Privacy Preserving Plans in Partially Observable Environments

Sarah Keren and **Avigdor Gal** and **Erez Karpas**
{sarahn@tx,avigal@ie,karpase@}.technion.ac.il
Technion — Israel Institute of Technology

Abstract

Big brother is watching but his eyesight is not all that great, since he only has partial observability of the environment. In such a setting agents may be able to preserve their privacy by hiding their true goal, following paths that may lead to multiple goals. In this work we present a framework that supports the offline analysis of goal recognition settings with non-deterministic system sensor models, in which the observer has partial (and possibly noisy) observability of the agent's actions, while the agent is assumed to have full observability of his environment. In particular, we propose a new variation of worst case distinctiveness (*wcd*), a measure that assesses the ability to perform goal recognition within a model. We describe a new efficient way to compute this measure via a novel compilation to classical planning. In addition, we discuss the tools agents have to preserve privacy, by keeping their goal ambiguous as long as possible. Our empirical evaluation shows the feasibility of the proposed solution.

1 Introduction

Modern life takes place in environments that are constantly being monitored. Smart cities are equipped with surveillance and traffic cameras, Internet purchasing activities are monitored by online vendors, social media is monitored by law enforcement, *etc.* While monitoring activities may assist in preventing crimes, assisting consumers in getting good deals on products, and preventing online incitement, it also intrudes on the privacy of individuals, allowing means to track them and recognize their goals. In a perfect world it would be up to the individual to decide when and where to share information, what he perceives to be public space and how he chooses his practices of 'seeing and being seen' [Hatuka and Toch, 2016]. The world is, however, imperfect in more ways than one and it is this imperfection that may allow us to preserve privacy even in monitored environments.

Goal recognition systems may suffer from reduced and noisy observability due to lack of suitable sensors, insufficient sensor coverage, faulty sensors, inaccurate measurements, *etc.* The literature offers ways for goal recognition systems to handle partial observability [Ramirez and Geffner, 2011; Geib and Goldman, 2005; Avrahami-Zilberbrand *et al.*, 2005], and a recent work [Keren *et al.*, 2016] discussed the ability of such systems to recognize goals as early as possible in an extreme case of non-observability, in which an action can either be observed fully and accurately or not at all.

In this work we present a framework that supports the offline analysis of goal recognition under a full-fledged partial observability model that handles partial observability and the uncertainty that stems from it. In the proposed model, agents may either act optimally [Keren *et al.*, 2014], or be boundedly suboptimal [Keren *et al.*, 2015]. The agent follows his own strategy in a deterministic fashion and is fully aware of his environment and the impact of his actions. The goal recognition system has information about the agent's possible paths to his goal, but suffers from a limited ability to observe the agent's actions.

To model partial observability, we use non-deterministic system sensor models, in which the observer has partial (and possibly noisy) observability of the agent's actions, while the agent is assumed to have full observability of his environment. Similarly to Bonet and Geffner (2014), we model the system's partial observability using observation tokens that are emitted when an agent performs an action. We use a special null token to denote an action that cannot be observed by the system. The sensor model of the system is a function that maps each action into a set of possible observation tokens that may be emitted when an action is taken. Non-determinism and partial observability are modeled by mapping multiple actions into the same observation token, or the same action possibly emitting different tokens.

Given the proposed model, we offer a revised version of worst case distinctiveness (*wcd*) [Keren *et al.*, 2014], a measure that assesses the ability to perform goal recognition within a model. We define non-distinctive paths as paths that lead to different goals, yet can emit the same sequence of observation tokens, due to either shared actions between the paths, or the inability of the system to distinguish different actions that emit the same observation token. We describe a new efficient way to compute this measure via a novel compilation to classical planning. *wcd*, being a worst case measure, takes into account all possible observation tokens for each action, regardless of how unlikely they are to occur. Therefore, our compilation uses an *all token determinization* (similarly

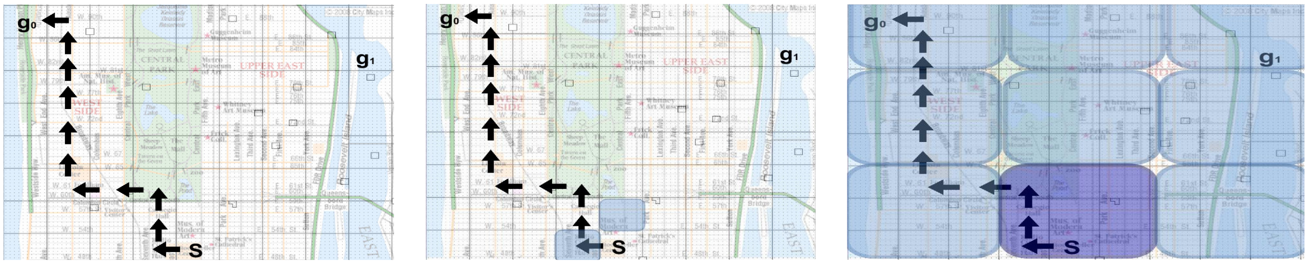

Figure 1: A *partially observable* goal recognition design problem

to the all outcome determinization in probabilistic planning), which ignores the probability of each observation token.

This measure serves as the basis for decision making for agents that wish to keep their goal ambiguous as long as possible to preserve privacy. To illustrate such decision making, we show how a user can choose, by using *wcd*, between two alternative monitoring environments that result in different levels of sensor refinement (*e.g.*, by turning the phone's GPS locator on or off). It is worth noting that our privacy preserving setting is different from other works that deal with privacy preservation in multi-agent planning, *e.g.*, [Brafman, 2015]. In our setting, there is a single agent who wants to keep its goal private from an observer, while in the multi-agent planning setting, we have multiple agents that must work together while attempting to keep some infromation private.

Example 1 *To illustrate the problem at hand, consider Figure 1, which illustrates a simplified grid setting. An agent walks from a starting point s to one of two goals, g_0 or g_1.*

Figure 1(left) illustrates the actual path an agent takes, aiming at g_0. The observation of this path differs according to the two possible sensor models depicted in Figure 1(middle) and Figure 1(right). Figure 1(middle) depicts a noisy sensor model. When the agent walks straight ahead there are two possible observations that may be produced, walking either straight or to the right. If the agent walks to the left the two possible observations are walking to the left or walking straight. In the worst case, walking to the left (towards g_0) does not emit an observation token that disambiguates between the paths to the two goals. On the right, the sensor is deterministic but its granularity forces it to increase its grid size, thus it cannot tell until late in the path which goal the agent targets (for example, because the GPS locator of the agent's phone is turned off and the use of location-by-cell decreases observability). The same observation token is shared by several of the original grid cells making it impossible to recognize the goal of the agent early on.

The rest of the paper is organized as follows. A model for partially observable environments is given in Section 2, followed by *wcd* calculation (Section 3) and a discussion on how *wcd* can assist in privacy preserving (Section 4). Empirical evaluation is presented in Section 5 followed by related work (Section 6) and concluding remarks (Section 7).

2 Model

A model for partially observable goal recognition design with action tokens (*grd-at*) is given by the tuple $D = \langle P_D, \mathcal{G}_D, \Pi_{leg}(\mathcal{G}_D), O_D, \mathcal{S}_D \rangle$, where:

- $P_D = \langle F_D, I_D, A_D \rangle$ is a planning domain [Ramirez and Geffner, 2009] using the STRIPS formalism [Fikes and Nilsson, 1972]. F_D is a set of fluents, $I_D \subseteq F_D$ is the initial state, and A_D is a set of actions, each represented by a triple $a = \langle pre(a), add(a), del(a) \rangle$, the precondition, add, and delete lists respectively, all subsets of F.
- $\mathcal{G}_D$ is a set of possible goals, where each possible goal $g \in \mathcal{G}_D$ is a subset of F_D.
- $\Pi_{leg}(\mathcal{G}_D) = \bigcup_{g \in \mathcal{G}_D} \Pi_{leg}(g)$ is a set of *legal* plans to each goal. A plan is a sequence of actions that take the agent from I_D to a goal in $\mathcal{G}_D$. Legal plans are those allowed under the assumptions on the agent's behavior.
- O_D is the observation token set, including the special token $o_\emptyset$, denoting that an action could be nonobservable.
- $\mathcal{S}_D : A_D \to 2^{O_D} \setminus \emptyset$ is a sensor model, mapping each action $a \in A_D$ into a set of observation tokens $\mathcal{S}_D(a) \subseteq O_D$ that may be emitted when a is executed.

An agent, aiming at one of the goals $g \in \mathcal{G}_D$, enters the system at the initial state I_D and executes one of the legal plans to g. The set of legal plans may include any plan an agent can take to achieve goal g, which can be described either explicitly or symbolically (e.g., the set of all optimal plans that do not make use of action a). Each action a performed by the agent emits one of the observation tokens $o \in \mathcal{S}_D(a)$, with the special token $o_\emptyset$ denoting that the action is not observed by the system. Note that the sensor model refers to how the goal recognition system observes the actions of the agent, while the agent himself is assumed to have full observability.

We now formally define the relationship between a path and the set of observation sequences it could emit.

Definition 1 *Given a path $\vec{\pi} = \langle a_1, ..., a_n \rangle$, the set of possible* observable projections *of $\vec{\pi}$ in D, denoted $op_D(\vec{\pi})$ ($op(\vec{\pi})$ when clear from the context), is defined as follows:*

$$op_D(\vec{\pi}) = \begin{cases} \langle\rangle & \vec{\pi} = \langle\rangle \\ \mathcal{S}(a_1) \times op(\langle a_2, ..., a_n \rangle) & \vec{\pi} = \langle a_1, \ldots, a_n \rangle \\ & \quad \wedge\, o_\emptyset \notin \mathcal{S}(a_1) \\ (\mathcal{S}(a_1) \setminus \{o_\emptyset\}) \times op(\langle a_2, ..., a_n \rangle) \bigcup & \vec{\pi} = \langle a_1, \ldots, a_n \rangle \\ \quad op(\langle a_2, .., a_n \rangle) & \quad \wedge\, o_\emptyset \in \mathcal{S}(a_1) \end{cases}$$

The empty token $o_\emptyset$ is excluded from the observable projection of a path. This allows the model to account for settings in which there is no way to know if and when some action has been performed. The *grd-at* setting is therefore a generalization of the *grd-po* setting [Keren *et al.*, 2016] in which each action is mapped to either the empty token (non-observable) or to the action name (observable).

Next, we define the relationship between a goal and the path executed by an agent as well as the relationship between a goal and the observation sequence emitted by the path.

Definition 2 *A path $\vec{\pi}$* satisfies *a goal g if it follows a valid plan to g, i.e., $\exists \pi \in \Pi_{leg}(g)$ s.t. $\vec{\pi}$ is a prefix of π. We denote the set of goals satisfied by path $\vec{\pi}$ in D by $\mathcal{G}_D^A(\vec{\pi})$.*
An observation sequence $\vec{o}$ satisfies *a goal g if $\exists \vec{\pi}$ that satisfies g and $\vec{o} \in op(\vec{\pi})$. We denote the set of goals satisfied by observation sequence $\vec{o}$ by $\mathcal{G}_D^O(\vec{o})$.*

Our analysis is based on the discovery of behaviors whose observable projection does not reveal the goal of the executing agent, *i.e.*, of paths whose observable projection satisfies more than one goal. We define non-distinctive observation sequences and paths as follows.

Definition 3 *$\vec{o}$ is a* non-distinctive observation sequence *if $|\mathcal{G}_D^O(\vec{o})| > 1$, that is, if it satisfies more than one goal. Otherwise, it is* distinctive.
$\vec{\pi}$ is a non-distinctive path *if $|\mathcal{G}_D^A(\vec{\pi})| \geq 1$ and $\max_{\vec{o} \in op(\vec{\pi})} |\mathcal{G}_D^O(\vec{o})| > 1$. Otherwise, it is* distinctive.

A non-distinctive path is therefore a path that might lead to some goal, and at least one of its observable projections $\vec{o} \in op(\vec{\pi})$ is non-distinctive. Note that even if a path can only lead to one goal, it is possible that its observable projection is non-distinctive, and therefore the path is non-distinctive as well. Finally, we can define worst case distinctiveness (*wcd*) in terms of non-distinctive paths. We denote the set of non-distinctive paths in D by $\vec{\Pi}_{nd}(D)$, and define the worst case distinctiveness of D as the maximum number of steps an agent can take in D without revealing his goal, that is $wcd(D) = \max_{\vec{\pi} \in \vec{\Pi}_{nd}(D)} |\vec{\pi}|$.

We denote the non-distinctive paths that are prefixes of legal plans to g_i by $\vec{\Pi}_{nd}(g_i)$, and the length of the longest path in $\vec{\Pi}_{nd}(g_i)$, by *wcd*-$g_i(D) = \max_{\vec{\pi} \in \vec{\Pi}_{nd}(g_i)} |\vec{\pi}|$. Following Keren *et al.* (2016), $wcd(D) = \max_i($*wcd*-$g_i(D))$, a property exploited by the compilation presented next.

3 Calculating *wcd*

The baseline method for *wcd* calculation is a breadth first search through the space of paths. A search node (path) can be pruned if it does not represent a prefix of a legal plan, or if it is distinctive. In order to determine if a path $\vec{\pi}$ is distinctive under the *grd-at* setting, we can solve a goal recognition problem for each possible observation sequence $\vec{o} \in op(\vec{\pi})$. If any of these observation sequences supports more than one possible goal, then $\vec{\pi}$ is non-distinctive.

For a non-deterministic sensor model, the number of possible observation sequences of a path may be exponential in path length. Thus, while the BFS method supports any possible set of legal plans, it is clearly inefficient. Next, we present a more efficient approach to calculate the *wcd* for the special cases of optimal or boundedly suboptimal (reaching a goal with a bounded cost beyond optimal) legal plans. Our approach uses a compilation to classical planning. For the sake of brevity, we only present the full details for optimal plans. The needed modification to fit the boundedly suboptimal case follows closely that of [Keren *et al.*, 2015].

For a pair of goals $\langle g_0, g_1 \rangle$, we formulate a planning problem that finds *wcd*-$g_{0,1}$, the maximal non-distinctive path leading to g_0, assuming g_1 is the only other possible goal. The *wcd* of the model is the maximum over all (ordered) goal pairs in the domain [Keren *et al.*, 2016].

The planning problem created for each goal pair (dubbed *common-declare*) is given in Figure 2. It involves two agents each aiming at one of the goals. Note that although the compilation includes two agents, the proposed model serves a single agent with the objective of finding a plan to follow that potentially maximizes privacy. The two-agent model is used to support two possible behaviors of agents aiming at different goals within a single search problem. This approach allows us to find the desired plan for an agent by solving a single planning problem using any off-the-shelf planner

Each agent ($agent_0$ and $agent_1$) has a copy f_i of each fact f. Both agents start at the same initial state, and aim at a different goal. Actions are divided into "real" actions, which change the state of the world, and "declare" actions, that correspond to the emission of the observation tokens and are used to account for what the goal recognition system sees. Whenever an agent performs a "real" action that emits an observation token k (indicated by the addition of $performed_{k,i}$ to the current state) it needs to declare the token before it can execute the next action (indicated by $declared_i$).

Both agents perform "real" actions separately, but can perform a "declare" either separately ($Declare_i^k$,$Declare_i^{o_\emptyset}$) or jointly ($Declare_{0,1}^k$), if both emit the same observation token k. We guarantee the discovery of *wcd*-$g_{0,1}$ by allowing joint declares only as long as $not\text{-}exposed$ (a boolean flag, initially set to true and allowing joint and unobserved declare statements) is true and assigning them a discount ϵ which is sufficiently small to prevent agents from deviating from an optimal plan. In addition, while $not\text{-}exposed$ is true, agents can declare an empty token ($Declare_i^{o_\emptyset}$) with no cost. Once either agent performs a separate $Declare_i^k$ action (corresponding to the execution stage in which the goal of the agents is revealed) $not\text{-}exposed$ is deleted once and for all, thus forcing the agents to report their action tokens separately and eliminating the discount $agent_0$ got for declare actions. Note that no cost is assigned to $agent_1$'s declare actions, guaranteeing an optimal solution maximizes the non-distinctive prefix of $agent_0$ rather then choosing a path that maximizes the number of non-observable actions performed by both agents.

We account for the non-deterministic system sensor model by considering all the possible observation sequences that may be produced by a path. This is achieved by creating for each action $a \in A$ a separate action copy a_i^k for each token $k \in \mathcal{S}(a)$, indicating that agent i executed action a and observation token k was emitted. This lets the planner choose the observation token each action occurrence emits, ensuring the compilation discovers the maximal non-distinctive path.

Using an optimal planner, we find an optimal plan for the compiled problem. The plan, which consists of actions for both agents, is divided into two parts, a joint declaration part, including all the "real" actions to which the joint "declare" actions refer, and a separate part. *wcd*-$g_{0,1}$ is the number of "real" actions $agent_0$ performs before the first action whose

For a *grd*-at problem $D = \langle P, \mathcal{G} = \{g_0, g_1\}, \Pi_{leg}(\mathcal{G}), O, \mathcal{S}\rangle$, where $P = \langle F, I, A\rangle$, we create a planning problem $P' = \langle F', I', A', G'\rangle$, with action costs C', where:

- $F' = \{f_0, f_1 \mid f \in F\} \cup \{\text{not-exposed}\} \cup \{declared_i \mid i \in \{0,1\}\} \cup \{performed_{k,i}\} \mid k \in O, i \in \{0,1\}\}$
- $I' = \{f_0, f_1 \mid f \in I\} \cup \{\text{not-exposed}, declared_0, declared_1\}$
- $A' = A_i \cup \{Declare^k_{0,1} \mid k \in O\} \cup \{Declare^k_i \mid i \in \{0,1\}, k \in O\} \cup \{Declare^{o_\emptyset}_i \mid i \in \{0,1\}\}$, where
  - A_i= $\{a^k_i \mid a \in A, k \in \mathcal{S}(a)\}$, where
    $a^k_i = \langle \{f_i \mid f \in pre(a)\} \cup \{declared_i\}, \{f_i \mid f \in add(a)\} \cup \{performed_{k,i}\}, \{f_i \mid f \in del(a)\} \cup \{declared_i\}\rangle$
  - $Declare^k_{0,1}$= $\langle \{performed_{k,0}, performed_{k,1}, \text{not-exposed}\}, \{declared_0, declared_1\}, \{performed_{k,0}, performed_{k,1}\}\rangle$
  - $Declare^{o_\emptyset}_i$= $\langle \{performed_{o_\emptyset,i}, \text{not-exposed}\}, \{declared_i\}, \{performed_{o_\emptyset,i}\}\rangle$
  - $Declare^k_i$= $\langle \{performed_{k,i}\}, \{declared_i\}, \{\text{not-exposed}, performed_{k,i}\}\rangle$
- $G' = \{f_0 | f \in g_0\} \cup \{f_1 | f \in g_1\} \cup \{declared_0, declared_1\}$
- $C'(a) = \begin{cases} 1 & \text{if } a \in A^i \\ \frac{\epsilon}{2} & \text{if } Declare^k_{0,1} \\ \epsilon & \text{if } Declare^k_0 \\ 0 & \text{if } Declare^k_1, Declare^{o_\emptyset}_i \end{cases}$

Figure 2: The *common-declare* compilation

token is declared separately.

Given a solution $\pi_{P'}$ to $P^{'}$, we mark the *projection* of $\pi_{P'}$ on each agent i as $\pi_{P'}(g_i)$, which includes all actions in A_i that appear in $\pi_{P'}$ (excluding the declare actions). Accordingly, the projection of the optimal solution $\pi^*_{P'}$ to $P^{'}$ on each agent is marked as $\pi^*_{P'}(g_i)$. We guarantee that $\pi^*_{P'}(g_i)$ yields a legal plan for both agents in D by bounding ϵ, the penalty for declare actions, such that the maximal accumulated penalty is lower than the smallest possible diversion from a legal path to any of the agents. The minimal cost diversion is 1. In addition, the model supports settings in which an agent may reach his goal without being observed. Accordingly, whenever $\epsilon < \frac{1}{max_i(C^*(g_i))}$, with C^* being the optimal cost, both agents act optimally in $P^{'}$ [Keren *et al.*, 2014].

Given a solution $\pi_{P'}$, $unexposed(\pi_{P'}(g_i))$ denotes the prefix of $\pi_{P'}(g_i)$ that includes all the actions excluding the last one that were performed by $agent_i$ before the first $Declare^k_i$ occurred.

Lemma 1 *$unexposed(\pi_{P'}(g_i))$ is non-distinctive.*

Proof: The compilation guarantees that any action in $unexposed(\pi_{P'}(g_i))$ is either non-observable, followed by a $Declare^{o_\emptyset}_i$ or observable, followed by a $Declare^k_{0,1}$ action in which both agents reported their pending token together. This means that actions that appear in $unexposed(\pi_{P'}(g_i))$ form a sequence that produces an observable projection shared by both goals, and is therefore non-distinctive. ■

Lemma 1 claims that $unexposed(\pi_{P'}(g_i))$ is non-distinctive. Theorem 1 shows that the optimal solution to P' yields *wcd*-$g_{0,1}$, thus concluding our proof of correctness.

Theorem 1 *Given a* grd-at *model D with two goals $\{g_0, g_1\}$ and a model P', created according to the* common-declare *compilation,* wcd-g$_{0,1}(D) = |unexposed(\pi^*_{P'}(g_0))|$.

Proof: The bound on ϵ described above guarantees that, apart from the no-cost operation *DoExpose*, the solution to P' consists solely of actions that form a pair of optimal paths to each of the goals (in the case of bounded non-optimal agents the cost of paths is optimal with regards to the bound). Lemma 1 shows that $unexposed(\pi^*_{P'}(g_0))$ represents a non-distinctive path. Since the only way to minimize the cost of P' is by maximizing the number of actions $agent_0$ performs before his first separate report guarantees that $\pi^*_{P'}$ is the solution to P' that maximizes $|unexposed(\pi_{P'}(g_0))|$. ■

As a final note, while the compilation described in Figure 2 could admit many permutations of the same plan, we can optimize the process by disallowing some permutations with additional constraints. First, before the agents' goals are exposed, we force them to act in a round-robin fashion, where each round consists of applying a sequence of non-observable actions, and ends with applying an action that emits a non-null observation token. Second, after exposing the goals, we force $agent_1$ to wait until $agent_0$ achieves his goal, similarly to [Keren *et al.*, 2014]. It is easy to see that from any pair of plans to g_0 and g_1, we can construct a solution for our compiled model that respects these constraints. We omit the full description of the compilation with these constraints for the sake of clarity.

4 Privacy Preservation with *wcd*

Having introduced a method for computing *wcd* of a model, we now discuss how it can be used to assist agents in their decision making. To start off, *wcd* can serve as a measure for privacy preservation, answering the question "what is the maximal duration an agent can keep his goal ambiguous?". The user can choose a path that potentially maximizes its privacy, which is the path found by solving a planning problem P' using the *common-declare* compilation, and which allows him to stay ambiguous for at most *wcd* steps.

Let g_0 be an agent's true goal. To maximize his privacy, the agent is interested in *wcd* value with respect to g_o only. Therefore, we seek *wcd*-$g_0(D)$ that can be computed by $max_{g_i \in G_D \setminus g_0} wcd\text{-}g_{0,i}(D)$.

While there is a single *wcd*-$g_o(D)$ value in a model, the compilation of Figure 2 returns only one among (possibly) several non-distinctive paths. An interesting question would be whether choosing one such path over another may bring an agent closer to his goal, in terms of number of steps? The following proposition provides an interesting observation to support the use of the chosen non-distinctive path.

Proposition 1 *Let $\vec{\pi}$ and $\vec{\pi}'$ be two non-distinctive paths that satisfy a goal g s.t. $|\vec{\pi}| = |\vec{\pi}'| = \text{wcd-}g$. Let $\Pi_{leg}(g)$ be the set of optimal plans to g and let $\vec{\pi}$ be a prefix of $\pi \in \Pi_{leg}(g)$ and $\vec{\pi}'$ be a prefix of $\pi' \in \Pi_{leg}(g)$. Then, $|\pi| - |\vec{\pi}| = |\pi'| - |\vec{\pi}'|$.*

Proof: $|\vec{\pi}| = |\vec{\pi}'| = \textit{wcd-}g$, which means that they are both maximal non-distinctive paths in the model, otherwise *wcd-g* would not be the longest non-distinctive path, contradicting its definition. Assume, by way of contradiction, and w.l.o.g. that $|\pi| - |\vec{\pi}| > |\pi'| - |\vec{\pi}'|$. Therefore, $|\pi| - \textit{wcd-}g > |\pi'| - \textit{wcd-}g$, yielding $|\pi| > |\pi'|$, which means that π is not an optimal plan, contradicting the fact that $\pi \in \Pi_{leg}(g)$ and that $\Pi_{leg}(g)$ is a set of optimal plans to g . ■

According to Proposition 1, any solution the planner provides may bring the agent as close as possible to his goal, while staying ambiguous about it, as long as the agent uses optimal plans. Whenever the set of legal plans is bounded optimal [Keren *et al.*, 2015] with a budget of b, it can be shown that the difference between the proposed path and the closest plan to his goal cannot be more than b.

An agent can affect the environment's ability to monitor his actions. We illustrate this ability via a modification example of *sensor cloaking*, a general approach for modifying partially observable goal recognition models by partitioning (or bundling) actions to emit one or more observation tokens. This is equivalent to adding (or removing) sensors. Next, we examine the effect of these modifications on the *grd-at* model.

We let $A_{\mathcal{S}}[a]$ represent the set of actions A that share a common observation token with action a and define the refinement relation between two sensor models as follows.

Definition 4 *Let $\mathcal{S} : A \to 2^O \setminus \emptyset$ and $\mathcal{S}' : A \to 2^{O'} \setminus \emptyset$ be two sensor models (defined over the same set of actions A but differ in their observation token set O and O'). $\mathcal{S}'$ is a refinement of $\mathcal{S}$ if for every action $a \in A$, (1) if $o_\emptyset \in \mathcal{S}'(a)$ then $o_\emptyset \in \mathcal{S}(a)$ and (2) $A_{\mathcal{S}'}[a] \subseteq A_{\mathcal{S}}[a]$.*

Note that the token set of a given action can include the empty token in the refined model, only if it was included in the original model. We show that cloaking by way of refinement cannot increase a model's *wcd*.

Theorem 2 *Given two* grd-at *models D and $D^{'}$ which differ only in their respective sensor model $\mathcal{S}$ and $\mathcal{S}'$ and induced observation token sets O and O' (and therefore $\vec{\Pi}_{leg}(\mathcal{G}_D) = \vec{\Pi}_{leg}(\mathcal{G}_{D'})$). If $\mathcal{S}'$ is a refinement of $\mathcal{S}$ then $\forall \vec{\pi} \in \vec{\Pi}_{leg}(\mathcal{G}_D)$, $\max_{\vec{o}' \in op_{D'}(\vec{\pi})} |\mathcal{G}^O_{D'}(\vec{o}')| \leq \max_{\vec{o} \in op_D(\vec{\pi})} |\mathcal{G}^O_D(\vec{o})|$.*

Proof: According to Definition 1, the set of possible observation sequences generated by the execution of a path $\vec{\pi}$ is $op_D(\vec{\pi})$ in D and $op_{D'}(\vec{\pi})$ in D'. Assume to the contrary that $\mathcal{S}'$ is a refinement of $\mathcal{S}$ but

$$\exists \vec{\pi} \in \vec{\Pi}_{leg}(\mathcal{G}_D) | \max_{\vec{o}' \in op_{D'}(\vec{\pi})} |\mathcal{G}^O_{D'}(\vec{o}')| > \max_{\vec{o} \in op_D(\vec{\pi})} |\mathcal{G}^O_D(\vec{o})| \qquad (1)$$

Since $\vec{\pi} \in \vec{\Pi}_{leg}(\mathcal{G}_D)$ there is at least one goal $g_{\vec{\pi}}$, which is the actual goal of the acting agent and which is satisfied by all the observable projections $\vec{o} \in op_D(\vec{\pi})$ (by Definition 2). Our assumption implies that there is at least one other goal g_o ($g_o \neq g_{\vec{\pi}}$) s.t. at least one observable projection of $\vec{\pi}$ satisfies both $g_{\vec{\pi}}$ and g_o in D', but no such observable projection in D, otherwise Eq. 1 fails to hold. We now look at the path $\vec{\pi}_{g_o} \in \vec{\Pi}_{leg}(g_o)$ that shares a common observable projection with $\vec{\pi}$ in D' (but not in D). $\vec{\pi}_{g_o}$ exists in both models since the legal paths are the same in both.

$\vec{\pi}$ and $\vec{\pi}_{g_o}$ share an observable projection in D'. Therefore, for all prefixes of $\vec{\pi}$ there is at least one prefix of $\vec{\pi}_{g_o}$ with which it shares an observable projection in D'. Let i represent the index of the first action a_i in $\vec{\pi}$ s.t. the prefix $\vec{\pi}_{1\cdots i} = \langle a_1, \cdots, a_i \rangle$ of π shares no common observable projection with a prefix of $\vec{\pi}_{g_o}$ in D (while there is always one for D', due to the way $\vec{\pi}$ is selected). According to Definition 1, this can happen in one of two cases. Either $o_\emptyset \in \mathcal{S}'(a_i)$ but $o_\emptyset \notin \mathcal{S}(a_i)$ and $op(\vec{\pi}_{1\cdots i}) \cap op(\vec{\pi}_{1\cdots i-1}) \neq \emptyset$. Since a_i is the first action for which no common observable projection exists in D, we know that $op(\vec{\pi}_{1\cdots i-1})$ shares its observable projection with some prefix of $\vec{\pi}_{g_o}$ both in D and D'. This, however, contradicts the assumption that if $\mathcal{S}'$ is a refinement of $\mathcal{S}$ then if $o_\emptyset \in \mathcal{S}'(a)$ then $o_\emptyset \in \mathcal{S}(a)$. Otherwise, there is at least one action in $\vec{\pi}_{g_o}$ that shares a common token with a_i in D' and not in D. This contradicts the assumption that for every action $a \in A$ $A_{\mathcal{S}'}[a] \subseteq A_{\mathcal{S}}[a]$. ■

	LOG					BLOCK					GRID			
	FULL	NO	POD-Obj	POD-Ac	POND	FULL	NO	POD-Obj	POD-Ac	POND	FULL	NO	POD	POND
wcd	1	1.2	1.2	13	13	5.3	6.1	6.1	8.5	8.5	2.8	3.02	3.09	3.18
time(LS)	2.85	—	—	—	—	4.9	—	—	—	—	0.3	—	—	—
time(LE)	35.1	83.75	—	—	—	72.4	74.1	—	—	—	0.3	0.24	—	—
time(CD)	263.8	107.1	94.7	117.3	397.3	82	103.3	96.1	113.2	373.5	0.63	0.64	0.48	1.33
% CD	0.8	0.9	0.9	0.85	0.7	1.0	1.0	1.0	1.0	0.75	1.0	1.0	1.0	1.0

Table 1: *wcd* Values, Running Time, and Coverage Ratio

Corollary 1 *Given two* grd-at *models D and $D^{'}$ that differ only in their sensor model s.t. $\mathcal{S}'$ is a refinement of $\mathcal{S}$ then* $\text{wcd}(D') \leq \text{wcd}(D)$.

Proof: Let $\vec{\pi}$ be a non-distinctive path in $\vec{\Pi}_{nd}(D')$. $\vec{\pi} \in \vec{\Pi}_{leg}(\mathcal{G}_D) \cap \vec{\Pi}_{leg}(\mathcal{G}'_D)$ since D and $D^{'}$ differ only in their sensor model. From Theorem 2, $\max_{\vec{o}' \in op_{D'}(\vec{\pi})} |\mathcal{G}^O_{D'}(\vec{o}')| \leq \max_{\vec{o} \in op_D(\vec{\pi})} |\mathcal{G}^O_D(\vec{o})|$. Therefore, $\vec{\pi} \in \vec{\Pi}_{nd}(D)$ (Definition 3). Since this holds for any non-distinctive path in D', we get that *wcd*$(D') \leq$ *wcd*(D). ■

To illustrate Corollary 1, consider Example 1 and Figure 1. The model on the left is more refined then the one on the right. In the left image, the marked path becomes distinctive after one step since moving left indicates the agent aims at g_0. The model on the right remains non-distinctive after the movement left and remains non-distinctive for the next two steps. Corollary 1 provides a useful tool for agents. By computing the *wcd* of two models, which differ in their sensor model, an agent can make an educated decision of whether the difference in the *wcd* value is worth the cost that comes with less refined sensors. For example, by turning off the phone's GPS, an agent may reach a higher value of *wcd* yet this comes at a cost of a less accurate positioning information that could confuse a navigation system. The agent can decide, based on his own utility function, which model is preferred.

5 Empirical Evaluation

We performed an evaluation of the *grd-at* framework to measure the effect non-deterministic partially observable sensor models have on the *wcd* value of a model and the efficiency of *wcd* calculation using the *common-declare* compilation.

Datasets As the number of possible sensor models is huge, we manually created a few such models for each domain, based on some simple rules. We used 20 problems from the LOGISTICS (LOG) domain and 12 from the BLOCKSWORLD domain (BLOCK), which were shown to be the most computationally challenging for goal recognition design in previous work [Keren *et al.*, 2016] and 34 GRID-NAVIGATION (GRID) problems of size 25×25.

Setup For each problem, we created 5 versions:

- Fully observable (FULL): all actions are observable.
- Non observable actions (NO): Non-observables in LOG are $Load$ and $Unload$, in BLOCK $PickUp$ and $PutDown$ and a random %10 of $Move$ actions in GRID.
- Partially observable deterministic (POD): For LOG and BLOCK we created two POD sensor models, in both of which we kept the same non-observable actions as NO. In POD-Ac, each observable action emits its type as an observation token (that is, in LOG all $Drive$ actions are indistinguishable, as are Fly, $LoadAirplane$, and $UnloadAirplane$, and $Stack$ and $Unstack$ in BLOCK). In POD-Obj, each observable action emits its first argument as an observation token (that is, we know which truck, airplane, package, or block were affected). For GRID we created a sensor model with reduced granularity, mapping each set of 4 target cells to the same token. Due to the domain simplicity, we refrained from presenting two model variations.
- Partially observable non-deterministic (POND): for BLOCK and LOG each of the observable actions can emit one of the two observation tokens described in POD. For GRID, we followed the example in Figure 1(right), where the movement to every cell can produce a token corresponding to a movement from any one of its adjacent cells. For all domains we keep the same non-observable actions as in NO and POD.

We compared three compilations: latestSplit (LS) [Keren *et al.*, 2014], latestExpose (LE) [Keren *et al.*, 2016], and *common-declare* (CD), using each compilation where it applies. We used the Fast Downward planning system [Helmert, 2006] running A^* with the LM-CUT heuristic [Helmert and Domshlak, 2009]. Experiments were run on Intel(R) Xeon(R) CPU X5690 machines, with a time limit of 30 minutes and memory limit of 2 GB.

Results Table 1 summarizes the results, and shows the average *wcd* and runtime for each compilation. The bottom row shows the ratio of problems solved by *common-declare*. The results show that *wcd* increases with the decrease of observability and increase of uncertainty. The GRID domain demonstrates the ability of an agent to improve his privacy by reducing sensor granularity. Comparing POD with FULL, we see an increase of *wcd*, on average, by about 10%.

As for running time, the more specialized compilations perform better when applicable, but the CD compilation solved most of the problems in all settings. The most time-demanding setting for *common-declare* is that of POND, due to the increased branching factor, with 50%-330% more time spent over the second time-demanding setting, on average.

6 Related Work

The first to establish the connection between the closely related fields of automated planning and goal recognition were Ramirez and Geffner (2009), presenting a compilation of plan recognition problems into classical planning problems. Several works on plan recognition followed this approach [Agotnes, 2010; Pattison and Long, 2011; Ramirez and Geffner, 2010; 2011] by using various automated planning techniques. We follow this approach as well and introduce a novel compilation into classical planning for finding *wcd* with non-deterministic sensor models.

Partial observability in goal recognition has been modeled in various ways [Ramirez and Geffner, 2011; Geib and Goldman, 2005; Avrahami-Zilberbrand *et al.*, 2005]. In particular, observability can be modeled using a sensor model that includes an observation token for each action [Geffner and Bonet, 2013]. The *grd-at* model covers all these aspects and more. In particular, we present a sensor model in which the set of observation tokens O includes an empty observation sequence $o_\emptyset$ and A includes a no-cost action a_{idle} by which an agent remains at his current position.

Goal recognition design was first introduced by Keren et al. (2014). This work, followed by several extensions [Keren *et al.*, 2015; Son *et al.*, 2016] offered tools to analyze and solve a *grd* model in fully observable settings. Another work [Keren *et al.*, 2016] presented a model that accounts for non-observable actions. Our work extends the system model by accounting for non-deterministic sensor models that can reflect any arbitrary assignment of action tokens emitted by actions. In addition, this work takes a point of view of an agent, demonstrating ways to affect the ability of a system to detect an agent's goal early on.

Distributed privacy-preserving multi-agent planning is a recent topic of interest [Bonisoli *et al.*, 2014; Torreño *et al.*, 2014; Luis and Borrajo, 2014; Brafman, 2015]. Agents supply a public interface and through a distributed planning process find a plan, without sharing a complete model of their actions and local state with other agents. In our setting, an agent works alone within an environment and "against" an observer, and tries to keep only his goal hidden for as long as possible.

7 Conclusions

The paper proposes a method for evaluating partially observable models as means to understanding the system ability to identify an agent's goal. We extend the definition of the *wcd* measure and propose ways to calculate it. We also present methods an agent may take to preserve his privacy, by modeling the environment, computing its *wcd*, identifying a path to maximize *wcd*, and changing sensor granularity to improve privacy. Our empirical evaluation supports the feasibility of our approach to find the *wcd* of such rich sensor models. In future work, we will test our model on additional (real-world) domains and explore the rich set of possible sensor model modifications for changing *wcd*.

Acknowledgements

The work was carried out in and partially supported by the Technion-Microsoft Electronic Commerce research center. The authors would like to thank Guy Shani for useful comments.

References

[Agotnes, 2010] T. Agotnes. Domain independent goal recognition. In *Stairs 2010: Proceedings of the Fifth Starting AI Researchers Symposium*, volume 222, page 238. IOS Press, Incorporated, 2010.

[Avrahami-Zilberbrand *et al.*, 2005] Dorit Avrahami-Zilberbrand, G Kaminka, and Hila Zarosim. Fast and complete symbolic plan recognition: Allowing for duration, interleaved execution, and lossy observations. In *Proc. of the AAAI Workshop on Modeling Others from Observations, MOO*, 2005.

[Bonet and Geffner, 2014] Blai Bonet and Hector Geffner. Belief tracking for planning with sensing: Width, complexity and approximations. *Journal of Artificial Intelligence Research*, pages 923–970, 2014.

[Bonisoli *et al.*, 2014] Andrea Bonisoli, Alfonso Gerevini, Alessandro Saetti, and Ivan Serina. A privacy-preserving model for the multi-agent propositional planning problem. In *PICAPS14 Workshop on Distributed and Multi-Agent Planning*, 2014.

[Brafman, 2015] Ronen I. Brafman. A privacy preserving algorithm for multi-agent planning and search. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1530–1536, 2015.

[Fikes and Nilsson, 1972] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3):189–208, 1972.

[Geffner and Bonet, 2013] Hector Geffner and Blai Bonet. A concise introduction to models and methods for automated planning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 8(1):1–141, 2013.

[Geib and Goldman, 2005] Christopher W Geib and Robert P Goldman. Partial observability and probabilistic plan/goal recognition. In *Proceedings of the International workshop on modeling other agents from observations (MOO-05)*, 2005.

[Hatuka and Toch, 2016] Tali Hatuka and Eran Toch. Being visible in public space: The normalisation of asymmetrical visibility. *Urban Studies*, page 0042098015624384, 2016.

[Helmert and Domshlak, 2009] Malte Helmert and Carmel Domshlak. Landmarks, critical paths and abstractions: What's the difference anyway? In Alfonso Gerevini, Adele E. Howe, Amedeo Cesta, and Ioannis Refanidis, editors, *Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009, Thessaloniki, Greece, September 19-23, 2009*. AAAI, 2009.

[Helmert, 2006] Malte Helmert. The fast downward planning system. *J. Artif. Intell. Res. (JAIR)*, 26:191–246, 2006.

[Keren *et al.*, 2014] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design. In *ICAPS Conference Proceedings*, June 2014.

[Keren *et al.*, 2015] 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.

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

[Luis and Borrajo, 2014] Nerea Luis and Daniel Borrajo. Plan merging by reuse for multi-agent planning. In *PICAPS14 Workshop on Distributed and Multi-Agent Planning*, 2014.

[Pattison and Long, 2011] D. Pattison and D. Long. Accurately determining intermediate and terminal plan states using bayesian goal recognition. *Proceedings of the First Workshop on Goal, Activity and Plan Recognition(GAPRec 2011)*, page 32, 2011.

[Ramirez and Geffner, 2009] M. Ramirez and H. Geffner. Plan recognition as planning. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI 2009)*, 2009.

[Ramirez and Geffner, 2010] 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.

[Ramirez and Geffner, 2011] M. Ramirez and H. Geffner. Goal recognition over pomdps: Inferring the intention of a pomdp agent. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence- Volume Three (IJCAI 2011)*, pages 2009–2014. AAAI Press, 2011.

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

[Torreño *et al.*, 2014] Alejandro Torreño, Eva Onaindia, and Oscar Sapena. Fmap: Distributed cooperative multi-agent planning. *Applied Intelligence*, 41(2):606626, 2014.