Information Shaping for Enhanced Goal Recognition

Sarah Keren
School of Engineering and Applied Sciences
Harvard University
skeren@seas.harvard.edu

Abstract

Understanding what agents, human or automated, know and how they choose to act given their knowledge is key to the ability to interpret their behavior and recognize what they are trying to achieve. Moreover, it may be possible to facilitate the recognition task by controlling the agent's access to information, a notion we refer to as *information shaping*. Relying on these observations, we consider here a two agent goal recognition setting; one agent, the *actor*, is partially informed and operates in an environment to achieve some goal, agnostic to the fact its[1] behavior may be monitored. The second agent, the *recognizer*, has perfect information, except for the actor's goal, which it tries to deduce by observing the actor's behavior. As a one time offline intervention, and with the objective of facilitating its ability to recognize the actor's goal as soon as possible, the recognizer can apply information shaping by revealing a bounded number of information items to the actor. Since the space of information items to reveal may be large, and since the goal of more informed agents is not necessarily easier to detect, information shaping needs to be done carefully and efficiently. After formally defining this new setting, we suggest several approaches for its solution and evaluate them on a set of standard benchmarks. Our results show the ability to facilitate recognition via information shaping, and the efficiency of our solution methods.

Introduction

We support a two agent setting, where a partially informed agent, the *actor*, acts in a deterministic environment to achieve some goal. The actor is not aware or indifferent to the fact its behavior may be monitored by the second agent, the *recognizer*, which has perfect knowledge, but does not know the actor's goal, and tries to deduce it as early as possible, by analyzing the actor's perceived behavior.

As a one time offline intervention, and with the objective of facilitating its ability to recognize the actor's goal as early as possible, the recognizer can apply *information shaping*, implemented as changes to the actor's sensor model. Such manipulations can potentially change the actor's behavior by making it easier to interpret. We restrict information shaping to be *truthful*, which means the information conveyed needs to hold in the true state and cannot mislead the actor.

[1] we hereon use 'it' to refer to agents, either automated or human

The ability to quickly understand what an agent is trying to achieve, without expecting it to explicitly communicate its objectives, is important in many current applications. These include assistive cognition (Kautz *et al.* 2003), where it may be critical to know when a visually impaired user is approaching a hot oven, security applications, where a system tries to detect users aiming at a specific destination (Boddy *et al.* 2005), and human-robot collaborative settings (Levine and Williams 2014), where a robot aims to recognize what component a human user is trying to assemble, so it can gather the tools needed for the task.

In the example applications above, a major task is to efficiently perform goal recognition, i.e., understand the goals of agents (human or virtual) by observing of their behavior (Ramirez and Geffner 2010; Cohen *et al.* 1981; Kautz and Allen 1986; Carberry 2001; Sukthankar *et al.* 2014). Common to all these settings, is that agents have incomplete information about their environment. This affects their behavior and is key to the ability to interpret it. In addition, all these settings can be controlled and modified in various ways. Specifically, since agents are partially informed, it may be possible to control their behavior by manipulating their knowledge and the way by which they acquire new information. Such manipulations may affect agent behavior and can be applied to induce a behavior that can be quickly associated to a specific goal, potentially facilitating the goal recognition task. To demonstrate, in an assisted cognition setting, an auditory signal can inform users about a hot oven. Early notification potentially causes agents aiming at a different goal (e.g., the cupboard) to move away from the oven, supporting early recognition of dangerous situations. To preserve usability, such sensors should be distributed economically, without misleading the human user.

This work offers an extension to the goal recognition design (GRD) framework, a recent task that deals with redesigning goal recognition settings in order to facilitate the early online goal detection (Keren *et al.* 2014; 2018a). GRD work so far assumes agents have perfect knowledge of their environment, thus failing to account for many realistic goal recognition settings, as the ones provided above. This work is the first step in extending the GRD framework to support agents with incomplete knowledge. Specifically, we focus on GRD in deterministic environments, and use a contingent planning (Bonet and Geffner 2011;

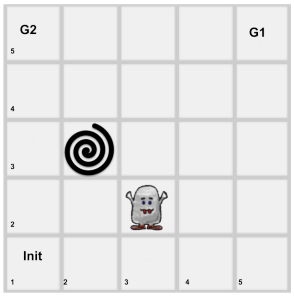

Figure 1: An example of a GRD-APK problem

Brafman and Shani 2012; Muise *et al.* 2014; Albore *et al.* 2009), to represent our actor. The design objective is to minimize the *worst case distinctiveness* (*wcd*)(Keren *et al.* 2014), denoting the maximal progress (measured as path cost) an agent can make, before its goal can be recognized. To minimize *wcd*, we suggest information shaping implemented as *sensor improvement* modifications, that ease the agent access to the true value of some environment variable.

The number of possible modifications may be extremely large. Also, as we demonstrate below, the goal of more knowledgeable agents is not necessarily easier to recognize, so such modifications need to be chosen carefully and efficiently to promote early goal recognition.

Example 1 *As a simple example, consider Figure 1, depicting a variation of the well known Wumpus domain(Russell and Norvig 2016), where a partially informed actor is trying to achieve one of two goals (indicated by G_1 and G_2 in the image), without falling into pits or encountering a deadly wumpus. The actor knows its current position, but initially does not know the locations of the pits and wumpuses. Nevertheless, when in a cell adjacent to a pit, it senses a 'breeze'. Equivalently, it can smell a Wumpus at an adjacent cell. The recognizer is assumed to have perfect information; it knows the locations of the actor, the pit (the spiral at cell $(2, 3)$) and the wumpus (at cell $(3, 2)$). It analyzes observations of the actor's behavior (its transitions between states) to recognize it's goal as early as possible.*

In our example, the actor start its progress at the state indicated by 'init'. Since it doesn't sense a breeze or stench, it deduces the adjacent cells are safe*, with no wumpus or breeze. Using a contingent solver known to the recognizer, an agent aiming at G_2 will start its progress by moving up. In contrast, an agent aiming at G_1 may go either up or right. This means that the choice to move up from the initial state, leaves the goal unrecognized.*

To promote early recognition, the recognizer can choose to share information with the actor. In particular, it can use its knowledge about the true positions of threats to reveal safe *cells. This choice is constraint by a design budget, limiting the number of facts to reveal, and by the obligation to convey truthful information (e.g., the recognizer cannot claim that cell $(3, 2)$ is safe).*

If the recognizer chooses to reveal cell $(5, 1)$ is safe, agents aiming at G_1 (originally indifferent to moving up or right) prefer moving right from the initial state, thanks to the added guarantees with regards to this option. In contrast, an actor aiming at G_2 may still prefer to move up due to the added cost of passing through $(5, 1)$ on the way to the goal. The goal of the actor becomes clear as soon as the first step is performed. Note that if, in addition, the recognizer reveals that cell $(1, 4)$ is also safe the initial situation is resumed, stressing the need to carefully select the information to reveal to facilitate the recognition task.

The contributions of this work are fourfold. First, we extend the deterministic GRD framework to support agents with partial information. We refer to our extended setting as GRD for Agents with Partial Knowledge (GRD-APK), and suggest information shaping modifications that can be applied to support early goal recognition. Second, after formally defining the GRD-APK setting, we show why previous pruning based approaches suggested in the GRD literature for optimal design (Keren *et al.* 2014; 2018a), cannot be used in this setting to guarantee an optimal sequence of modifications is found. Nevertheless, since our extended design frame induces a large space of possible modifications, in which the evaluation of each node (goal recognition setting) is costly, it is important to develop efficient methods for redesign, even without optimality guarantees. Accordingly, we describe several techniques developed for effective and efficient design in our setting. Third, by using PDDL (McDermott *et al.* 1998) to represent the design process, we offer a generic and adaptable framework for redesigning GRD-APK settings. Finally, we implement our suggested approaches for *wcd* calculation and redesign, and evaluate our techniques. Our preliminary results on a set of standard benchmarks demonstrate both the efficiency of our methods and the *wcd* reduction achievable by our information shaping modifications.

Background

Goal Recognition Design

Goal Recognition Design (GRD), (Keren *et al.* 2014; 2015; 2016a; 2018a; Wayllace *et al.* 2016; Son *et al.* 2016) is the task of analyzing and redesigning environments (either physical or virtual) to allow efficient online goal recognition. Following (Keren *et al.* 2018a), the definition of a GRD setting includes two main components; the analyzed goal recognition setting, and a design model.

A goal recognition setting can be defined in various ways (see (Sukthankar *et al.* 2014) for a recent survey), but typically involves two main components: a description of the agents that act in an environment to achieve one of a set of possible goals, and a goal recognition system (recognizer) that tries to deduce the agents' goals by observations collected on their behavior. The recognition settings can be classified as either 'keyhole', when actors are agnostic to the recognition process, 'adversarial' when they try to conceal their objectives, and 'intended', when the actor helps the recognizer detect its objective (Carberry 2001; Cohen *et al.* 1981). In this work, we assume the former, and support settings where agent behavior is not affected by the recognizer's presence.

The design model specifies the way by which a goal recognition setting can be modified. It includes the type of modifications that can be applied, the effect each modification may have an a goal recognition setting, and possible design constraints. Modifications suggested so far in the literature (Keren *et al.* 2018a) include action conditioning modifications, that limit the applicability of actions, and sensor refinements, that improve the recognizer's perception of the actor, itself assumed to have perfect knowledge. Design constraints included a design budget, limiting the number of allowed modifications, and a restriction that design could not increase the optimal cost to any of the goals. As we show later, we suggest new modifications relevant to our extended GRD setting, where agents may be partially informed.

Contingent Planning

To support agents with partial knowledge, we follow (Bonet and Geffner 2011), and consider partially observable (contingent) planning formulated as follows.

Definition 1 *A* **partially observable planning with deterministic actions model (POP-det)** *problem is a tuple* $P = \langle \mathcal{F}, \mathcal{A}, \mathcal{I}, \mathcal{G}, \mathcal{O} \rangle$ *where* $\mathcal{F}$ *is a set of fluent symbols,* $\mathcal{A}$ *is a set of actions,* $\mathcal{I}$ *is a set of* clauses *over* $\mathcal{F}$*-literals defining the initial situation,* $\mathcal{G}$ *is a set of* $fluents$*-literals defining the goal condition, and* $\mathcal{O}$ *represents the agent sensor model.*

An action $a \in \mathcal{A}$ is associated with a set of literals *prec*(a) representing the action's preconditions, and a set of conditional effects *eff*(a).The complement of a literal L is denoted by $\neg L$. The sensor model $\mathcal{O}$ is a set of observations $o \in \mathcal{O}$ represented as pairs (C, L) where C is a set of literals and L is a positive literal. The pair indicates that the true value of L is observable when C is true. Each observation $o = \langle C, L \rangle$ can be conceived as representing a sensor on the value of L that can be activated when C is true. We follow (Muise *et al.* 2014) in supporting a model for which observations correspond to sensing actions that can be applied by the agent when the conditions hold in its current state.

A state s is a truth valuation over the fluents F, for which the value may be 'true' or 'false'. For an agent, the value of a fluent may be known ('true' or 'false') or unknown. A *belief state* b is therefore a non-empty collection of states the agent deems possible at some point. A formula $\mathbb{F}$ *holds* in b if it holds for every state $s \in b$. The initial belief is the set of states that satisfy I, and the goal belief are those that satisfy G. A formula is *invariant* if it is true in each possible initial state, and remains true in any state that can be reached from the initial state using the available actions. A fluent is *hidden* if its true value is unknown. An action a is applicable in b if the preconditions of a hold in b, and the *successor* belief state $b^{'}$ is the set of states that results from applying the actions a to each state s in b. When an observation $o = (C, L)$ is activated, the successor belief is the *maximal* set of states in b that agree on L. An *action sequence* $\alpha = a_0, \ldots, a_m$ (possibly containing observations) is applicable in $b = b_0$ and results in the belief $b^{'} = b_{m+1}$ if the action a_i maps b_i into b_{i+1} for $i = 0, \ldots, m$. An *execution* is the sequence of belief states $e = b_0, b_1, \ldots, b_n$ and *history* is the sequence of actions and beliefs $h = b_0, a_0, b_1, a_1, \ldots, b_n, a_n, b_{n+1}$. A sequence is *complete* if the performing agent reaches a goal belief state.

A solution to a POP-det problem P is a *policy* Π which is a partial function from beliefs to action sequences. Following (Cimatti *et al.* 2003), a policy is *deterministic* if any belief b is mapped to at most one action sequence. Otherwise it is *non-deterministic*. There are 3 types of policies: *weak*, where at least one trajectory achieves the goal, and *strong* and *strong cyclic*, that require every possible trajectory achieves the goal. Strong policies impose the extra constraint that trajectories need to visit any state at most once.

A variety of solvers have been developed to solve a POP-det problem (e.g., (Bonet and Geffner 2011; Muise *et al.* 2014)). Specifically, to achieve efficient solutions, (Bonet and Geffner 2011) propose methods for *simple* POP-det models. A POP-det model is simple if the non-unary clauses in I are all invariant, and no hidden fluent appears in the body of a conditional effect. In simple problems there is no *information loss* and the model is considered *monotonic*. A POP-det problem is monotonic if for every literal L, if L is known in a reachable belief state b over a simple problem P, and $b^{'}$ is a belief reachable form b, then L is known in $b^{'}$. As a consequence, for every policy π and every induced trajectory $h = b_0, \ldots, b_n$ it follows that the number of states that make up beliefs b_i is a monotonically decreasing function over $[0, N]$, i.e. $|b_i| \geq |b_{i+1}|$ for every $0 \leq i < n$. By focusing on *simple problems*, Bonet and Geffner obtain an efficient translation that is linear in P which provides the basis for solving P using classical planners. We hereon assume all our problems are simple.

Goal Recognition Design Agents with Partial Knowledge (GRD-APK)

Our focus is on using information shaping to redesign goal recognition models with agents with partial knowledge. The *goal recognition design for agents with partial knowledge* (GRD-APK) is therefore comprised of an initial goal recognition setting, a measure by which a setting is evaluated and a design model, specifying the information shaping modifications that can be applied. We define each of these components separately, before combining them in Definition 7.

Goal Recognition

Our goal recognition model includes two agents. The *actor*, modeled as a partially informed contingent planning agent (Definition 1) with a goal, enters the system and executes history h until reaching a goal belief. The recognizer, with perfect knowledge except for the actor's goal, uses observations on the actor's behavior to recognize its goal.

We let $\mathbb{A}$, $\mathbb{F}$ and $\mathbb{B}$ represent the set of all actions, fluents, and belief states, respectively[2], and define goal recognition for agent with partial knowledge (GR-APK) as follows.

Definition 2 ***Goal recognition for agents with partial knowledge*** *(GR-APK) is a tuple* $R = \langle \epsilon, \mathcal{G}, \theta, \mathcal{O}^{ac} \rangle$ *where:*

[2]Our use of the universal sets of actions, fluents and belief states stems from our wish to support the process of modifying goal recognition settings, which will be described in following sections.

- $\epsilon = \langle \mathcal{F}, \mathcal{A}, \mathcal{I} \rangle$ *is the environment, which consists of the fluents* $\mathcal{F} \subseteq \mathbb{F}$*, actions* $\mathcal{A} \subseteq \mathbb{A}$ *(including sensing actions) and initial state* $\mathcal{I}$ *as defined in Definition 1. In addition, the definition may include a cost* $\mathcal{C}(a)$ *associated each action.*
- $\mathcal{G}$ *is a set of possible goals* G *s.t.* $|\mathcal{G}| \geq 2$ *and* $G \subseteq F$*.*
- $\theta : \mathbb{B} \times \mathcal{G} \rightarrow 2^{\mathcal{A}}$ *is the actor's decision making mechanism, specifying the set of actions an agent in belief state* b *and goal* G *may execute.*
- $\mathcal{O}^{ac}$ *is the actor sensor model.*

The cost of history h, denoted $\mathcal{C}_a(h) = \Sigma_i \mathcal{C}(a_i)$, is the accumulated cost of the performed actions (equal to path length when action cost is uniform). In executing h, the actor is following a policy from the set $\Pi(G)$ of possible policies to its goal $G \in \mathcal{G}$. This set is induced by the environment ϵ, describing the possible states and the transitions between them, the sensor models $\mathcal{O}^{ac}$, describing the way the actor collects observations from its surrounding and updates its belief state, and the actor decision making mechanism θ, specifying the actions an actor aiming at G may execute at each belief state (our framework supports non-deterministic policies, where there may be more than one possible action associated with each belief state).

Note that since we are analyzing the goal recognition setting, and need to account for all possible observations that may be collected when agents are active in the system, the definition of our goal recognition setting does not specify a particular history to be analyzed (a typical component in goal recognition models such as (Ramirez and Geffner 2010; Pereira *et al.* 2017)). Instead, it characterizes the different agent behaviors in the system, and the way they are perceived by the recognizer.

In our setting, the actor and recognizer both know ϵ and the set $\mathcal{G}$ of possible goals. While the actor has partial observability and collects information about the environment via its sensor model, the recognizer knows the true state of the world, but does not know beforehand the actor's goal. Also, it cannot see the actor's actions, but can observe its transitions between belief state. In addition, the recognizer knows $\mathcal{O}^{ac}$, and the actor's decision making mechanism θ. It observes the actor's behavior to recognize its goal as early as possible.

Evaluating a GR-APK model

Our objective is to use design to facilitate the online recognition task. Particularly, we want to minimize the *worst case distinctiveness* (*wcd*), which represents the maximal progress an actor can make before its goal is revealed. To define *wcd* in the context of GR-APK, we therefore first define the relationship between the observations collected by the recognizer and a goal.

Under the assumptions we make, when an actor follows a history h, the recognizer only observes the actor's transitions between belief states. We say that history h *satisfies* a policy π in an environment ϵ, if it can be generated by an agent following π in ϵ. It *satisfies* a goal G, if π is a policy to G.

Definition 3 *Given a GR-APK model* R*, history* h **satisfies** *a policy* π*, if* $\forall i\ 0 \leq i \leq n$*,* $a_i = \pi(b_i^{ac})$*.*
h **satisfies** *goal* $G \in \mathcal{G}$ *if exists* $\pi \in \Pi(G)$ *s.t.* h *satisfies* π*.*

In the following, we let $\mathcal{G}^{rec}(h)$ represent the set of goals that history h satisfies, i.e., the set of goals the recognizer deems as possible actor goals. We define an execution as *non-distinctive* if it satisfies more than one goal.

Definition 4 *Given a GR-APK model* R*, a history* h *is* **non distinctive***, if exists* G, G' *s.t.* $G \neq G'$*, and* h *satisfies* G *and* G'*.*

We mark the set of non-distinctive histories of a GRD-APK model T by $H^{nd}(R)$. The *worst case distinctiveness* (*wcd*) of a goal recognition model is defined as follows.

Definition 5 *The* **worst case distinctiveness** *of a model* R*, denoted by* $\text{wcd}(R)$ *is:*

$$\text{wcd}(R) = \begin{cases} \max\limits_{\alpha \in H^{nd}(R)} \mathcal{C}_a(\alpha) & H^{nd}(R) \neq \emptyset \\ 0 & otherwise \end{cases}$$

Recall that a policy $\pi \in \Pi(\mathcal{G})$ may be strong cyclic, potentially containing infinite loops. A policy with such a cycle, is considered as one that includes an infinite cost execution. In particular, this means *wcd* may have infinite cost.

Information Shaping for Minimizing *wcd*

To minimize *wcd* a system can be modified in various ways. As far as redesign options, our setup supports all modifications discussed in previous works, such as those that limit the applicability of certain actions and sensor refinement modifications, that improve the recognizer's sensor model.

Specific to our setting, where agents may have partial knowledge, is the ability to control agent behavior by manipulating its knowledge and the way by which it acquires new information from its surrounding. We restrict such manipulations to be *truthful*, i.e., they cannot be used to convey false information. In the context of contingent agents, this requirement is naturally implemented by improving the actor's sensor model, thus facilitating access to the true value of some environment feature. Specifically, we define *sensor extension* modifications, that add a single observation to a senor model, using $\boldsymbol{\mathcal{O}}$ to denote the set of all sensor models.

Definition 6 *A modification* $\delta : \boldsymbol{\mathcal{O}} \rightarrow \boldsymbol{\mathcal{O}}$ *is a* **sensor extension** *if exists* $o = (C, L)$ *s.t. for all* $\mathcal{O} \in \boldsymbol{\mathcal{O}}$*,* $\delta(\mathcal{O}) = \mathcal{O} \cup \{o\}$*.*

In practice, sensor extensions correspond to adding new sensors to the environment, or, in the extreme case, communicating to the actor the true value of a feature (setting $C = \emptyset$).

To demonstrate, in Example 1 the recognizer can apply sensor extensions by allowing actors to sense a stench in cell $(1, 2)$, two (rather then one) cells away from the wumpus in cell (3,2). This extension is implemented by adding the observation $o = (C, L)$ to the actor's sensor model, with L equal to the fluent $IsPitInCell(3, 2)$ and C corresponding to $AgentAtCell(1, 2)$. In practice, this corresponds to

a visual indication or sign, similar to the auditory signal indicating the oven is on in our assisted cognition example. The recognizer can also simply communicate with the actor and inform it of the true location of the wumpus ('there is a wumpus in cell 3,2'). Another, perhaps more subtle, option is the choice to reveal a location without a wumpus (e.g., 'no wumpus pit in cell 4,4' implemented by setting $C = True$ for $L = AgentAtCell(4, 4)$).

Finally, using the components above we define a GRD-APK problem as follows.

Definition 7 *A* **goal recognition design for agents with partial knowledge***(GRD-APK) problem is defined as a tuple $T = \langle R_0, \mathbf{\Delta}, \mathbf{b} \rangle$ where:*

- *R_0 is an initial goal recognition model, and*
- *$\mathbf{\Delta}$ are the possible sensor extensions*
- *b is a design budget, limiting the number of applied sensor extentions.*

Our objective is to find a set $\Delta \subseteq \mathbf{\Delta}$ of up to b sensor extensions to apply to R_0 to minimize the *wcd*. This objective is formally defined below, letting $wcd^{min}(T)$ represent the minimal *wcd* achievable in a GRD model T, and $R^{\mathbf{\Delta}}$ represent the goal recognition model that results from applying the set sensor extensions $\mathbf{\Delta}$ to R.

$$\begin{aligned} &wcd^{min}(T) = \underset{\Delta \subseteq \mathbf{\Delta}}{\text{minimize}} \quad wcd(R_0^{\Delta}) \\ &s.t. |\Delta| \leq b \end{aligned} \tag{1}$$

Any solution Δ^* to Equation 1 is *optimal*. It is *strongly optimal*, if it has minimal size among all optimal solutions.

Solution Methods

Equipped with the ability to change the actor's sensor model, we want to find the best way to apply information shaping to minimize the *wcd*. The challenge of this task lies in two key features. First, the number of possible information shaping options may be extremely large and evaluating the effect of each change may be costly, making it impractical to explore all design options exhaustively and important to develop efficient search techniques. Also, as we demonstrated in Example 1, the goal of more knowledgeable agents is not necessarily easier to recognize. This means that applying more information shaping modifications, and sensor extension modifications in particular, does not guarantee *wcd* reduction. This means that such modifications need to be chosen carefully to promote early goal recognition.

To confront these challenges, we follow (Keren *et al.* 2018a) and view the design process as a search in the space of modification sets $\Delta \subseteq \mathbf{\Delta}$. Each nodes is evaluated by its *wcd*, which is the measure we wish to minimize. The operators are the sensor extension modifications $\delta \in \mathbf{\Delta}$ that transition between goal recognition models. We start this section by describing the way by which each node is evaluated, i.e., our *wcd* calculation method, and then describe the overall design process and the methods we have devised for finding the best way to apply design.

Node Evaluation: Calculating *wcd*

According to Definition 5, the *wcd* of a model represents the maximal prefix of a policy that satisfies more than one goal. Recall that we assume the actor's decision making mechanism is known to the recognizer, who cannot observe which actions are performed by the actor, but is at least as knowledgeable and knows the actor's belief state.

At the initial state, all goals are possible. As the actor progresses in the system, the set of goals satisfied by its execution decreases. Given a history $h = b_0, a_0, b_1, a_1, \ldots, b_n, a_n, b_{n+1}$, we let h_i represent its prefix $h = b_0, a_0, b_1, a_1, \ldots, b_i$ up to belief state i.

Lemma 1 (GR Monotonicity) *Given a GRD-APK model R and a history h of n actions, for all $0 \leq i < j \leq n$, $\mathcal{G}^{rec}(h_j) \subseteq \mathcal{G}^{rec}(h_i)$.*

Proof Skecth: The definition of the sensor models guarantees the true state of the world always belongs to b^{ac}. Moreover, it guarantees the belief of the actor is *monotonic*, i.e. as the actor advances, the number of states in the belief state of cannot increase. Consequentially, as the actor advances, the set of plans, and therefore goals, satisfied by the history can only decrease. ∎

Lemma 1 guarantees a non-distinctive execution cannot have a distinctive prefix. We can therefore find the *wcd* of a GR-APK model by starting at the initial state and iteratively exploring the non-distinctive policy prefixes, until its most distant boundary is found.

Specifically, our *wcd* calculation method includes two main steps. First, we use the actor's decision making mechanism to find the policy (or policy set if the decision making mechanism is non-deterministic) for each goal. To reduce policy size, we can use the recognizer's knowledge to prune impossible outcomes, accounted for by the actor's policy, but guaranteed not to occur in practice. To demonstrate, in Example 1, the actor's policy accounts for the possibility of sensing a breeze in cell $(1, 2)$. This branch of the policy can be pruned, since the recognizer knows this cell is not adjacent to a pit. At the second stage, starting at the initial state, we iteratively explore the policy graphs of the different goals in parallel. Each node is represented by a belief state. For each node, we consider the action (or actions for non-deterministic policies) an agent may execute for each of the policy graphs. For each such action, we calculate the outcome state and the corresponding agent belief state. We group together transitions that lead to the same belief state, and prune transitions that lead to belief states that are not common to at least a pair of policies to two different goals. We stop our search when the most distant node, representing the maximal cost of a non-distinctive execution, is found. *wcd* is the cost of this execution.

Design: Algorithms for *wcd* Minimization

Given a way to evaluate a node, the challenge is to find an efficient way to search through the design options. The baseline approach for finding an optimal modification sequence in a GRD-APK model is a breadth first search (BFS) in the

modification space[3]. Since this approach iteratively explores modification sets of increasing size, it is guaranteed to find a strongly optimal solution. However, this approach is impractical as the problem size increases.

To promote efficiency, several approach have been suggested in the literature (Keren *et al.* 2018a). However, the conditions for which the suggested approach are guaranteed to yield optimal solutions do not hold in our setting. Specifically, Keren *et al.* 2018a provide conditions under which it is safe[4] to prune modifications that do not affect the pair of *wcd* plans of the currently explored node, i.e., the plans that share the maximal non-distinctive prefix. Specifically, one of the conditions requires that a GRD model is *monotonic-nd*, i.e., no new non-distinctive paths can be added to the model via design (and the *wcd* cannot increase). This pruning approach is not safe in our GRD-APK setting since our models are not monotonic-nd when information shaping modifications are used. As demonstrated in Example 1, the *wcd* can both increase and decrease as agents become more knowledgeable.

Even without the guarantees for strongly optimal solutions, efficiency is still a high priority. We therefore perform a heuristic best first search in the modification space. To guide the search, we map each modification to a superset of modifications to which it belongs, and use the value of applying the superset as a heuristic estimation of the node. We cache computed results, and reuse them for each modification that is mapped to the same superset.

To implement this approach, we exploit the fact that sensor extensions can be represented as parameterized modifications. We implement our approach by considering *padded* modification sequences, which include the set of modifications that share some value of a modification parameter with the one we want to evaluate. These values are stored and reused for all sequences that map to the same padded sequence. For example, to assess the value of a sensor extension that reveals the true value of an existence of a wumpus in cell $(1, 5)$ ($IsWumpusInCell(1, 5)$), we consider a modification that reveals this value for all cells in row 1. We reuse this value to estimate the value of modifications for cells $(1, 2)$, $(1, 3)$ etc.

The benefit of this approach, which we call *padding heuristic*, comes from the ability to store and reuse precomputed values, thus avoiding redundant computation of many nodes. This is an adaptation of the *relaxed modification heuristic*, suggested by Keren *et al.* 2018b, where padding is used to produce admissible estimations (and strongly optimal solutions) to an equi-reward utility maximizig design problem, where the objective is to maximize agent utility.

Since, as we demonstrated in Example 1, adding information shaping modifications can both increase and decrease the *wcd*, our approach cannot be shown to be admissible in our context, i.e., it cannot be shown to always underestimate the *wcd*. This means that our suggested heuristic approach does not necessarily produce optimal solutions when used to guide a best first search. We therefore suggest to enhance of our approach, by considering different patterns according to which the value of a modification is estimated. In the context of parameterized modifications, this means we consider different combinations of parameters according to which padding is performed. The heuristic value of a modification is then the minimal value among the different computations. For example, to assess the value of the sensor extension mentioned above, we use the value achieved when revealing the value of all cells in row 1 ($IsWumpusInCell(1, 1)$, $IsWumpusInCell(1, 2)$, ...). We also consider the modification set that reveals the value of all cells in column 5 ($IsWumpusInCell(1, 5)$, $IsWumpusInCell(2, 5)$, ...). The heuristic value is the minimal value among the two. As before, we cache and reuse the computed values. We call this approach *multi-pattern padding heuristic*. While computationally more demanding than the padding heuristic, by accounting for more aspects of the problem we can potentially increase the information provided by the heuristic.

Preliminary Empirical Evaluation

The objectives of our evaluation are twofold. First, we want to measure the effect sensor extensions have on *wcd*. Second, we want to measure the efficiency of our suggested design approach, comparing it to an exhaustive best first search exploration of the space of modifications.

Dataset We used 4 domains adapted from Bonet and Geffner 2011, using 20 instances of each.

- WUMPUS: corresponding to the setting in Example 1.
- C-BALLS (Colored-balls): the actor navigates a grid to pick up and deliver balls of different colors to destinations that depend on the color of the ball. The positions and colors of the balls are unknown to the actor, but when at a position, it observes if there are balls there, and if so, their colors.
- ROCK (Rock sample): a robot must navigate a grid in order to locate good rocks to sample. The robot has a sensor that detects good rocks at its vicinity, defined by the height of the antenna when the sensor is active.
- TRAIL: An agent must follow a trail of bits in a rectangular grid. The agent does not know the trail but it senses the bits surrounding it.

The adaptation from contingent planning to our GRD-APK setting involves specifying for each instance the set of possible goals and the set of possible sensor extension modifications. Table 1 specified how these were implemented for each domain.

	Possible Goals	Sensor Extensions
WUMPUS	gold locations	reveal safe cells
C-BALLS	ball distribution	reveal locations without a ball
ROCK	rocks to sample	reveal rock quality
TRAIL	the final stone	reveal stone locations

Table 1: Possible goals and design options for each domain

To support the design process, we introduced a PDDL file specifying the available modifications (and their precondi-

[3] in the case of a non-uniform cost for applying sensor extension, we can replace the BFS with a Dijkstra-based exploration.

[4] according to (Wehrle and Helmert 2014), pruning is safe as long as at least one optimal solution is not pruned.

tions). Sensor extension modifications were implemented as design actions, that add to the initial state fluents that represent the true value of some variable.

Setup We used the *k-replanner* (Bonet and Geffner 2011) to represent the actor's decision making mechanism and produce the plan an actor would follow w.r.t. to each goal. This means that actor follows a *planning under optimism* approach; it makes the most convenient assumptions about the values of the hidden variables, executes the plan that is obtained from the resulting classical planning problem, and revises the assumptions and re-plans, if during the execution, the observations refute the assumptions made. It only performs actions for which all preconditions are met, and fails when no such action exits. We note that in our computation of *wcd* we also considered failed executions, since they represent valid agent behavior.

Using Python, the design process was implemented as a best first search [5], using three heuristics for each instance.

- EX- zero heuristic which translated into an exhaustive exploration of the state space.
- PH- padding heuristic - a best first search with the padding heuristic over the first variable of a modification.
- MPPH multi-pattern padding heuristic- same as above, taking the minimal value over all parameters.

The design budget for all domains is 2. Each execution had a time limit of 1,800 seconds and 1000 design iterations.

To parse the design file, we adopted the parser of the pyperplan (Alkhazraji *et al.* 2016), to support the additional functionality of contingent planning. For each modification sequence, which represent a GRD-APK model and a node in our search, the parser provides the applicable modifications and the model that results from applying each of them.

Initial Results Table 2 summarizes the initial results. For each domain, the table shows 'sol', as the total number of solved instances (those completed within the allocated time and iteration limit). For instances completed by all approaches, the average *wcd* reduction is represented by 'Δ-*wcd*', the average calculation time by 'time', and the average number of nodes evaluated by 'nodes'.

First, we observe that design via information shaping reduces the *wcd* for all domains, with a reduction of 5.79 (about half) in the WUMPUS domain. For all domains, the heuristic approaches manage to solve the same number of instances and achieve the same *wcd* as the exhaustive approaches. As far as computation time, the heuristic approaches manage to outperform the exhaustive approach in all domains but TRAIL. This indicates that the overhead of computation time due to the heuristic calculations did translated into improved performance for most domains.

	Exhaustive				PH				MPPH			
	sol	Δwcd	time	nodes	sol	Δwcd	time	nodes	sol	Δwcd	time	nodes
WUMPUS	0.9	5.79	25.95	754.5	0.9	5.79	21.2	475.2	0.9	5.79	17.2	475.2
C-BALLS	0.6	4.1	55.41	402.5	0.6	4.1	39.82	395.2	0.6	4.1	35.6	358.2
ROCK	1.0	4.3	44.2	142.7	1.0	5.7	31.7	145.8	1.0	4.3	32.1	152.3
TRAIL	1	3.85	9.2	57.7	1	3.85	12.1	45.7	1	3.85	11.5	45.7

Table 2: Results per domain

Related Work

GRD, a special case of environment design (Zhang *et al.* 2009), was first introduced by Keren et al. (2014) and later extended (Keren *et al.* 2015; Son *et al.* 2016; Keren *et al.* 2016a; 2016b; Wayllace *et al.* 2016; Ang *et al.* 2017) by offering tools to analyze a variety of GRD settings. Common to all previous GRD work, is the assumption actors have perfect observability of their environment. This includes the work of (Keren *et al.* 2015), where agents may be sub-optimal, but their sensing and way to perceive their environment is not modeled. Similarly, the work by (Keren *et al.* 2016a; 2016b; 2018a) accounts for setting where the recognizer has partial observability and sensor refinement modifications are applied to enhance it's sensor model. Our work is the first to account for agents with partial information and suggests new information shaping modifications, implemented as sensor extensions, as a way to facilitate goal recognition by reducing the *wcd*.

Efficient communication protocols for information sharing is fundamental to various multi agent settings, e.g., (Xuan *et al.* 2001; Wu *et al.* 2011; Unhelkar and Shah 2016; Dughmi and Xu 2016). This work is the first to suggest using information sharing as a one time and offline intervention stage done to facilitate goal recognition.

Conclusion and Discussion

We support GRD for agents with partial knowledge, in which the recognizer cannot see the actions performed by the actor, but knows its belief state. We formulate the *wcd* measure by which we evaluate such goal recognition settings, and present new sensor extension modifications, used to enhance recognition by minimizing the *wcd*. Using a heuristic search, our preliminary results show how *wcd* can be efficiently reduced via redesign on a set of standard benchmarks adopted for our GRD setting.

There are many ways to extend this work. First, we use qualitative contingent planning models to represent the partially informed agents and their belief states. A natural next step is to extend our setting to quantitative models, and use partially observable markov descision processes(pomdps) (Kaelbling *et al.* 1998) to represent the actor, with belief states represented as probability distribution over the set of states. Another important extension involves extending our approach beyond the keyhole recognition settings we focus on. Transparent or explainable planning (MacNally *et al.* 2018) represent settings where and actors choose behaviors that facilitate recognition. These models completely rely on partially informed agents to be able to choose a behavior that maximizes the implicit communication of their intentions. In such settings, GRD can be viewed as a complementary approach, that can be applied to alleviate the need to completely rely on the actor, and reduce the non-distinctive behavior that are possible in the model.

[5] a link to code and benchmarks will be available in the final version, and omitted here to respect the blind review process.

Acknowledgements

The author would like to thank Miquel Ramirez and Nir Lipovetzky for the fruitful discussion that led to the formulation of the framework presented in this paper.

References

Alexandre Albore, Héctor Palacios, and Héctor Geffner. A translation-based approach to contingent planning. In *IJCAI*, pages 1623–1628, 2009.

Yusra Alkhazraji, Matthias Frorath, Markus Grutzner, Thomas Liebetraut, Manuela Ortlieb, Jendrik Seipp, Tobias Springenberg, Philip Stahl, Jan Wulfing, Malte Helmert, and Robert Mattmuller. Pyperplan: https://bitbucket.org/malte/pyperplan, 2016.

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.

M. S. Boddy, J. Gohde, T. Haigh, and S. A. Harp. Course of action generation for cyber security using classical planning. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS 2005)*, pages 12–21, 2005.

Blai Bonet and Hector Geffner. Planning under partial observability by classical replanning: Theory and experiments. In *IJCAI*, pages 1936–1941, 2011.

Ronen I Brafman and Guy Shani. A multi-path compilation approach to contingent planning. In *AAAI*, 2012.

S. Carberry. Technique for plan recognition. *User Modeling and User-Adapted Interaction*, 11(1-2):31–48, 2001.

Alessandro Cimatti, Marco Pistore, Marco Roveri, and Paolo Traverso. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 147(1-2):35–84, 2003.

P. R. Cohen, C. R. Perrault, and J. F. Allen. Beyond question-answering. Technical report, DTIC Document, 1981.

Shaddin Dughmi and Haifeng Xu. Algorithmic bayesian persuasion. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 412–425. ACM, 2016.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

H. Kautz and J. F. Allen. Generalized plan recognition. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI 1986)*, volume 86, pages 32–37, 1986.

H. Kautz, O. Etzioni, D. Fox, D. Weld, and L. Shastri. Foundations of assisted cognition systems. *University of Washington, Computer Science Department, Technical Report*, 2003.

Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS 2014)*, June 2014.

Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design for non optimal agents. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI 2015)*, January 2015.

Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design with non-observable actions. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI 2016)*, February 2016.

Sarah Keren, Avigdor Gal, and Erez Karpas. Privacy preserving plans in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2016)*, July 2016.

Sarah Keren, Avigdor Gal, and Erez Karpas. Strong stubborn sets for efficient goal recognition design. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS 2018)*, June 2018.

Sarah Keren, Luis Pineda, Avigdor Gal, Erez Karpas, and Shlomo Zilberstein. Relaxed modification heuristics for equi-reward utility maximizing design. In *In the ICAPS Workshop on Heuristic Search in Domain-independent Planning (HSDIP 2018)*, June 2018.

Steven James Levine and Brian Charles Williams. Concurrent plan recognition and execution for human-robot teams. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS 2014)*, June 2014.

Aleck MacNally, Nir Lipovetzky, Miquel Ramirez, and Adrian Pearce. Action selection for transparent planning. 2018.

Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. Pddl-the planning domain definition language. 1998.

Christian J Muise, Vaishak Belle, and Sheila A McIlraith. Computing contingent plans via fully observable non-deterministic planning. In *AAAI*, pages 2322–2329, 2014.

Ramon Fraga Pereira, Nir Oren, and Felipe Meneguzzi. Landmark-based heuristics for goal recognition. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI 2017)*, 2017.

M. Ramirez and H. Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI 2010)*, 2010.

Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

Tran Cao Son, Orkunt Sabuncu, Christian Schulz-Hanke, Torsten Schaub, and William Yeoh. Solving goal recognition design using asp. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI 2016)*, 2016.

Gita Sukthankar, Christopher Geib, Hung Hai Bui, David Pynadath, and Robert P Goldman. *Plan, activity, and Intent Recognition: Theory and practice*. Newnes, 2014.

Vaibhav V Unhelkar and Julie A Shah. Contact: Deciding to communicate during time-critical collaborative tasks in unknown, deterministic domains. In *AAAI*, pages 2544–2550, 2016.

Christabel Wayllace, Ping Hou, William Yeoh, and Tran Cao Son. Goal recognition design with stochastic agent action outcomes". In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 2016.

Martin Wehrle and Malte Helmert. Efficient stubborn sets: Generalized algorithms and selection strategies. In *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS 2014)*, 2014.

Feng Wu, Shlomo Zilberstein, and Xiaoping Chen. Online planning for multi-agent systems with bounded communication. *Artificial Intelligence*, 175(2):487–511, 2011.

Ping Xuan, Victor Lesser, and Shlomo Zilberstein. Communication decisions in multi-agent cooperation: Model and experiments. In *Proceedings of the fifth international conference on Autonomous agents*, pages 616–623. ACM, 2001.

Haoqi Zhang, Yiling Chen, and David C Parkes. A general approach to environment design with one agent. *Morgan Kaufmann Publishers Inc.*, 2009.