{subj}(s)$.

We refer to sets H as goal hypotheses. They represent the remaining goals that an optimal subject could be pursuing.

Definition 7. A solution to an AGRD instance I is a policy $\pi : S \rightarrow A_{obs}$ that maps a state s to an intervention a_{obs} .

For any intervention $a_{obs}(s) = s'$, any subject plans from s may be invalidated as long as $H_s = H_{s'}$. In other words, the observer cannot remove goals from the hypothesis, only available plans to those goals.

4.3 Defining Optimal Solutions

We define optimal solutions in terms of a generic objective function $\rho : S \rightarrow \mathbb{R}$, a scoring function that maps a state to a real number. We minimize the expected value of ρ over a state space tree formed by simulating interventions and subject actions; see Figure 2 for an example. At any state, there

exists some number of optimal subject plans to each possible goal. Each intervention may change the available optimal plans by invalidating or enabling subject plans. However, an intervention may not change the optimal cost required for the subject to reach any goal it might be pursuing. Each subject action may reduce the number of optimal plans if the subject transitions beyond non-distinct plan prefixes. In the figure, the number of subject plans available to goal $G1$ in s_3 is 0, meaning $G1$ can be pruned from the goal hypothesis for s_3 . Since $|H_{s_3}| = 1$, s_3 is a leaf node of the tree whose value will be defined as $\rho(s_3)$. The value of non-leaf nodes is computed by backing up the value of child nodes to their parents. We take the minimum over possible interventions since we are looking for interventions that minimize our objective, but we must take the expectation over possible subject actions (weighted, as we explain below, by conditional goal probability) because we assume a non-strategic subject. This is an expectimin tree (expectation nodes depicted as circles, min as triangles) rather than a two-player game minimax tree.

Definition 8. Let P_s be a probability distribution over all actions the subject could take in state s , representing the probability the subject will select the corresponding action.

We will formally derive P_s and our method for pruning goals from H in Section 4.4.

Definition 9. $e_\rho : S \rightarrow \mathbb{R}$ is a function that computes the expectation of the best achievable ρ of a given state for a given scoring function ρ :

$$e_\rho(s) = \begin{cases} \rho(s) & \text{if } |H_s| \leq 1 \\ \min_{a_{obs} \in A_{obs}(s)} \text{score}(a_{obs}(s)) & \text{otherwise} \end{cases} \quad (1)$$

$$\text{score} = \mathbb{E}_{a_{subj} \sim P_s} [e_\rho(a_{subj}(s))] \quad (2)$$

The mutual recursion among equations 1 and 2 yields the alternating layers of the state space tree.

Given a problem instance I , an optimal policy π^* returns the intervention a_{obs}^* that will lead to the lowest score:

$$a_{obs}^* = \underset{a \in A_{obs}(s)}{\text{argmin}} \text{score}(a(s)) \quad (3)$$

4.4 Goal Probabilities

The probability distribution P_s over subject actions in a state s depends on the probability distribution over the goals it might be pursuing from that state. At s_{root} , these are P_G . As the subject acts to achieve its goal, some goals may become inconsistent with the subject's executed plan, requiring us to compute a goal posterior describing the updated probabilities over the remaining goals given our observations. We denote these posteriors as P_{H_s} , referring to the probability of each goal in the goal hypothesis H_s for the state s . We compute P_{H_s} via a sequence of Bayesian updates where the goal posterior of each update is used as the prior for the next.

Let $P_{H_s}(g_i)$ be the prior probability of g_i before subject action a and let $Plans_i(s)$ return all optimal subject plans ϕ_{s,g_i}^* . We assume that all optimal plans to a given goal are equally likely, so the likelihood $P_s(a|g_i)$, the probability of

action a given g_i , is the fraction of optimal plans to g_i for which a is the next action:

$$P_s(a|g_i) = \frac{|Plans_i(a(s))|}{|Plans_i(s)|}. \quad (4)$$

Then the marginal likelihood $P_s(a)$, the probability of a , is

$$P_s(a) = \sum_{g_i \in H_s} P_{H_s}(g_i) P_s(a|g_i) \quad (5)$$

and Bayes' rule tell us the posterior $P_{H_{s'}}(g_i)$, the probability of the goal g_i given an action a such that $a(s) = s'$, is

$$P_{H_{s'}}(g_i) = P_{H_s}(g_i|a) = \frac{P_{H_s}(g_i)P_s(a|g_i)}{P_s(a)}. \quad (6)$$

In other words, $P_{H_{s'}}(g_i) \propto P_{H_s}(g_i)P_s(a|g_i)$. The normalization factor is the sum of the action likelihood times goal prior over all possible goals (Eq. 5).

Eq. 6 is computed over all goals to get the full posterior $P_{H_{s'}}$. When we receive our next observation, $P_{H_{s'}}$ will be used as the prior. If the subject transitions to a successor that is not on any optimal plan to some goal g_i , the posterior p'_i becomes 0 and g_i is pruned from the hypothesis $H_{s'}$.

We note that the marginal likelihood (5) can be used as transition probabilities to define our problem as an MDP. In this MDP, the actions represent observer interventions and the transitions represent subject actions. The algorithm presented in Section 5 can then be viewed as a bottom-up dynamic programming solution to this MDP.

4.5 Objectives ρ

The metrics we discussed earlier can be defined as alternative functions ρ .

Definition 10. $\rho^\psi : S \rightarrow [0, 1]$ returns the fraction of the subject's plan that has been executed:

$$\rho^\psi(s) = \begin{cases} \frac{\text{length}_{\text{subj}}(s)}{\text{length}_{\text{subj}}(s) + |\phi_{s,g}^*|} & \text{if } |H_s| = |\{g\}| = 1 \\ \text{undefined} & \text{otherwise} \end{cases}. \quad (7)$$

Minimizing this function prioritizes distinguishing goals that will be achieved more quickly, thus preventing goals from being achieved before they are distinguished.

To see how wcd and ecd relate to our approach, we can re-frame distinctiveness:

Definition 11. $\rho^d : S \rightarrow \mathbb{R}^{\leq 0}$ returns the length of the action sequence the subject has executed:

$$\rho^d(s) = \begin{cases} \text{length}_{\text{subj}}(s) - 1 & \text{if } |H_s| = 1 \\ \text{undefined} & \text{otherwise} \end{cases}. \quad (8)$$

Using e_{ρ^d} approximately minimizes ecd, not wcd, since we are using an expectation instead of a global value. e_{ρ^d} and ecd are not exactly equivalent because of differences in how the action probability is computed, namely that e_{ρ^d} considers the number of possible plans the subject could be pursuing whereas ecd considers only the relative size of goal hypotheses between actions. Nevertheless, using ρ^d , our formulation can optimize distinctiveness if an application requires it.

Algorithm 1: Optimal AGRD

```

1 Init  $H, P_H, s$  from  $\mathcal{G}, P_{\mathcal{G}}, s_{\text{root}}$ 
2 while  $|H| > 1$  do
3   COMPUTEOPTIMALPLANS( $s, H$ )
4    $a_{\text{obs}} \leftarrow \text{IDENTITY}$ 
5   foreach  $a \in A_{\text{obs}}(s)$  do
6     if DFS-TRIAL( $s, a$ ) is best trial then
7        $a_{\text{obs}} \leftarrow a$ 
8    $s \leftarrow a_{\text{obs}}(s)$ 
9    $s' \leftarrow \text{OBSERVESUBJECTTRANSITION}(s)$ 
10   $H, P_H \leftarrow \text{UPDATEHYPOTHESIS}(H, P_H, s, s')$ 
11   $s \leftarrow s'$ 
12 return  $\rho(s)$ 

```

5 Optimal AGRD

As our third contribution, we present a straightforward optimal algorithm for AGRD. Algorithm 1 computes and executes the optimal intervention, then observes the subject and repeats. When the subject transitions to a state s where $|H_s| \leq 1$, it outputs $\rho(s)$. Note that when $|H_s| = 1$, we have uniquely identified the subject's goal, and that under our assumption of an optimal subject, it will never transition to a state where $|H_s| = 0$.

The goal hypothesis initially includes all possible goals. A goal is pruned when the subject takes an action that leads to a state from which there are no optimal plans to that goal. To determine this, we begin each loop iteration by finding all optimal plans to every goal in the current goal hypothesis from the current state [Line 3]. COMPUTEOPTIMALPLANS stores with every successor of the current state the number of those plans to each goal the subject could follow from that state. Once the subject transitions to a state where the number of plans to some goal g is 0, g is pruned from the hypothesis. To find the optimal intervention, we explore the tree in Figure 2 depth-first. DFS-TRIAL [Line 6] searches the tree described by e_ρ by simulating observer interventions and subject actions in sequence.

Once the full tree has been explored, we take the intervention that led to the best expected outcome and apply it to the domain [Line 8]. If all interventions from the current state produce the same score, the IDENTITY intervention will be selected by default. After observing the subject's next action, we update the goal hypothesis and compute its posterior probabilities according to Eq. 6 [Line 10].

6 Evaluation

To characterize the behavior of this optimal AGRD algorithm, we implemented it in C++ and ran experiments on Intel E8500 3.16 GHz CPUs. Our primary concern is how our implementation scales, i.e., which domain characteristics affect runtime the most.

We use grid pathfinding with 4-way movement as a testbed for our experiments. We used two patterns of obstacles. In uniform grids, each cell had a 0.2 probability of being blocked. Room grids contain walls with randomly placed

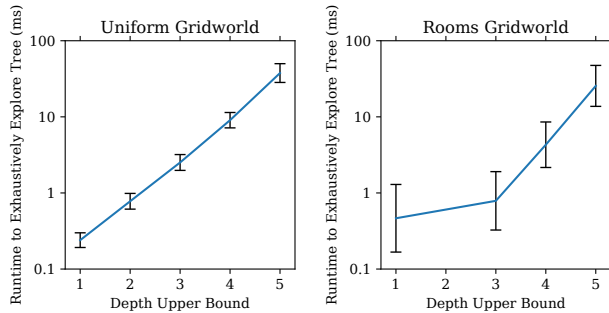


Figure 3: Avg runtime as a function of search tree depth.

openings, scaled down from those in the Moving AI repository [Sturtevant, 2012]. From among unblocked cells, the subject and observer’s start locations, possible goals, and cells the observer could block were chosen randomly. The observer follows the same movement rules as the subject and both can occupy the same cell. To block a cell, the observer must be adjacent to it. This is nearly identical to the terrorist domain used in previous work [Pozanco *et al.*, 2018; Shvo and McIlraith, 2020]. Instances were generated with 2, 3, or 4 goals, and we only report on instances where observer interventions can affect $e_p(s_{root})$, i.e. there exist meaningful interventions, which filtered out many instances.

Some factors clearly will not affect the running time of the algorithm: 1) The length of the plan to the subject’s actual goal, as we will have achieved goal recognition once the subject diverges from all plans to other goals, and 2) the prior distribution over goals, as the algorithm must examine all possibilities to ensure optimality. The size of the state space is a less precise parameter than the lengths of optimal plans to goals since the explored states are limited to those required to find plans to goals, therefore we examine the latter instead of the former. We did examine runtime as a function of the number of goals, which affects branching factor because there will be more subject actions. This did appear to play a minor role. We did not examine runtime as a function of the number of interventions, as this should have a similar effect as the number of goals, namely an increased branching factor.

The dominating factor in runtime was the length of the optimal plan to the second-furthest goal. Note that the search tree depth can be upper bounded by this length, as the observer is guaranteed to have achieved recognition at that depth. Figure 3 plots the geometric mean of the runtime with 95% confidence intervals as a function of this upper bound. (No room maps with an upper bound of 2 survived filtering.) The plot shows that, as one might expect, the runtime of optimal AGRD scales exponentially in the depth of the tree. The grids used in our experiments had varying numbers of cells, but the overwhelming factor was the depth of the search tree.

7 Discussion

Active Goal Recognition Design is just one way to formalize an observer with agency to aid goal recognition online. For example, Active Goal Recognition (AGR) [Shvo and McIlraith, 2020] focuses on partially observable domains. The

observer’s plans are limited to facilitating observations that do not alter any of the subject’s possible plans. They choose interventions to maximally reduce the size of their goal hypothesis prior to the next observation of the subject. However, this planning procedure does not consider how the system may evolve in two key ways. Firstly, if there is a choice between an intervention that distinguishes goal A and another that distinguishes goals B and C, AGR will always select the one that distinguishes 2 goals, even if the first intervention is the observer’s only opportunity to distinguish goal A before it is achieved. Secondly, as we discussed in Section 3.2, the assumption that the observer will have time to execute their plan without modeling the evolving system is problematic in domains with time pressure.

Goal Elicitation Planning [Amos-Binks and Cardona-Rivera, 2020] defines a setting where the observer’s interventions are interleaved with the subject’s actions as the observer seeks to minimize wcd online. From the start configuration, they generate a set of possible plans for the observer, then select from this set as the subject begins executing its own plan. However, once committed to a plan, their formulation does not permit an observer to react to actions the subject takes, even if the subject transitions to a part of the state space unaffected by the observer’s plan. Moreover, once an observer plan finishes executing, it has no means of generating a new plan to further reduce wcd .

In contrast to these approaches, we have presented a problem formulation closer to online planning. By continuously minimizing the expectation of the objective, we allow our observer to react to an evolving world. Our assumptions that the observer cannot change the cost to any goal and that the subject acts optimally place us closer to the original Goal Recognition Design setting, but relaxing these assumptions provides a promising direction for future work.

Our approach is not without drawbacks. As our evaluation demonstrates, our simple optimal algorithm is impractical for large domains since simulating every possible branch of the search tree is exponential in the depth of the tree. Adapting approximate MDP methods, such as MCTS, or tree pruning algorithms, similar to α - β pruning [Knuth and Moore, 1975], may be avenues for future work. For additional detail on the work presented in this paper, see Gall [2021].

8 Conclusions

As AI-driven systems become widespread, it is crucial to extend models of multiagent interaction. We formally defined AGRD, which extends GRD to handle problems that require reasoning online about other actors amid the passage of time. We showed that wcd and ecd can behave poorly in domains with goals whose plan lengths vary significantly. We also presented an algorithm that finds optimal interventions and analyzed its performance characteristics in benchmark domains. To the extent that a domain puts time pressure on the observer, the AGRD formulation will be a useful advance over previous work in Active Goal Recognition, and our observer’s ability to select interventions in reaction to subject actions is an advantage over the selection of precomputed plans in Goal Elicitation Planning.

References

- [Amos-Binks and Cardona-Rivera, 2020] Adam Amos-Binks and Rogerlio E. Cardona-Rivera. Goal elicitation planning: Acting to reveal the goals of others. *Proceedings of Advancements in Cognitive Systems*, 2020.
- [Ang *et al.*, 2017] Samuel Ang, Hau Chan, Albert Xin Jiang, and William Yeoh. Game-theoretic goal recognition models with applications to security domains. In *International Conference on Decision and Game Theory for Security*, pages 256–272. Springer, 2017.
- [Bisson *et al.*, 2011] Francis Bisson, Froduald Kabanza, Abder Rezak Benaskeur, and Hengameh Irandoust. Provoking opponents to facilitate the recognition of their intentions. In *Twenty-Fifth AAI Conference on Artificial Intelligence*, 2011.
- [Gall, 2021] Kevin C Gall. Active goal recognition design. Master’s thesis, University of New Hampshire, 2021.
- [Ha *et al.*, 2011] Eun Young Ha, Jonathan P Rowe, Bradford W Mott, and James C Lester. Goal recognition with markov logic networks for player-adaptive games. In *Seventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2011.
- [Kabanza *et al.*, 2010] Froduald Kabanza, Philippe Bellefeuille, Francis Bisson, Abder Rezak Benaskeur, and Hengameh Irandoust. Opponent behaviour recognition for real-time strategy games. In *Workshops at the Twenty-Fourth AAI Conference on Artificial Intelligence*, 2010.
- [Keren *et al.*, 2014] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design. In *Twenty-Fourth International Conference on Automated Planning and Scheduling*, 2014.
- [Keren *et al.*, 2015] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design for non-optimal agents. In *Twenty-Ninth AAI Conference on Artificial Intelligence*, 2015.
- [Keren *et al.*, 2016a] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design with non-observable actions. In *Thirtieth AAI Conference on Artificial Intelligence*, 2016.
- [Keren *et al.*, 2016b] Sarah Keren, Avigdor Gal, and Erez Karpas. Privacy preserving plans in partially observable environments. In *IJCAI*, pages 3170–3176, 2016.
- [Keren *et al.*, 2020] Sarah Keren, Haifeng Xu, Kofi Kwapong, David C Parkes, and Barbara Grosz. Information shaping for enhanced goal recognition of partially-informed agents. In *AAAI*, pages 9908–9915, 2020.
- [Knuth and Moore, 1975] Donald E Knuth and Ronald W Moore. An analysis of alpha-beta pruning. *Artificial Intelligence*, 6(4):293–326, 1975.
- [Masters and Sardina, 2019] Peta Masters and Sebastian Sardina. Cost-based goal recognition in navigational domains. *Journal of Artificial Intelligence Research*, 64:197–242, 2019.
- [Mirsky *et al.*, 2018] Reuth Mirsky, Roni Stern, Kobi Gal, and Meir Kalech. Sequential plan recognition: An iterative approach to disambiguating between hypotheses. *Artificial Intelligence*, 260:51–73, 2018.
- [Pozanco *et al.*, 2018] Alberto Pozanco, E Yolanda, Susana Fernández, and Daniel Borrajo. Counterplanning using goal recognition and landmarks. In *IJCAI*, pages 4808–4814, 2018.
- [Shvo and McIlraith, 2020] Maayan Shvo and Sheila A McIlraith. Active goal recognition. In *AAAI*, pages 9957–9966, 2020.
- [Sturtevant, 2012] Nathan Sturtevant. Benchmarks for grid-based pathfinding. *Transactions on Computational Intelligence and AI in Games*, 4(2):144–148, 2012.
- [Sukthankar *et al.*, 2014] Gita Sukthankar, Christopher Geib, Hung Hai Bui, David Pynadath, and Robert P Goldman. *Plan, Activity, and Intent Recognition: Theory and Practice*. Elsevier, 2014.
- [Wayllace *et al.*, 2017] Christabel Wayllace, Ping Hou, and William Yeoh. New metrics and algorithms for stochastic goal recognition design problems. In *IJCAI*, pages 4455–4462, 2017.