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, 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-
compilation allows us to exploit existing efficient tools for calculating \( \text{wcd} \). Our empirical analysis shows that the compilation allows efficient computation of \( \text{wcd} \).

The third contribution of this work involves finding an optimal set of modifications that can be introduced to the model in order to reduce \( \text{wcd} \). 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 \( \text{wcd} \) 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 \( \text{wcd} \) of a goal recognition design model, consider Figure 1, which demonstrates a simplified setting from the logistics domain. There are 3 locations, \( \text{Loc}_1, \text{Loc}_2 \), and \( \text{Loc}_3 \), a single truck that is initially located at \( \text{Loc}_1 \), and 3 objects that are initially placed such that \( O_1 \) and \( O_2 \) are at \( \text{Loc}_1 \), and \( O_3 \) is at \( \text{Loc}_2 \). 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 \( \text{Loc}_2 \) and \( O_2 \) and \( O_3 \) at \( \text{Loc}_3 \), and \( g_1: O_1 \) at \( \text{Loc}_3 \) and \( O_2 \) at \( \text{Loc}_1 \). Optimal plans are the only valid plans in this example.

In the fully observable setting (see Figure 1(left)) \( \text{wcd} = 1 \) 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 \( D(\text{Loc}_3, \text{Loc}_1) \) 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 \( \text{Loc}_1 \) to \( \text{Loc}_2 \) and then to \( \text{Loc}_3 \) for achieving both goals, the goal is revealed only if \( D(\text{Loc}_3, \text{Loc}_1) \) is performed. This means that \( g_0 \) can be achieved without the system being aware of it \( (\text{wcd} = 8) \). Exposing \( L(O_2) \), by placing a sensor on the object, changes the situation dramatically by reducing \( \text{wcd} \) 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): \( P = (F, I, A, G, C) \) 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 \( a = \langle \text{pre}(a), \text{add}(a), \text{del}(a) \rangle \), which represents the precondition, add, and delete lists respectively, all are subsets of \( F \). An action \( a \) is applicable in state \( s \) if \( \text{pre}(a) \subseteq s \). If action \( a \) is applied in state \( s \), it results in a new state \( s' = (s \setminus \text{del}(a)) \cup \text{add}(a) \). \( C : A \rightarrow \mathbb{R}^+ \) 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 \( \pi \) is \( \sum_{i=1}^{n}(C(a_i)) \). Often, the objective is to find an optimal solution for \( P \), an optimal plan, \( \pi^* \), 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 (\( \text{grd-po} \)) is given as \( D = (P_D, G_D, \Pi_{\text{leg}}(G_D)) \) where:
- \( P_D = \langle \bar{P}_D, \bar{I}_D, \bar{A}_D \rangle \) is a planning domain where \( \bar{A}_D = A_D^o \cup A_D^n \) is a partition of \( A_D \) into observable and non-observable actions, respectively.
- \( G_D \) is a set of possible goals, where each possible goal \( g \in G_D \) is a subset of \( \bar{A}_D \).
- \( \Pi_{\text{leg}}(G_D) = \bigcup_{g \in G_D} \Pi_{\text{leg}}(g) \) is the set of legal plans for each of the goals. A plan is an execution of actions that take the agent from \( I \) to a goal in \( G_D \). A legal plan is one that is allowed under the assumptions made on the behavior of the agent.

The \( \text{grd-po} \) model divides the system description into three elements: system dynamics, defined by \( P_D \) and \( G_D \), agent strategy defined by \( \Pi_{\text{leg}}(G_D) \), and observability defined by the partition of \( A_D \) into \( A_D^o \) and \( A_D^n \). Whenever \( D \) is clear from the context we shall refrain from adding the subscript.

Whereas a plan \( \pi \) is a full execution, a path is a prefix of a legal plan. We denote the set of paths in \( D \) as \( \bar{P}(G_D) \) and the set of paths that are prefixes of plans achieving goal \( g \in G_D \) as \( \bar{P}(g) \). 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 \( \bar{o} = \langle a_1, \ldots, a_n \rangle \) is a sequence of actions \( a_j \in A_o \). For any two action sequences \( \langle a_1, \ldots, a_n \rangle \) and \( \langle a_1, \ldots, a_m \rangle \) the concatenation of the action sequences is denoted by \( \langle a_1, \ldots, a_n \rangle \cdot \langle a_1, \ldots, a_m \rangle \).

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

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

\[
\text{op}_D(\bar{\pi}) = \begin{cases} 
\emptyset & \text{if } \bar{\pi} = \emptyset \\
\langle a_1 \rangle \cdot \text{op}_D(\langle a_2, \ldots, a_n \rangle) & \text{if } \bar{\pi} = \langle a_1, \ldots, a_n \rangle \text{ and } a_1 \in A_D^o \\
\text{op}_D(\langle a_2, \ldots, a_n \rangle) & \text{if } \bar{\pi} = \langle a_1, \ldots, a_n \rangle \text{ and } a_1 \in A_D^n 
\end{cases}
\]

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, \( A_D^{no} = \emptyset, A_D^o = \)
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 \( \vec{\pi} \in \overline{\Pi}(g) \). An observation sequence \( \vec{o} \) satisfies a goal \( g \) if \( \exists \vec{\pi} \in \overline{\Pi}(g) \) s.t. \( \vec{\pi} = o(\vec{\pi}) \).

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 \( o(\vec{\pi}_{ex1}) = \langle D(Loc_1, Loc_2) \rangle \) satisfies both \( g_0 \) and \( g_1 \).

We let \( G_{D}^{\omega}(\vec{\pi}) \) and \( G_{D}^{\omega}(o(\vec{\pi})) \) represent the set of goals that are satisfied by the executed path \( \vec{\pi} \) and its observable projection \( o(\vec{\pi}) \), respectively. The distinction the grd-po model creates between \( G_{D}^{\omega}(\vec{\pi}) \) and \( G_{D}^{\omega}(o(\vec{\pi})) \) 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 \( A^i \) 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 \( A_{D}^{no} \subseteq A_{D'}^{no} \). For any \( \vec{\pi} \in \overline{\Pi}(G_{D}) \),

\[ G_{D}^{\omega}(o(\vec{\pi})) \subseteq G_{D'}^{\omega}(o(\vec{\pi})) \]

Proof: According to Definition 1, the observation sequence generated by the execution of a path \( \vec{\pi} \) is \( o(\vec{\pi}) = \langle a_1, \ldots, a_n \rangle \) where \( a_i \in A^i \). If \( \forall a \in \vec{\pi}, a \notin A_{D'}^{no} \), then \( o(\vec{\pi}) = o(\vec{\pi}') \) since none of the actions in \( \vec{\pi} \) changed their observability property. Therefore, \( G_{D}^{\omega}(o(\vec{\pi})) = G_{D'}^{\omega}(o(\vec{\pi})) \). Otherwise, \( \exists a_i \in \vec{\pi}, 1 \leq i \leq n \) s.t. \( a_i \in A_{D}^{no} \setminus A_{D'}^{no} \). According to Definition 1,

\[ o(\vec{\pi}) = o(\vec{\pi}') \cdot a_i \cdot o(\vec{\pi}) \]

where \( o(\vec{\pi}') = o(\vec{\pi}) \langle a_1, \ldots, a_i-1 \rangle \cdot o(\vec{\pi}) \langle a_i+1, \ldots, a_n \rangle \). For any path \( \vec{\pi} \) identical to \( \vec{\pi} \), except that \( a_i \) is replaced by a possibly empty sequence of non-observable actions, \( o(\vec{\pi}) \neq o(\vec{\pi}') \) but \( o(\vec{\pi}') = o(\vec{\pi}') \). Since \( \vec{\pi} \) may lead to a different goal than \( \vec{\pi}, G_{D}^{\omega}(o(\vec{\pi})) \subseteq G_{D'}^{\omega}(o(\vec{\pi})) \).

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 \( o(\vec{\pi}) \) 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 \( \overline{\Pi}_{nd}(D) \) and define wcd as maximal length of paths in \( \overline{\Pi}_{nd}(D) \).

Definition 4 The worst case distinctiveness of a model \( D \), denoted by wcd\( (D) \), is:

\[ \text{wcd}(D) = \max_{\vec{\pi} \in \overline{\Pi}_{nd}(D)} |\vec{\pi}| \]

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 \( \vec{\pi} \in \overline{\Pi}(G) \) that satisfy at least one goal (1 ≤ |\( G_{D}^{\omega}(\vec{\pi}) \)| and whose observable projection satisfies at least two goals (2 ≤ |\( G_{D}^{\omega}(o(\vec{\pi})) \)|). 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 \( \overline{\Pi}_{nd}(g) \) represent the non-distinctive paths of \( g \) i.e., the non-distinctive paths that are prefixes of legal plans to \( g \) we define wcd\( _g \) as the maximal wcd shared by goal \( g \) and any other goal.

Definition 5 The worst case distinctiveness of a goal \( g \), in model \( D \), denoted by wcd\( _g(D) \), is:

\[ \text{wcd}_g(D) = \max_{\vec{\pi} \in \overline{\Pi}_{nd}(g)} |\vec{\pi}| \]

The wcd of the entire model can be found by taking the maximum over individual results for wcd\( _g(D) \).

Theorem 2 Given a grd-po model \( D \),

\[ \text{wcd}(D) = \max_{g} \text{wcd}_g(D) \]

Proof: Assume to contrary that \( \exists \vec{\pi} \in \overline{\Pi}_{nd}(D) \) s.t. \( |\vec{\pi}| > \max_{g} \text{wcd}_g(D) \). According to Definition 5, \( \exists g_i \in \gamma \) s.t.

\[ \vec{\pi} \in \overline{\Pi}_{nd}(g_i) \text{ but } |\vec{\pi}| > \text{wcd}_g(D) \]

which serves as a contradiction to the definition of wcd\( _g(D) \).

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 wcd\( _g(D) = wcd_g(D) = wcd_g(D) \) for any pair of goals \( g_0, g_1 \). In Example 1, \( wcd(D) = wcd_g(D) = wcd_g(D) = 1 \) for the fully observable setting since \( \vec{\pi} = \langle \text{Load}(O_1) \rangle \) is both in \( \overline{\Pi}_{nd}(g_0) \) and \( \overline{\Pi}_{nd}(g_1) \). In the partially observable setting, \( wcd(D) = wcd_g(D) = 8 > wcd_g(D) = 5 \) since the maximal non-distinctive path \( \vec{\pi}_{wcd-ex1} = \langle L(O_1), L(O_2), D(Loc_1, Loc_2), L(O_3), U(L(O_1), D(Loc_2, Loc_3), U(L(O_2), U(L(O_3)) \rangle \) satisfies \( g_0 \) but not \( g_1 \).

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

3154
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 \( \text{wcd} \) 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 \( \text{wcd-g}_t \) by pairing \( g_i \) to each of the other goals. Relying on Theorem 2 we find the \( \text{wcd} \) of the model by combining individual results to find the maximal value over \( \text{wcd-g}_t \). 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 \( \text{wcd} \) of each goal pair \( G = \{g_0, g_1\} \), denoted by \( \text{wcd-g}_{0,1} \), is found by solving a single planning problem \( P' \) involving two agents (\( \text{agent}_0 \) and \( \text{agent}_1 \)) 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 \( (A_{n}^{o}) \). The exposure point is represented by \( \text{exposed} \), which is a fluent representing the no-cost action \( \text{DoExpose} \) 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 \( A_{e}^{i} \). 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 \textit{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 \( A_{n}^{o} \)) 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 \( \text{agent}_0 \)) to benefit from the discount assigned to performing non-observable actions.

The \textit{latest-expose} compilation (Definition 6) finds \( \text{wcd-g}_{0,1} \) for optimal agents for each pair of goals \( \{g_0, g_1\} \).

\textbf{Definition 6} For a \textit{grd-po} problem \( D = \langle P, G = \{g_0, g_1\}, \Pi_{g, G} \rangle \) where \( P = \langle F, I, A = A^{o} \cup A^{n} \rangle \) we create a planning problem \( P' = \langle F', I', A', G' \rangle \), with action costs \( C' \), where:

\[ f' = \{f_0, f_1 | f \in F \} \cup \{\text{exposed}\} \cup \{\text{done}\} \]

\[ I' = \{f_0, f_1 | f \in I \} \]

\[ A' = A_{0,1} \cup A^{n}_{0} \cup A' \cup \{\text{DoExpose}\} \cup \{\text{Done}\} \]

\[ A_{0,1} = \{(f_0, f_1 | f \in \text{pre}(a)) \cup \{\text{exposed}\}, \}

\[ \{f_0, f_1 | f \in \text{add}(a)\}, \]

\[ \{f_0, f_1 | f \in \text{del}(a)\} | a \in A \} \]

\[ A^{n}_{0} = \{(f_1 | f \in \text{pre}(a)) \cup \{\text{exposed}\}, \}

\[ \{f_1 | f \in \text{add}(a)\}, \]

\[ \{f_1 | f \in \text{del}(a)\} | a \in A^{n}_{0} \} \]

\[ A^{n}_{1} = \{(f_0 | f \in \text{pre}(a)) \cup \{\text{exposed}\} \cup \{\text{done}\}, \}

\[ \{f_0 | f \in \text{add}(a)\}, \]

\[ \{f_0 | f \in \text{del}(a)\} | a \in A \} \]

\[ \text{Done} = \{\text{exposed}, \text{done}, \emptyset\} \]

\[ \text{DoExpose} = \{\emptyset, \text{exposed}, \emptyset\} \]

\[ G' = \{f_0 | f \in g_0 \} \cup \{f_1 | f \in g_1 \} \]

\[ C'(a) = \begin{cases} 2 - \epsilon & \text{if } a \in A_{0,1} \\ 1 - \epsilon & \text{if } a \in A^{n}_{0} \\ 1 & \text{if } a \in A^{n}_{0} \cup A^{*} \\ 0 & \text{if } a \in \{\text{DoExpose}\} \cup \{\text{Done}\} \end{cases} \]

Note that after agent 0 accomplishes its goal, \( \text{Done}_0 \) is performed, allowing the application of actions in \( A_{1}^{*} \) 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 \( \text{DoExpose} \) 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 \( a^{*} \) 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 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 actions is the same as regular actions and can only be performed after \( \text{exposed} \) becomes true as to not effect the \( \text{wcd} \) value.

Given a solution \( \pi_{p'} \) to \( P' \), we mark the \textit{projection} of \( \pi_{p'} \) on each agent \( i \) as \( \pi_{p'}(g_i) \), which includes all actions in \( A_{0,1}, A_{i}^{n}, \) and \( A_{i}^{*} \) 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 \( \epsilon \), the discount that may be collected for performing actions before \( \text{DoExpose} \) occurs, to be lower than the smallest possible diversion from a legal path to any of the agents. Accordingly, whenever \( \epsilon < \min\{C_{p}'(g_0 + B_0), C_{p}'(g_1 + B_1)\} \), 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 \( \pi_{p'} \), \( \text{unexposed}(\pi_{p'}(g_i)) \) denotes the prefix of \( \pi_{p'}(g_i) \) prior to the exposure point.

\textbf{Lemma 2} \( \text{unexposed}(\pi_{p'}(g_i)) \) is non-distinctive.

\textbf{Proof:} To show that \( \text{unexposed}(\pi_{p'}(g_i)) \) is non-distinctive we need to show that it satisfies both \( g_0 \) and \( g_1 \). The compilation guarantees that for any action \( a \in \)
Finally, Theorem 3 shows that the optimal solution to $P'$ yields the wcd-$\pi_0$, 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-$\pi_0(D) = [\text{unexposed}(\pi_{P'}(g_0))]$.

**Proof:** We have described the bound on $\epsilon$ that guarantees that, apart from the no-cost operation DoExpose and Done0, 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 algorithm guarantees that the only way to accumulate discount is by maximizing the number of actions agent0 performs before the exposure point, therefore $\pi_{P'}$ is the solution to $P'$ that maximizes $[\text{unexposed}(\pi_{P'}(g_0))]$. Therefore $[\text{unexposed}(\pi_{P'}(g_0))] = \text{wcd-}\pi_0(D)$.

In Example 1, $\bar{\pi}_{\text{wcd-r,ex-1}}$ sets the wcd to be 8. Using our calculation, $\bar{\pi}_{\text{wcd-r,ex-1}}$ is represented by $[\text{unexposed}(\pi_{P'}(g_0))]$ in which $L(O_1)$ is performed by both agents (and belongs to $A_{o,1}$), $L(O_2)$ is unobserved ($A_{o}^{\text{no}}$), $D(Loc_1, Loc_2)$ and $L(O_3)$ are performed together ($A_{o,1}$), $UL(O_3)$ is unobserved, $D(Loc_2, Loc_3)$ is performed together ($A_{o,1}$), and finally $UL(O_2)$ and $UL(O_3)$ are both unobserved ($A_{o}^{\text{no}}$).

**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 $A^{\text{no}}$ 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, $B_{\text{total}}$, or separate budgets $B_{\text{rep}} = (B_{\text{rem}}, B_{\text{sen}})$, where $B_{\text{rem}}$ and $B_{\text{sen}}$ 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 $A_{rem}, A_{sen}$, 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 $A_{rem}, A_{sen}$. 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 $\Pi_{\text{wcd}}(D)$ represent the path set $\bar{\pi}$ s.t. $\bar{\pi} = \text{argmax} |\bar{\pi}|$. In addition, $\Pi_{\text{wcd}}(D)$ represents the set of plans that have a path in $\Pi_{\text{wcd}}(D)$ 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 $\Pi_{\text{wcd}}(D)$ (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 $\Pi_{\text{wcd}}(D)$.

**Theorem 4** Let $D$ and $D_1$ be two grd-po models that are identical except that $A_{o}^{\text{no}} \subseteq A_o$. If $\forall a \in A_{o}^{\text{no}} \setminus A_{o}^{\text{no}}$, $a \not\in \Pi_{\text{wcd}}(D)$ then wcd$(D) = \text{wcd}(D_1)$.

**Proof:** Theorem 1 assures that any distinctive path in $D$ remains distinctive in $D_1$ and $\Pi_{\text{nd}}(D_1) \subseteq \Pi_{\text{nd}}(D)$. Since the wcd value of a model is determined by the maximal length of the paths in $\Pi_{\text{nd}}$ then wcd$(D_1) \leq \text{wcd}(D)$. We need to show that under the specified conditions, the wcd cannot decrease in $D_1$. Assume to the contrary that wcd$(D_1) < \text{wcd}(D)$. This means that there is a non distinctive path $\bar{\pi} \in \Pi_{\text{nd}}(D)$ s.t. $\bar{\pi}$ is a maximal non-distinctive path in $D$ and is distinctive in $D_1$ (i.e., $\bar{\pi} \in \Pi_{\text{wcd}}(D)$ and $\bar{\pi} \in \Pi_{\text{nd}}(D) \setminus \Pi_{\text{nd}}(D_1)$). Definition 1 guarantees that since $\forall a \in A_{o}^{\text{no}} \setminus A_{o}^{\text{no}}$, $a \not\in \Pi_{\text{wcd}}(D_1)$ then $\forall \bar{\pi} \in \Pi_{\text{wcd}}(D_1)$, the observable projection did not change $op_D(\bar{\pi}) = op_{D_1}(\bar{\pi})$ and therefore $\bar{\pi} \not\in \Pi_{\text{nd}}(D_1)$, which serves as a contradiction.

The reduction algorithm creates, for each node, one successor for disallowing each action that appears in $\Pi_{\text{wcd}}(D)$ and one successor for exposing each non-observable action in the path $\bar{\pi} \in \Pi_{\text{wcd}}(D)$ 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-
### Related Work

Goal recognition design was first introduced by Keren et al. (2014; 2015), offering tools to analyze and solve the grid 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. The full observability assumption 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.

### 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


