Goal Recognition Design with Non-Observable Actions

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

Abstract

Goal recognition design involves the offline analysis of goal recognition models by formulating measures that assess the ability to perform goal recognition within a model and finding efficient ways to compute and optimize them. In this work we relax the full observability assumption of earlier work by offering a new generalized model for goal recognition design with non-observable actions. A model with partial observability is relevant to goal recognition applications such as assisted cognition and security, which suffer from reduced observability due to sensor malfunction or lack of sufficient budget. In particular we define a *worst case distinctiveness* (*wcd*) measure that represents the maximal number of steps an agent can take in a system before the observed portion of his trajectory reveals his objective. We present a method for calculating *wcd* based on a novel compilation to classical planning and propose a method to improve the design using sensor placement. Our empirical evaluation shows that the proposed solutions effectively compute and improve *wcd*.

Introduction

Goal recognition design (*grd*) (Keren, Gal, and Karpas 2014; 2015) provides an offline analysis of goal recognition models, which are also known as plan recognition (Kautz and Allen 1986; Lesh and Etzioni 1995; Pattison and Long 2011). Goal recognition design formulates measures to assess the ability to understand the goal of an agent by the online observation of his behavior and finds efficient ways to compute and optimize them.

Goal recognition design consists of two main stages. The calculation stage finds the worst case distinctiveness (*wcd*) of the model, representing the maximal number of steps an agent can take in a system before the observed portion of his trajectory reveals his objective. *wcd* serves as an upper bound on the number of actions an agent can perform before his goal can be recognized. The second stage involves modifying the system (hence, goal recognition design) to minimize *wcd*.

Goal recognition design is applicable to any domain for which quickly performing goal recognition is essential and in which the model design can be controlled. In particular, goal recognition design is relevant to goal and plan recognition applications such as assisted cognition (Kautz et al. 2003) and security (Jarvis, Lunt, and Myers 2004; Kaluza, Kaminka, and Tambe 2011; Boddy et al. 2005) that suffer from reduced observability due to sensor malfunction, deliberate sabotage, or lack of sufficient budget. In a safe home setting, for example, reduced coverage means less control over access to sensitive areas such as a hot oven.

Earlier work on goal recognition design (Keren, Gal, and Karpas 2014; 2015) assumed fully observable models. In this work we relax this assumption and offer innovative tools for a goal recognition design analysis that accounts for a model with non-observable actions and a design process that involves sensor placement. The *partially observable* setting partitions the set of actions into observable and non-observable actions, reflecting, for example, partial sensor coverage. The proposed analysis relies on a partial incoming stream of observations. A key feature of this setting is that it supports a scenario where the system has no information regarding the actions of agents beyond what is observed. Therefore, in the absence of an observation, the system cannot differentiate unobserved actions from idleness of an agent. An example of such a scenario can be found in Real Time Location Systems (RTLS) where the last known location of an agent is taken as its current position.

The *partially observable* setting provides three contributions to the goal recognition design state-of-the-art. The key novelty lies in the differentiation it creates between an execution sequence and the observation sequence it emits. The fact that an observation sequence includes only the observable actions performed by an agent means that the same observation sequence may be generated by more than one execution sequence. In previous settings an execution sequence was considered *non distinctive* if it represented a prefix of legal plans to more than one goal. Here this condition is generalized to include execution sequences for which the emitted observation sequence is shared by prefixes of paths to more than one goal. The *worst case distinctiveness* (*wcd*) is then the length of the maximal execution that produces a non-distinctive observation sequence.

Second, despite the partial observability and the asymmetry of the model, attributed to the difference between an observation sequence and an execution sequence, we propose a method for calculating *wcd* that is based on a compilation to a fully observable classical planning framework. This com-

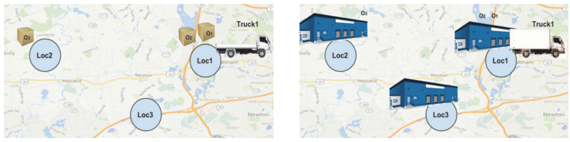

Figure 1: A *partially observable* goal recognition design problem

pilation allows us to exploit existing efficient tools for calculating *wcd*. Our empirical analysis shows that the compilation allows efficient computation of *wcd*.

The third contribution of this work involves finding an optimal set of modifications that can be introduced to the model in order to reduce *wcd*. We introduce a new design-time modification method that involves exposing non-observable actions, *e.g.*, by (re)placing sensors. This modification method is used, in addition to removing actions from the model (Keren, Gal, and Karpas 2014), to minimize *wcd* while respecting restrictions on the number of allowed modifications. The empirical analysis reveals that the combination of these two types of modifications leads to greater improvements than each of the measures separately.

Example 1 *To illustrate the objective of calculating and optimizing* wcd *of a goal recognition design model, consider Figure 1, which demonstrates a simplified setting from the logistics domain. There are* 3 *locations,* Loc_1, Loc_2 *and* Loc_3*, a single truck that is initially located at* Loc_1*, and* 3 *objects that are initially placed such that* O_1 *and* O_2 *are at* Loc_1*, and* O_3 *is at* Loc_2*. Objects can be moved by loading them (L) onto the truck and unloading them (*UL*) in their destination after the truck reaches it using a drive action (D). There are two possible goals,* g_0*:* O_1 *at* Loc_2 *and* O_2 *and* O_3 *at* Loc_3*, and* g_1*:* O_1 *at* Loc_3 *and* O_3 *at* Loc_1*. Optimal plans are the only valid plans in this example.*

In the fully observable setting (see Figure 1(left)) wcd $= 1$ *since* O_1 *needs to be loaded on the truck for both goals to be achieved. The goal is revealed by the next action, which can either be* $L(O_2)$ *for* g_0 *or* $D(Loc_1, Loc_2)$ *for* g_1*.*

In Figure 1(right)), the loading depot is covered. Therefore, all load and unload actions are non-observable and the only observable actions are those that relate to the movements of the truck. Since the truck needs to travel from Loc_1 *to* Loc_2 *and then to* Loc_3 *for achieving both goals, the goal is revealed only if* $D(Loc_3, Loc_1)$ *is performed. This means that* g_0 *can be achieved without the system being aware of it (*wcd $= 8$*). Exposing* $L(O_2)$*, by placing a sensor on the object, changes the situation dramatically by reducing* wcd *to its value in the fully observable setting.*

Background

The basic form of automated planning, referred to as *classical planning*, is a model in which the actions of agents are fully observable and deterministic. A common way to represent classical planning problems is the STRIPS formalism (Fikes and Nilsson 1972): $P = \langle F, I, A, G, C\rangle$ where F is a set of fluents, $I \subseteq F$ is the initial state, $G \subseteq F$ represents the set of goal states, and A is a set of actions. Each action is a triple $a = \langle pre(a), add(a), del(a)\rangle$, which represents the precondition, add, and delete lists respectively, all are subsets of F. An action a is applicable in state s if $pre(a) \subseteq s$. If action a is applied in state s, it results in a new state $s' = (s \setminus del(a)) \cup add(a)$. $C : A \to \mathbb{R}_0^+$ is a cost function that assigns each action a non-negative cost.

The objective of a planning problem is to find a plan $\pi = \langle a_1, \ldots, a_n\rangle$, a sequence of actions that brings an agent from I to a goal state. The cost $c(\pi)$ of a plan π is $\Sigma_{i=1}^n(C(a_i))$. Often, the objective is to find an optimal solution for P, an optimal plan, π^*, that minimizes the cost. We assume the input of the problem includes actions with a uniform cost of 1. Therefore, plan cost is equivalent to plan length, and the optimal plans are the shortest ones.

Model

A model for partially observable goal recognition design (*grd-po*) is given as $D = \langle P_D, \mathcal{G}_D, \Pi_{leg}(\mathcal{G}_D)\rangle$ where:

- $P_D = \langle F_D, I_D, A_D\rangle$ is a planning domain where $A_D = A_D^o \cup A_D^{no}$ is a partition of A_D into observable and non-observable actions, respectively.
- $\mathcal{G}_D$ 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 the set of *legal* plans to each of the goals. A plan is an execution of actions that take the agent from I to a goal in $\mathcal{G}_D$. A legal plan is one that is allowed under the assumptions made on the behavior of the agent.

The *grd-po* model divides the system description into three elements: *system dynamics*, defined by P_D and $\mathcal{G}_D$, *agent strategy* defined by $\Pi_{leg}(\mathcal{G}_D)$, and *observability* defined by the partition of A_D into A_D^o and A_D^{no}. Whenever D is clear from the context we shall refrain from adding the subscript.

Whereas a plan π is a full execution, a *path* is a prefix of a legal plan. We denote the set of paths in D as $\vec{\Pi}(\mathcal{G}_D)$ and the set of paths that are prefixes of plans achieving goal $g \in \mathcal{G}_D$ as $\vec{\Pi}(g)$. In the *partially observable* setting the observation sequence that is produced by a path includes only the observable actions that are performed. Accordingly, an *observation sequence* $\vec{o} = \langle a_1, \cdots, a_n\rangle$ is a sequence of actions $a_j \in A^o$. For any two action sequences $\langle a_1, \cdots, a_n\rangle$ and $\langle a_1', \cdots, a_m'\rangle$ the concatenation of the action sequences is denoted by $\langle a_1, \cdots, a_n\rangle \cdot \langle a_1', \cdots, a_m'\rangle$.

The relationship between a path and an observation sequence is formally defined next.

Definition 1 *Given a path* $\vec{\pi}$*, the* observable projection *of* $\vec{\pi}$ *in D, denoted* $op_D(\vec{\pi})$ *(*$op(\vec{\pi})$ *when clear from the context), is recursively defined as follows:*

$$op_D(\vec{\pi}) = \begin{cases} \langle\rangle & \text{if } \vec{\pi} = \langle\rangle \\ \langle a_1\rangle \cdot op_D(\langle a_2 \cdots a_n\rangle) & \text{if } \vec{\pi} = \langle a_1, \cdots, a_n\rangle \text{ and } a_1 \in A_D^o \\ op_D(\langle a_2, \cdots, a_n\rangle) & \text{if } \vec{\pi} = \langle a_1, \cdots, a_n\rangle \text{ and } a_1 \in A_D^{no} \end{cases}$$

It is worth noting that the fully observable setting (Keren, Gal, and Karpas 2014; 2015) is a special case in which the entire action set is observable. In this case, $A^{no} = \emptyset$, $A^o =$

A, and the observable projection of any action sequence is equivalent to the action sequence itself.

The relation between a path and a goal and an observation sequence and a goal are defined as follows.

Definition 2 *A path* $\vec{\pi}$ satisfies *a goal* g *if* $\vec{\pi} \in \vec{\Pi}(g)$.
An observation sequence $\vec{o}$ satisfies *a goal* g *if* $\exists \vec{\pi} \in \vec{\Pi}(g)$ *s.t.* $\vec{o} = op(\vec{\pi})$.

For example, $\langle L(O_1), L(O_2), D(Loc_1, Loc_2)\rangle$ in Example 1, hereon referred to as $\vec{\pi}_{ex1}$, satisfies only g_0. However, its observable projection $op(\vec{\pi}_{ex1}) = \langle D(Loc_1, Loc_2)\rangle$ satisfies both g_o and g_1.

We let $\mathcal{G}^A_D(\vec{\pi})$ and $\mathcal{G}^O_D(op(\vec{\pi}))$ represent the set of goals that are satisfied by the executed path $\vec{\pi}$ and its observable projection $op(\vec{\pi})$, respectively. The distinction the *grd-po* model creates between $\mathcal{G}^A_D(\vec{\pi})$ and $\mathcal{G}^O_D(op(\vec{\pi}))$ is a key element of the proposed framework. More generally, we examine the effect of *concealing* an action by making it non-observable (that is, moving it from A^o to A^{no}) on the number of goals the observable projection of a path satisfies. The fact that concealment may only increase the number of goals will be fundamental in our analysis of the *grd-po* model.

Theorem 1 *Let* D *and* D' *be two* grd-po *models that are identical except that* $A^{no}_D \subseteq A^{no}_{D'}$. *For any* $\vec{\pi} \in \vec{\Pi}(\mathcal{G}_D)$,

$$\mathcal{G}^O_D(op(\vec{\pi})) \subseteq \mathcal{G}^O_{D'}(op(\vec{\pi}))$$

Proof: According to Definition 1, the observation sequence generated by the execution of a path $\vec{\pi}$ is $op(\vec{\pi}) = \langle a_1, ..., a_n\rangle$ where $a_i \in A^o$. If $\forall a \in \vec{\pi}$, $a \notin A^{no}_{D'} \setminus A^{no}_D$ then $op_D(\vec{\pi}) = op_{D'}(\vec{\pi})$ since none of the actions in $\vec{\pi}$ changed their observability property. Therefore, $\mathcal{G}^O_D(op(\vec{\pi})) = \mathcal{G}^O_{D'}(op(\vec{\pi}))$. Otherwise, $\exists a_i \in \vec{\pi}$, $1 \le i \le n$ s.t. $a_i \in A^{no}_{D'} \setminus A^{no}_D$. According to Definition 1,
$op_D(\vec{\pi}) = op_D(\langle a_1, ..., a_{i-1}\rangle \cdot a_i \cdot op_D(\langle a_{i+1}, ..., a_n\rangle)$
whereas
$op_{D'}(\vec{\pi}) = op_{D'}(\langle a_1, ..., ai-1\rangle \cdot op_{D'}(\langle a_{i+1}, ..., a_n\rangle)$.
For any path $\vec{\pi}'$ that is identical to $\vec{\pi}$ except that a_i is replaced by a possibly empty sequence of non-observable actions, $op_D(\vec{\pi}) \neq op_D(\vec{\pi}')$ but $op_{D'}(\vec{\pi}) = op_{D'}(\vec{\pi}')$. Since $\vec{\pi}'$ may lead to a different goal than $\vec{\pi}$, $\mathcal{G}^O_D(op(\vec{\pi})) \subseteq \mathcal{G}^O_{D'}(op(\vec{\pi}))$. ■

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

Definition 3 $\vec{o}$ *is a* non-distinctive observation sequence *if it satisfies more than one goal. Otherwise, it is* distinctive.
$\vec{\pi}$ *is a* non-distinctive path *if its observable projection* $\vec{o} = op(\vec{\pi})$ *is non-distinctive. Otherwise, it is* distinctive.

The path $\vec{\pi}_{ex1}$ (Example 1) is non-distinctive since its observable projection satisfies both g_0 and g_1. Lemma 1, given next, sets the relationship between a path and its prefixes (The proof is omitted due to space restrictions).

Lemma 1 *Any prefix of a non-distinctive path is non-distinctive.*

Next, we define the measure by which we evaluate a model. The worst case distinctiveness (*wcd*) of a *grd-po* model represents the maximal number of steps an agent can advance in a system without revealing his goal. We mark the set of non-distinctive paths in D as $\vec{\Pi}_{nd}(D)$ and define *wcd* as maximal length of paths in $\vec{\Pi}_{nd}(D)$.

Definition 4 *The* worst case distinctiveness *of a model* D, *denoted by* $\text{wcd}(D)$ *is:*

$$wcd(D) = \max_{\vec{\pi} \in \vec{\Pi}_{nd}(D)} |\vec{\pi}|$$

The distinction the *grd-po* model creates between the set of goals a path satisfies and the set of goals satisfied by its observable projection affects the way *wcd* can be calculated. To find the *wcd* of a model D one needs to account for all paths $\vec{\pi} \in \vec{\Pi}_D(\mathcal{G})$ that satisfy at least one goal $(1 \le |\mathcal{G}^A_D(\vec{\pi})|)$ and whose observable projection satisfies at least two goals $(2 \le |\mathcal{G}^O_D(op(\vec{\pi}))|)$. This requirement promotes an analysis that partitions the set of valid paths according to the goals they satisfy, and examines each group separately before combining the results. We let $\vec{\Pi}_{nd}(g_i)$ represent the non-distinctive paths of g_i, i.e., the non-distinctive paths that are prefixes of legal plans to g_i. We define *wcd-*g_i as the maximal *wcd* shared by goal g_i and any other goal.

Definition 5 *The* worst case distinctiveness *of a goal* g_i *in model* D*, denoted by* $\text{wcd-g}_i(D)$ *is:*

$$\text{wcd-g}_i(D) = \max_{\vec{\pi} \in \vec{\Pi}_{nd}(g_i)} |\vec{\pi}|$$

The *wcd* of the entire model can be found by taking the maximum over individual results for *wcd-*$g_i(D)$.

Theorem 2 *Given a* grd-po *model* D,

$$\text{wcd}(D) = \max_i(\text{wcd-g}_i(D)).$$

Proof: Assume to contrary that $\exists \vec{\pi} \in \vec{\Pi}_{nd}(D)$ s.t. $|\vec{\pi}| > \max_i(\textit{wcd-g}_i(D))$. According to Definition 5, $\exists g_i \in \mathcal{G}$ s.t. $\vec{\pi} \in \vec{\Pi}_{nd}(g_i)$ but $|\vec{\pi}| > \textit{wcd-g}_i(D)$, which serves as a contradiction to the definition of $\textit{wcd-g}_i(D)$. ■

A key issue to notice is that in the *partially observable* setting we lose the convenient symmetry that existed in the fully observable setting in which $\textit{wcd-g}_0 = \textit{wcd-g}_1 = \textit{wcd}$ for any pair of goals $\langle g_0, g_1\rangle$. In Example 1, $\textit{wcd}(D) = \textit{wcd-g}_0(D) = \textit{wcd-g}_1(D) = 1$ for the fully observable setting since $\vec{\pi} = \langle \textit{Load}(O_1)\rangle$ is both in $\vec{\Pi}_{nd}(g_0)$ and $\vec{\Pi}_{nd}(g_1)$. In the partially observable setting, $\textit{wcd}(D) = \textit{wcd-g}_0(D) = 8 > \textit{wcd-g}_1(D) = 5$ since the maximal non-distinctive path $\vec{\pi}_{wcd\text{-}ex1} = \langle L(O_1), L(O_2), D(Loc_1, Loc_2), L(O_3), UL(O_1), D(Loc_2, Loc_3), UL(O_2), UL(O_3)\rangle$ satisfies g_0 but not $g1$.

Calculating *wcd*

The baseline method for *wcd* calculation is a breadth first search through the space of paths. A search node (path) is 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 we can solve a goal recognition problem and remove a node whose observable projection satisfies more than one goal.

While the BFS method supports any possible set of legal plans, the need to solve a separate goal recognition problem for each node makes this method inefficient. Next, we present a classical planning compilation used to calculate the *wcd* more efficiently in the case of optimal and boundedly suboptimal (reaching a goal with a bounded cost beyond optimal) legal plans. The compilation finds the maximal non-distinctive path shared by a goal pair. We use it to calculate *wcd*-g_i by pairing g_i to each of the other goals. Relying on Theorem 2 we find the *wcd* of the model by combining individual results to find the maximal value over *wcd*-g_i. For the sake of clarity, the following description focuses on settings where the set of legal plans is the set of optimal plans. The extension to boundedly suboptimal case (Keren, Gal, and Karpas 2015) will follow the compilation description.

The *wcd* of each goal pair $\mathcal{G} = \{g_0, g_1\}$, denoted by *wcd*-$g_{0,1}$, is found by solving a single planning problem P' involving two agents ($agent_0$ and $agent_1$) each with a copy f_i of F. Both agents start at the same initial state (I') but each aim at a goal g_i. The solution to the problem is a plan (for both agents), divided into two parts by a common *exposure point*. The prefix of the plan up to the exposure point represents a non-distinctive path, one that does not reveal the goal of both agents and may consist of actions performed by both agents simultaneously (denoted $A_{0,1}$) in addition to non-observable actions performed by one of the agents (A_i^{ne}). The exposure point is represented by $exposed$, which is a fluent representing the no-cost action *DoExpose* has occurred. After the exposure point the goal of the agent is recognized. The actions, either observed or non-observed, performed by a single agent after the *exposure point* are denoted by A_i^e. Since our objective is to reveal the maximal non-distinctive path of the model, we discount the cost of actions that belong to unexposed prefixes of plans, encouraging the agents to extend the unexposed prefix.

The use of the exposure point is similar to the use of *split* (Keren, Gal, and Karpas 2014; 2015), where agents are encouraged to act together. However, the addition of non-observable actions to the unexposed prefix breaks the symmetry that existed in the fully observable setting. The objective is no longer to find a path that maximizes the number of steps both agents share (actions in $A_{0,1}$). Rather, one of the agents seeks a path that keeps the agent unrecognized by combining non-observable actions (actions in A_i^{ne}) and observable actions that are on legal paths to a different goal (actions in $A_{0,1}$). To reflect this asymmetry we change the objective to allow only one agent (arbitrarily chosen as $agent_0$) to benefit from the discount assigned to performing non-observable actions.

The *latest-expose* compilation (Definition 6) finds *wcd*-$g_{0,1}$ for optimal agents for each pair of goals $\{g_0, g_1\}$.

Definition 6 *For a* grd-po *problem* $D = \langle P, \mathcal{G} = \{g_0, g_1\}, \Pi_{leg}(\mathcal{G})\rangle$ *where* $P = \langle F, I, A = A^o \cup A^{no}\rangle$ *we create a planning problem* $P' = \langle F', I', A', G'\rangle$*, with action costs* C'*, where:*

- $F' = \{f_0, f_1 \mid f \in F\} \cup \{exposed\} \cup \{done_0\}$
- $I' = \{f_0, f_1 \mid f \in I\}\}$
- $A' = A_{0,1} \cup A_i^{ne} \cup A_i^e \cup \{DoExpose\} \cup \{Done_i\}$
  - $A_{0,1} = \{\langle\{f_0, f_1 \mid f \in pre(a)\} \cup \{\neg exposed\},$ $\{f_0, f_1 \mid f \in add(a)\},$ $\{f_0, f_1 \mid f \in del(a)\}\rangle \mid a \in A\}$
  - $A_i^{ne} = \{\langle\{f_i \mid f \in pre(a)\} \cup \{\neg exposed\},$ $\{f_i \mid f \in add(a)\},$ $\{f_i \mid f \in del(a)\}\rangle \mid a \in A^{no}\}$
  - $A_0^e = \{\langle\{f_0 \mid f \in pre(a)\} \cup \{exposed\} \cup \{\neg done_0\},$ $\{f_0 \mid f \in add(a)\},$ $\{f_0 \mid f \in del(a)\}\rangle \mid a \in A\}$
  - $A_1^e = \{\langle\{f_1 \mid f \in pre(a)\} \cup \{exposed\} \cup \{done_0\},$ $\{f_1 \mid f \in add(a)\},$ $\{f_1 \mid f \in del(a)\}\rangle \mid a \in A\}$
  - $Done_0 = \langle exposed, done_0, \emptyset\rangle$
  - $DoExpose = \langle\emptyset, exposed, \emptyset\rangle$
- $G' = \{f_0 | f \in g_0\} \cup \{f_1 | f \in g_1\}$
- $C'(a) = \begin{cases} 2 - \epsilon & \text{if } a \in A_{0,1} \\ 1 - \epsilon & \text{if } a \in A_0^{ne} \\ 1 & \text{if } a \in A_1^{ne}, A_i^e \\ 0 & \text{if } a \in \{DoExpose\} \cup \{Done_0\} \end{cases}$

Note that after agent 0 accomplishes its goal, $Done_0$ is performed, allowing the application of actions in A_1^e until g_1 is achieved. We force agent 1 to wait until agent 0 reaches its goal before starting to act to make the search for a solution to P' more efficient by removing symmetries between different interleaving of agent plans after $DoExpose$ occurs.

Accounting for the bounded-non optimal agent setting (with a cost B_i) requires constraining the path lengths of each agent to be $C^*(g_i) + B_i$ (Keren, Gal, and Karpas 2015). This is achieved by adding an action counting mechanism to the model such that each action a_i^k advances the counter of the corresponding agent. Both counters are initialized to 0 and each agent's goal requires performing $C^*(g_i) + B_i$ actions. We also add $idle_i$ actions, which only advance the counter, to support settings in which an agent cannot reach the goal in *exactly* $C^*(g_i) + B_i$ steps. The cost of $idle_i$ is the same as regular actions and can only be performed after $exposed$ becomes true as to not effect the *wcd* value.

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 $A_{0,1}$, A_i^{ne}, and A_i^e that appear in $\pi_{P'}$. 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 discount that may be collected for performing actions before *DoExpose* occurs, to be lower than the smallest possible diversion from a legal path to any of the agents. Accordingly, whenever $\epsilon < \frac{1}{min(C^*_D(g_0+B_0), C^*_D(g_1+B_1))}$, both agents act optimally in P' (Keren, Gal, and Karpas 2015).

Next, we show that the observable projection of the paths prior to the exposure point is non-distinctive. Given a solution $\pi_{P'}$, $unexposed(\pi_{P'}(g_i))$ denotes the prefix of $\pi_{P'}(g_i)$ prior to the exposure point.

Lemma 2 *$unexposed(\pi_{P'}(g_i))$ is non-distinctive.*

Proof: To show that $unexposed(\pi_{P'}(g_i))$ is non-distinctive we need to show that it satisfies both g_0 and g_1. The compilation guarantees that for any action $a \in$

$unexposed(\pi_{P'}(g_i))$, $a \in A_{0,1}$ or A_i^{no}. According to Definition 1, $op(unexposed(\pi_{P'}(g_i))) = \{\langle a_1...a_n | a \in A_{0,1}\rangle\}$. This means that $op(unexposed(\pi_{P'}(g_0))) = op(unexposed(\pi_{P'}(g_1))$ and according to Definition 3, $op(unexposed(\pi_{P'}(g_i))$ is non-distinctive. ■

Finally, Theorem 3 shows that the optimal solution to P' yields the *wcd*-g_0, thus concluding our proof of correctness.

Theorem 3 *Given a* grd-po *model D with two goals $\langle g_0, g_1\rangle$ and a model P', created according to Definition 6,* wcd-$g_0(D) = |unexposed(\pi^*_{P'}(g_0))|$.

Proof: We have described the bound on ϵ that guarantees that, apart from the no-cost operation *DoExpose* and Done$_0$, the solution to P' consists solely of actions that form a pair of legal paths to each of the goals. Therefore, among the solutions that comply with this condition, $\pi^*_{P'}$ is the one that maximizes the accumulated discount. The compilation guarantees that the only way to accumulate discount is by maximizing the number of actions $agent_0$ performs before the exposure point, therefore $\pi^*_{P'}$ is the solution to P' that maximizes $|unexposed(\pi_{P'}(g_0))|$. Therefore $|unexposed(\pi^*_{P'}(g_0))| =$ *wcd*-$g_0(D)$. ■

In Example 1, $\vec{\pi}_{wcd\text{-}ex1}$ sets the *wcd* to be 8. Using our calculation, $\vec{\pi}_{wcd\text{-}ex1}$ is represented by $unexposed(\pi^*_{P'}(g_0))$ in which $L(O_1)$ is performed by both agents (and belongs to $A_{0,1}$), $L(O_2)$ is unobserved (A_0^{ne}), $D(Loc_1, Loc_2)$ and $L(O_3)$ are performed together ($A_{0,1}$), $UL(O_1)$ is unobserved, $D(Loc_2, Loc_3)$ is performed together($A_{0,1}$), and finally $UL(O_2)$ and $UL(O_3)$ are both unobserved (A_0^{ne}).

Reducing *wcd*

Having formulated the *wcd* measure, we turn to our second objective of finding ways to optimize *wcd* by redesigning the model. Optimization can be achieved using two possible modifications, namely *action removal* and *exposure*. The former disallows actions from being performed while the latter exposes actions by moving them from A^{no} to A^o, *e.g.*, by placing a new sensor.

wcd reduction is performed within a modification budget that represents the constraints to be respected by the reduction method. Given the two possible modifications of a model, we can either provide an integrated budget, B_{total}, or separate budgets $B_{sep} = \langle B_{rem}, B_{sen}\rangle$, where B_{rem} and B_{sen} are the bounds on the number of actions that can be removed and exposed, respectively.

Our objective is to minimize the *wcd* value of the model, subject to a budget constraint. We mark the modifications by a pair $\langle A_{rem}, A_{sen}\rangle$, where A_{rem} and A_{sen} are the disallowed and exposed actions in the transformed model, respectively. In our exploration we assume a uniform cost for the removal and exposure of all actions. In addition, we force the cost of achieving any of the goals to not increase. Both simplifying assumptions can be easily relaxed without major modification to the reduction algorithm.

The reduction is performed using a BFS search that iteratively explores all possible modifications to the model. The initial state is the original model and each successor node introduces a single modification, either exposure or reduction, that was not included in the parent node. A node in the search tree is therefore represented by a pair $\langle A_{rem}, A_{sen}\rangle$. A node is pruned from the search if any of the constraints have been violated or if there are no more actions to add.

The key question remaining is what are the modifications that should be considered at each stage. A naïve approach would be to consider all possible modifications, which is impractical and wasteful. Instead, we focus our attention on modifications that have the potential of reducing *wcd* by either eliminating the *wcd* path (action removal) or by reducing the length of its non-distinctive prefix (exposure). According to Definition 4, we let $\vec{\Pi}_{wcd(D)}$ represent the path set $\vec{\pi}$ s.t. $\vec{\pi} = \underset{\vec{\pi}\in\vec{\Pi}_{nd}(D)}{\operatorname{argmax}} |\vec{\pi}|$. In addition, $\Pi_{wcd(D)}$ represents the set of plans that have a path in $\vec{\Pi}_{wcd(D)}$ as their prefix. It was already shown that the only actions that need to be considered for elimination are the actions that belong to plans in $\Pi_{wcd(D)}$ (Keren, Gal, and Karpas 2014). We show that the only actions that need to be considered for exposure are the non-observable actions that appear in paths in $\vec{\Pi}_{wcd(D)}$.

Theorem 4 *Let D and D_t be two* grd-po *models that are identical except that $A^{no}_{D_t} \subseteq A^{no}_D$. If $\forall a \in A^{no}_D \setminus A^{no}_{D_t}$, $a \notin \vec{\Pi}_{wcd(D)}$ then* wcd$(D) =$ wcd(D_t).

Proof: Theorem 1 assures that any distinctive path in D remains distinctive in D_t and $\vec{\Pi}_{nd}(D_t) \subseteq \vec{\Pi}_{nd}(D)$. Since the *wcd* value of a model is determined by the maximal length of the paths in $\vec{\Pi}_{nd}$ then *wcd*$(D_t) \leq$ *wcd*(D). We need to show that under the specified conditions, the *wcd* cannot decrease in D_t. Assume to the contrary that *wcd*$(D_t) <$ *wcd*(D). This means that there is a non distinctive path $\vec{\pi} \in \vec{\Pi}_{nd}(D)$ s.t. $\vec{\pi}$ is a maximal non-distinctive path in D and is distinctive in D_t (i.e., $\vec{\pi} \in \vec{\Pi}_{wcd(D)}$ and $\vec{\pi} \in \vec{\Pi}_{nd}(D) \setminus \vec{\Pi}_{nd}(D_t)$). Definition 1 guarantees that since $\forall a \in A^{no}_D \setminus A^{no}_{D_t}$, $a \notin \vec{\Pi}_{wcd(D)}$ then $\forall \vec{\pi} \in \vec{\Pi}_{wcd(D)}$, the observable projection did not change $op_D(\vec{\pi}) = op_{D_t}(\vec{\pi})$ and therefore $\vec{\pi} \in \vec{\Pi}_{nd}(D_t)$, which serves as a contradiction. ■

The reduction algorithm creates, for each node, one successor for disallowing each action that appears in $\Pi_{wcd}(D)$ and one successor for exposing each non-observable action in the path $\vec{\pi} \in \vec{\Pi}_{wcd(D)}$ found by the calculation performed at the parent node. To avoid redundant computation, we cache computed actions combination.

In Example 1, disallowing actions is impossible without increasing the optimal costs. However, by exposing $L(O_2)$ by placing a sensor on O_2, *wcd* is reduced to 1, the same as in the fully observable setting.

Empirical Evaluation

Our empirical evaluation has several objectives. Having shown that reduced observability may increase *wcd* we first examine empirically the extent of this effect. In addition, we compare the efficiency of methods proposed for the fully observable (Keren, Gal, and Karpas 2014) and *partially observable* settings. Finally, we evaluate the reduction process as well as the effectiveness of action reduc-

	latest-split		0%		5%		10%		20%	
	Time	*wcd*	Time	*wcd*	Time	*wcd*	Time	*wcd*	Time	*wcd*
GRID	0.32	10.36	0.32	10.36	0.36	10.41	0.35	10.46	0.36	11.1
GRID+	3.53	3.45	8.75	3.45	9.56	3.55	9.64	3.67	9.96	3.84
BLOCK	3.03	2.06	29.2	2.06	33.2	2.12	25.89	2.14	31.01	2.82
LOG	238.5 (0.9)	3.51	165.2 (0.6)	3.71	153.26 (0.31)	3.71	155.48 (0.29)	3.78	191.56 (0.2)	4.1

Table 1: Average running time and values for *wcd* calculation over solved problems for varying non-observable actions ratio

	5 %				10 %				20 %			
	0	4	4:0	0:4	0	4	4:0	0:4	0	4	4:0	0:4
GRID	10.41	9.64	9.71	10.36	10.46	9.34	9.76	10.36	11.1	10.91	11.1	10.91
GRID+	3.55	2.01	2.01	3.55	3.67	1.75	1.87	2.93	3.84	2.6	2.92	3.35
BLOCK	2.12	1.78	1.83	2.12	2.14	1.58	1.64	2.1	2.82	2.15	2.45	2.67
LOG	3.71	3.37	3.44	3.56	3.78	3.26	3.42	3.51	4.1	3.47	3.8	3.67

Table 2: Average *wcd* after reduction for each ratio and budget allocation achieved within allocated time

tion vs. exposure. We describe the datasets and the experiment setup before presenting and discussing the results.

Datasets We use 4 domains of plan recognition (Ramirez and Geffner 2009), namely GRID-NAVIGATION(GRID), IPC-GRID$^+$(GRID+), BLOCK-WORDS(BLOCK), and LOGISTICS(LOG). Each problem description contains a domain description, a template for a problem description without the goal, a set of goals and a set of non-observable actions. For each benchmark we generated a separate *grd* problem for each pair of hypotheses and randomly sampled actions to form the non-observable set creating 3 instances with 5%, 10% and 20% randomly chosen non-observable actions. We tested 216 GRID instances, 660 GRID+ instances, 600 BLOCK instances, and 300 LOG instances. In addition, we created a hand crafted benchmark for the LOGISTICS domain dubbed LOG-Generated, which corresponds to Example 1 where packages load and unload actions are non-observable. This corresponds to real-world settings where satellite imaging can easily track movement of vehicles between locations, but the actual actions performed are obscured from view.

Setup For each problem instance we calculate *wcd* and runtime for the fully observable and *partially observable* settings. For *wcd* reduction we examine the *partially observable* setting with 3 bound settings: an integrated bound of $B_{total} = 4$ and 2 separate bounds $\langle 0, 4\rangle$ and $\langle 4, 0\rangle$, where the first element of each pair represents B_{rem} and the second B_{sen}. We used the Fast Downward planning system (Helmert 2006) running A^* with the LM-CUT heuristic (Helmert and Domshlak 2009). The 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 impact the ratio of non-observable actions has on execution time and *wcd*. The *partially observable* setting is partitioned into the various ratios examined, including a problem with no non-observable actions, which is compared against the values collected for the fully observable setting solved using *latest-split*. For each setting we compare average run time (in seconds) over solved problems. Whenever some of the problems timed-out, we mark in parentheses the ratio of solved instances. For all domains, *wcd* increases with the increase in the ratio of non-observable actions. As for running time, *latest-split* outperforms the equivalent *partially observable* setting for all domains except GRID, for which performance is similar. However, the overhead for adding non-observable actions is negligible. For the LOG-Generated domain the increase in *wcd* was more noticeable, with the average *wcd* increasing from 3.77 in the fully observable setting to 4.87 in the *partially observable* setting.

Table 2 summarizes the results for *wcd* reduction for the *partially observable* setting for each ratio, showing for each budget allocation the average *wcd* reduction achieved within the allocated time (for the LOG domain results refer only to the problems that were successfully solved in the *wcd* calculation stage). The evaluation shows that for all domains *wcd* can be decreased by applying at least one of the modification methods separately, but the most substantial reduction is achieved by combining the methods. Note that this observation is relevant to the entire domain, while individual instances used one modification form. We intend to investigate this phenomenon in future work. For the LOG-Generated domain the results for the reduction went from 4.87 in the original partially observable setting to 3.34, 3.9 and 3.8 for the 4,$\langle 0, 4\rangle$,$\langle 4, 0\rangle$ bound allocations, respectively.

Related Work

Goal recognition design was first introduced by Keren et al. (2014; 2015), offering tools to analyze and solve the *grd* model in fully observable settings. This work relaxes the full observability assumption.

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 of goal recognition design problems with non observable actions into classical planning.

Partial observability in goal recognition has been modeled in various ways (Ramirez and Geffner 2011; Geib and Goldman 2005; Avrahami-Zilberbrand, Kaminka, and Zarosim 2005). In particular, observability can be modeled using a sensor model that includes an observation token for each action (Geffner and Bonet 2013). Note that the *grd-po* model, presented for the *partially observable* setting, can be thought of as one 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.

Conclusions

We presented a goal recognition design model that accounts for partial observability by partitioning the set of actions to observable and non-observable actions. We extend the *wcd* measure and proposed ways to calculate and reduce it.

By accounting for non-observable actions, we increase the model's relevancy to a variety of real-world settings.

Our empirical evaluation shows that non-observable actions typically increases the *wcd* value. In addition, we showed that for all of the domains, *wcd* reduction using both disallowed and exposed actions is preferred over each of the methods separately.

References

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

Avrahami-Zilberbrand, D.; Kaminka, G.; and Zarosim, H. 2005. 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*.

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

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

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

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

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In Gerevini, A.; Howe, A. E.; Cesta, A.; and Refanidis, I., eds., *Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009, Thessaloniki, Greece, September 19-23, 2009*. AAAI.

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

Jarvis, P. A.; Lunt, T. F.; and Myers, K. L. 2004. Identifying terrorist activity with ai plan recognition technology. In *Proceedings of the Sixteenth National Conference on Innovative Applications of Artificial Intelligence (IAAI 2004*, 858–863. AAAI Press.

Kaluza, B.; Kaminka, G. A.; and Tambe, M. 2011. Towards detection of suspicious behavior from multiple observations. In *AAAI Workshop on Plan, Activity, and Intent Recognition (PAIR 2011)*.

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

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

Keren, S.; Gal, A.; and Karpas, E. 2014. Goal recognition design. In *ICAPS Conference Proceedings*.

Keren, S.; Gal, A.; and Karpas, E. 2015. Goal recognition design for non optimal agents. In *Proceedings of the Conference of the American Association of Artificial Intelligence (AAAI 2015)*.

Lesh, N., and Etzioni, O. 1995. A sound and fast goal recognizer. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 1995)*, volume 95, 1704–1710.

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

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

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

Ramirez, M., and Geffner, H. 2011. 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)*, 2009–2014. AAAI Press.