# New Metrics and Algorithms for Stochastic Goal Recognition Design Problems*

**Christabel Wayllace**
Computer Science Department
New Mexico State University
Las Cruces, NM 88003, USA
cwayllac@cs.nmsu.edu

**Ping Hou**
Computer Science Department
New Mexico State University
Las Cruces, NM 88003, USA
phou@cs.nmsu.edu

**William Yeoh**
Computer Science Department
New Mexico State University
Las Cruces, NM 88003, USA
wyeoh@cs.nmsu.edu

## Abstract

*Goal Recognition Design* (GRD) problems involve identifying the best ways to modify the underlying environment that agents operate in, typically by making a subset of feasible actions infeasible, in such a way that agents are forced to reveal their goals as early as possible. The *Stochastic GRD* (S-GRD) model is an important extension that introduced stochasticity to the outcome of agent actions. Unfortunately, the *worst-case distinctiveness* ($wcd$) metric proposed for S-GRDs has a formal definition that is inconsistent with its intuitive definition, which is the maximal number of actions an agent can take, in the expectation, before its goal is revealed. In this paper, we make the following contributions: (1) We propose a new $wcd$ metric, called *all-goals* $wcd$ ($wcd_{ag}$), that remedies this inconsistency; (2) We introduce a new metric, called *expected-case distinctiveness* ($ecd$), that weighs the possible goals based on their likelihood of being the true goal; (3) We provide theoretical results comparing these different metrics as well as the complexity of computing them optimally; and (4) We describe new efficient algorithms to compute the $wcd_{ag}$ and $ecd$ values.

## 1 Introduction

Discovering the objective of an agent based on observations of its behavior is a problem that has interested both AI and psychology researchers for many years [Schmidt *et al.*, 1978; Kautz, 1987]. In AI, this problem is known as *goal recognition* or, more generally, *plan recognition* [Sukthankar *et al.*, 2014], and it has been used to model a number of applications ranging from software personal assistants [Oh *et al.*, 2010; 2011a; 2011b]; robots that interact with humans in social settings such as homes, offices, and hospitals [Tavakkoli *et al.*, 2007; Kelley *et al.*, 2012]; intelligent tutoring systems that recognize sources of confusion or misunderstanding in students through their interactions with the system [McQuiggan *et al.*, 2008; Johnson, 2010; Lee *et al.*, 2012; Min *et al.*, 2014]; and security applications that recognize the plan or goal of terrorists [Jarvis *et al.*, 2005].

Researchers have recently introduced a newly formulated problem that is aimed towards helping the objective of goal recognition by performing an offline analysis of the model. The problem, proposed by Keren *et al.* [2014], is called *goal recognition design* (GRD) and it is intended to reduce the complexity of the online goal recognition task by modifying the underlying environment that the agent operates in. The goal is to find a subset of modifications that *forces the agent to reveal its goal as early as possible*. This problem finds itself relevant in many of the same applications of goal recognition because, typically, the underlying environment can be easily modified. Buoyed by this new model, researchers have extended the GRD problem to a number of extensions with accompanying algorithms to solve them [Keren *et al.*, 2015; 2016a; 2016b; 2017; Son *et al.*, 2016]. One of these extensions is the *Stochastic GRD* (S-GRD) problem, where the outcomes of the agent's actions are stochastic [Wayllace *et al.*, 2016]. In all these problems, the "goodness" of a solution is measured using the *worst-case distinctiveness* ($wcd$) metric, which, intuitively, measures the maximal number of actions an agent can take before its goal is revealed.

Unfortunately, the formal $wcd$ definition proposed by Wayllace *et al.* [2016] for S-GRDs is inconsistent with this intuitive definition; we provide an example later in Section 3.1 that highlights this inconsistency. Based on this finding, we propose a new $wcd$ metric, called *all-goals* $wcd$ ($wcd_{ag}$), that remedies this inconsistency as well as new efficient algorithms to compute them. Additionally, another deficiency of the existing $wcd$ metric for all GRD variants is its implicit assumption that there is no prior information about the agent's true goal. While this assumption is reasonable in many problems, it may be the case that this information is available in some applications, thereby allowing the observer to assign different weights to the possible goals. We thus propose a new metric for S-GRDs, called the *expected-case distinctiveness* ($ecd$), that weighs the length of a path to a goal by the likelihood of that goal being the true goal.[1] This new $ecd$

*This research is partially supported by NSF grant 1550662. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, or the U.S. government.

[1] While we only describe this metric for S-GRDs, it can be easily extended for other GRD variants as well.

metric will thus allow practitioners to better incorporate domain knowledge into the S-GRD model and, combined with the updated $wcd$ metric, will allow them to better understand the tradeoffs between different S-GRD solutions.

## 2 Background

### 2.1 Markov Decision Process (MDP)

A *Stochastic Shortest Path Markov Decision Process* (SSP-MDP) [Mausam and Kolobov, 2012] is represented as a tuple $\langle \mathbf{S}, s_0, \mathbf{A}, \mathbf{T}, \mathbf{C}, \mathbf{G} \rangle$. It consists of a set of states $\mathbf{S}$; a start state $s_0 \in \mathbf{S}$; a set of actions $\mathbf{A}$; a transition function $\mathbf{T} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \to [0, 1]$ that gives the probability $T(s, a, s')$ of transitioning from state $s$ to $s'$ when action $a$ is executed; a cost function $\mathbf{C} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \to \mathbb{R}$ that gives the cost $C(s, a, s')$ of executing action $a$ in state $s$ and arriving in state $s'$; and a set of goal states $\mathbf{G} \subseteq \mathbf{S}$. The goal states are terminal, that is, $T(g, a, g) = 1$ and $C(g, a, g) = 0$ for all goal states $g \in \mathbf{G}$ and actions $a \in \mathbf{A}$.

An SSP-MDP must also satisfy the following two conditions: (1) There must exist a *proper policy*, which is a mapping from states to actions with which an agent can reach a goal state from any state with probability 1. (2) Every *improper policy* must incur an accumulated cost of $\infty$ from all states from which it cannot reach the goal with probability 1. In this paper, we will focus on SSP-MDPs and will thus use the term MDPs to refer to SSP-MDPs. A "solution" to an MDP is a policy $\pi$, which maps states to actions. Solving an MDP is to find an optimal policy, that is, a policy with the smallest expected cost. Finally, we use the term "optimal actions" to refer to actions in an optimal policy.

### 2.2 Value Iteration (VI) and Topological VI (TVI)

*Value Iteration* (VI) [Bellman, 1957] is one of the fundamental algorithms to find an optimal policy. It uses a value function $V$ to represent expected costs. The expected cost of an optimal policy $\pi^*$ for the starting state $s_0 \in \mathbf{S}$ is the expected cost $V(s_0)$, and the expected cost $V(s)$ for all states $s \in \mathbf{S}$ is calculated using the Bellman equation [Bellman, 1957]:

$$V(s) = \min_{a \in \mathbf{A}} \sum_{s' \in \mathbf{S}} T(s, a, s') \big[ C(s, a, s') + V(s') \big] \qquad (1)$$

The action chosen by the policy for each state $s$ is then the one that minimizes $V(s)$.

VI suffers from a limitation that it updates each state in every iteration even if the expected cost of some states have converged. *Topological VI* (TVI) [Dai *et al.*, 2011] addresses this limitation by repeatedly updating the states in only one *strongly connected component* (SCC) until their values converge before updating the states in another SCC. Since the SCCs form a directed acyclic graph, states in an SCC only affect the states in upstream SCCs. Thus, by choosing the SCCs in reverse topological sort order, it no longer needs to consider SCCs whose states have converged in a previous iteration.

### 2.3 Goal Recognition Design (GRD) and Stochastic GRD (S-GRD)

A *Goal Recognition Design* (GRD) problem [Keren *et al.*, 2014] is represented as a tuple $P = \langle D, \mathbf{G} \rangle$, where $D = \langle \mathbf{S}, s_0, \mathbf{A}, \mathbf{T}, \mathbf{C} \rangle$ captures the domain information and $\mathbf{G}$ is a set of possible goal states of the agent. The elements in the tuple $D$ are as they are described in MDPs, except that the transition function $T$ is deterministic and the cost function $C$ is restricted to positive costs.[2] In this paper, we assume that the cost of all actions is 1 for simplicity.

The *worst case distinctiveness* ($wcd$) of problem $P$ is the length of a longest sequence of actions $\pi = \langle a_1, \ldots, a_k \rangle$ that is the prefix in *cost-minimal* plans $\pi^*_{g_1}$ and $\pi^*_{g_2}$ to distinct goals $g_1, g_2 \in \mathbf{G}$. Intuitively, as long as the agent executes $\pi$, it does not reveal its goal to be either $g_1$ or $g_2$.

The objective in GRD is to find a subset of actions such that if they are removed from the domain, the $wcd$ of the resulting problem is minimized. This optimization problem is subject to the requirement that the cost of cost-minimal plans to achieve each goal $g \in \mathbf{G}$ is the same before and after removing the subset of actions.

A *Stochastic Goal Recognition Design* (S-GRD) problem [Wayllace *et al.*, 2016] is an extension of a GRD problem that assumes the actions executed by the agent have stochastic outcomes. It is represented as a tuple $P = \langle D, \mathbf{G} \rangle$, where, like in GRDs, $D = \langle \mathbf{S}, s_0, \mathbf{A}, \mathbf{T}, \mathbf{C} \rangle$ captures the domain information and $\mathbf{G}$ is a set of possible goal states of the agent. The elements in the tuple $D$ are as they are described in GRDs, except that the transition function $T$ is now stochastic. The *worst case distinctiveness* ($wcd$) of problem $P$ is the largest expected cost incurred by the agent over all non-distinctive policy prefixes. A non-distinctive policy prefix is an optimal policy common to a pair of goals.

Like in GRDs, the objective in S-GRD is to find a subset of actions $\hat{\mathbf{A}}^* \subset \mathbf{A}$ such that if they are removed from the set of actions $\mathbf{A}$, then the $wcd$ of the resulting problem is minimized. This optimization problem is subject to the requirement that the expected cost of the optimal policies to achieve each goal $g \in \mathbf{G}$ is the same before and after removing the subset of actions and that the number of reduced actions is less than or equal to a user-defined parameter $k$. More specifically, the objective is to find:

$$\begin{aligned} \hat{\mathbf{A}}^* &= \operatorname*{argmin}_{\hat{\mathbf{A}} \subset \mathbf{A}} wcd(\hat{P}) \\ \text{subject to } & V_{\pi^*_g}(s_0) = V_{\hat{\pi}^*_g}(s_0) \qquad \forall g \in \mathbf{G} \\ & |\hat{\mathbf{A}}^*| \leq k \end{aligned} \qquad (2)$$

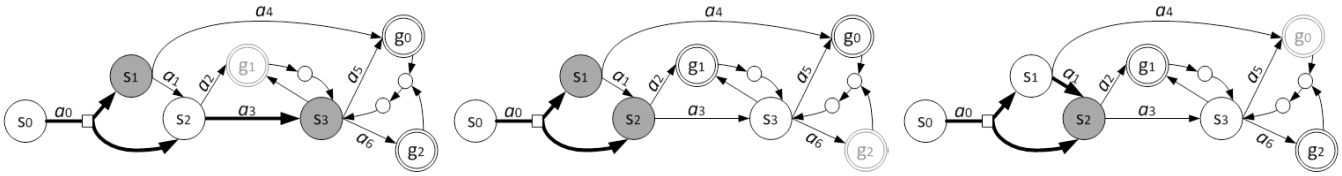
where $\hat{P} = \langle \hat{D}, \mathbf{G} \rangle$ is the problem with domain $\hat{D} = \langle \mathbf{S}, s_0, \mathbf{A} \setminus \hat{\mathbf{A}}, \mathbf{T}, \mathbf{C} \rangle$ after removing actions $\hat{\mathbf{A}}$, $\pi^*_g$ is an optimal policy to achieve goal $g$ in the original problem $P$, and $\hat{\pi}^*_g$ is an optimal policy to achieve goal $g$ in problem $\hat{P}$.

## 3 Redefining the *wcd* Metric for S-GRDs

The intuition behind the *worst case distinctiveness* ($wcd$) is that it measures the longest path (i.e., a path with the largest cost) an agent can take without revealing its goal.

Figure 1 illustrates a small example, where the agent starts at state $s_0$ and has one of three possible goals $g_0$, $g_1$, or $g_2$.

[2] The domain information $D$ was originally described by Keren *et al.* [2014] using a classical planning model [Geffner and Bonet, 2013].




(a) $wcd$ for Goals $g_0$ and $g_2 = 1.5$ (b) $wcd$ for Goals $g_0$ and $g_1 = 1$ (c) $wcd$ for Goals $g_1$ and $g_2 = 1.5$

Figure 2: $wcd$ Example

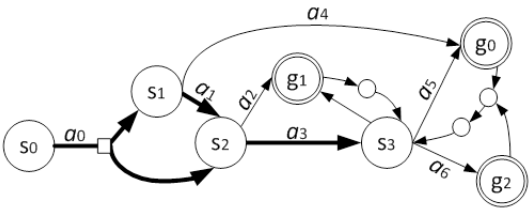



Figure 1: Example

All actions are deterministic except for action $a_0$ out of $s_0$, which can transition to either $s_1$ or $s_2$ with equal probability. All actions also have the same cost of 1. In this example, there are two possible paths, each with two actions, that the agent can take before it has to reveal its goal in the third action. The two paths are denoted by bold arrows in the figure.

For the first path, starting at $s_0$, the agent has to take action $a_0$ as it is the only action available. If it transitions to $s_1$, then it can take action $a_1$ to transition to $s_2$. At this point, its goal can be either $g_1$ or $g_2$. From $s_2$, it reveals its goal to be $g_1$ if it takes action $a_2$ and goal $g_2$ if it takes action $a_3$. Note that its goal cannot be $g_0$ as it would otherwise have taken action $a_4$ instead of $a_1$ when it was in $s_1$.

For the second path, starting at $s_0$, the agent may transition to $s_2$ after taking action $a_0$. Then, it can take action $a_3$ to transition to $s_3$, at which point its goal can be either $g_0$ or $g_2$. From $s_3$, it reveals its goal to be $g_0$ if it takes action $a_5$ and goal $g_2$ if it takes action $a_6$.

Note that the cost of both paths is 2 since they both have two actions each. Consequently, the $wcd$ of this problem should be 2 intuitively. However, using the $wcd$ definition proposed by Wayllace *et al.* [2016], the $wcd$ is 1.5! Therefore, we now describe the reason for this inconsistency and propose a new updated definition for $wcd$ that is more in line with its intuitive definition.

Wayllace *et al.* [2016] defined the $wcd$ as "the largest expected cost to reach a boundary state from the start state over all possible non-distinctive policy prefixes," where for a pair of distinct goals $g_i$ and $g_j$, (1) a non-distinctive policy prefix is an optimal policy common to those two goals, and (2) boundary states are states where an agent, through its next action, will reveal its goal to be either one of those two goals.

Figure 2(a) shows the same example again but with the non-distinctive policy prefixes for pairs of goals $\langle g_0, g_2 \rangle$. Here, $s_1$ and $s_3$ are boundary states (shaded in grey), and the expected cost of this policy prefix is 1.5 ($= 0.5 * 1 + 0.5 * (1 + 1)$). Figures 2(b) and 2(c) show the same example but for pairs of goals $\langle g_0, g_1 \rangle$ and $\langle g_1, g_2 \rangle$, respectively. The expected cost is 1 for the former and 1.5 for the latter. The $wcd$ of this problem is thus 1.5, the largest expected cost over all pairs of goals.

This definition fails to capture the intuitive paths of cost 2 because it considers only *pairs* of goals in its $wcd$, non-distinctive policy prefix, and boundary state definitions. Instead, it needs to consider a tuple of *all* goals in its definitions. For example, for the pair of goals $\langle g_0, g_2 \rangle$, $s_1$ is a boundary state and $s_2$ is a regular state in a non-distinctive policy prefix (i.e., the agent reveals its goal when taking an action from $s_1$ but may conceal its goal when taking an action from $s_2$). In contrast, for the pair $\langle g_1, g_2 \rangle$, the situation is reversed, where $s_2$ is a boundary state and $s_1$ is a regular state in a non-distinctive policy prefix. Therefore, when considering all three goals, neither $s_1$ nor $s_2$ are actual boundary states. In other words, the agent can still conceal its goal to be either $g_1$ or $g_2$ in state $s_1$ or either $g_0$ or $g_2$ in state $s_2$. The definition of Wayllace *et al.* [2016] fails to capture this scenario.

### 3.1 All-Goals *wcd* ($wcd_{ag}$)

We now propose a new $wcd$ definition, called *All-Goals wcd* ($wcd_{ag}$), that captures the above scenario and is more consistent with its intuitive definition.

In addition to the need to consider all goals at the same time instead of pairs of goals, we also make another key observation: that the set of possible goals for a particular state can differ based on the observed path of the agent to that state. Using Figure 1 as an example again, if the agent arrives at state $s_3$ through path $\langle s_0, s_2, s_3 \rangle$, then its goal is either $g_0$ and $g_2$. However, if it arrives at state $s_3$ through path $\langle s_0, s_1, s_2, s_3 \rangle$, then its goal is definitely $g_2$. This observation causes a challenge in that, unlike the previous $wcd$ definition, the set of possible goals of the agent in this new definition is no longer Markovian as it depends on the entire history of states visited.

We address this challenge by modeling the problem using augmented MDPs instead of regular MDPs, where each augmented state in the augmented MDP is a tuple $\langle s, \mathbf{G}' \rangle$ that consists of the actual state $s$ of the MDP and the set of goals $\mathbf{G}' \subseteq \mathbf{G}$ that are possible goals for that state. Therefore, in the scenario above, we will have two different augmented states $\langle s_3, \{g_0, g_2\} \rangle$ and $\langle s_3, \{g_2\} \rangle$, and different actions can be attributed to the two augmented states. In this augmented MDP formulation, the set of possible goals of the agent is now Markovian again.

We now describe the other elements of this augmented MDP, defined as the tuple $\langle \tilde{\mathbf{S}}, \tilde{s}_0, \tilde{\mathbf{A}}, \tilde{\mathbf{T}}, \tilde{\mathbf{C}}, \tilde{\mathbf{G}} \rangle$, and how to construct them from S-GRDs, whose domain information is

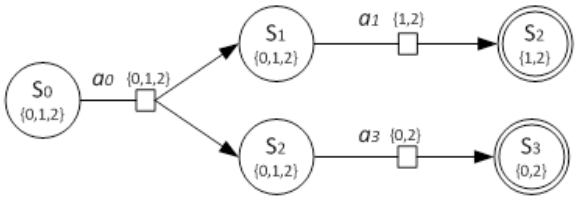



Figure 3: Reachable Augmented MDP

modeled as regular MDPs. The augmented start state is now $\tilde{s}_0 = \langle s_0, \mathbf{G} \rangle$, where $s_0$ is the start state and $\mathbf{G}$ is the set of all possible goals in the S-GRD problem. Each augmented action $\tilde{a} \in \tilde{\mathbf{A}}$ is a tuple $\langle a, \mathbf{G}' \rangle$, where $a \in \mathbf{A}$ is an action in the regular MDP and $\mathbf{G}'$ is the set of all goals for which that action is an *optimal action*. The new transition function $\tilde{\mathbf{T}} : \tilde{\mathbf{S}} \times \tilde{\mathbf{A}} \times \tilde{\mathbf{S}} \rightarrow [0, 1]$ gives the probability $\tilde{T}(\tilde{s}, \tilde{a}, \tilde{s}')$ of transitioning from augmented state $\tilde{s}$ to $\tilde{s}'$ when augmented action $\tilde{a}$ is executed. This transition probability equals the transition probability $\tilde{T}(\tilde{s}, \tilde{a}, \tilde{s}') = T(s, a, s')$ in the regular MDP for $\tilde{s} = \langle s, \mathbf{G}' \rangle$, $\tilde{a} = \langle a, \mathbf{G}'' \rangle$, and $\tilde{s}' = \langle s', \mathbf{G}' \cap \mathbf{G}'' \rangle$ if $|\mathbf{G}' \cap \mathbf{G}''| > 1$ and equals 0 otherwise. The cost function $\tilde{C} : \tilde{S} \times \tilde{A} \times \tilde{S} \rightarrow \mathbb{R}^+$ gives the cost $\tilde{C}(\tilde{s}, \tilde{a}, \tilde{s}')$ of executing action $\tilde{a}$ in augmented state $\tilde{s}$ and arriving in $\tilde{s}'$. This cost equals the cost $\tilde{C}(\tilde{s}, \tilde{a}, \tilde{s}') = C(s, a, s')$ for the same case for the transition probabilities above. Finally, the augmented goal states $\tilde{\mathbf{G}} \subseteq \tilde{\mathbf{S}}$ are those augmented states $\langle s, \mathbf{G}' \rangle$ that are boundary states – in other words, any augmented action from those states will transition to an augmented state $\langle s', \mathbf{G}''' \rangle$ with one goal or no goals (i.e., $|\mathbf{G}'''| \leq 1$) in the regular MDP. We provide the formal definition for augmented MDPs below.

**Definition 1 (Augmented MDP)** *For an S-GRD problem $P = \langle D, \mathbf{G} \rangle$ with domain information $D = \langle \mathbf{S}, s_0, \mathbf{A}, \mathbf{T}, \mathbf{C} \rangle$, an augmented MDP is defined by a tuple $\langle \tilde{\mathbf{S}}, \tilde{s}_0, \tilde{\mathbf{A}}, \tilde{\mathbf{T}}, \tilde{\mathbf{C}}, \tilde{\mathbf{G}} \rangle$ that consists of the following:*

- *a set of augmented states $\tilde{\mathbf{S}} = \langle s, \mathbf{G}' \rangle$, where $s \in \mathbf{S}$ and $\mathbf{G}' \subseteq \mathbf{G}$;*
- *an augmented start state $\tilde{s_0} = \langle s_0, \mathbf{G} \rangle$;*
- *a set of augmented actions $\tilde{\mathbf{A}} = \langle a, \mathbf{G}'' \rangle$, where $\mathbf{G}''$ is the set of all goals for which $a \in \mathbf{A}$ is an optimal action;*
- *a transition function $\tilde{\mathbf{T}} : \tilde{\mathbf{S}} \times \tilde{\mathbf{A}} \times \tilde{\mathbf{S}} \rightarrow [0, 1]$ that gives the probability $\tilde{T}(\langle s, \mathbf{G}' \rangle, \langle a, \mathbf{G}'' \rangle, \langle s', \mathbf{G}' \cap \mathbf{G}'' \rangle)$ of transitioning from augmented state $\langle s, \mathbf{G}' \rangle$ to augmented state $\langle s', \mathbf{G}' \cap \mathbf{G}'' \rangle$ when augmented action $\langle a, \mathbf{G}'' \rangle$ is executed; this probability equals $T(s, a, s')$ if $|\mathbf{G}' \cap \mathbf{G}''| > 1$ and equals 0 otherwise;*
- *a cost function $\tilde{\mathbf{C}} : \tilde{\mathbf{S}} \times \tilde{\mathbf{A}} \times \tilde{\mathbf{S}} \rightarrow \mathbb{R}^+$ that gives the cost $C(\langle s, \mathbf{G}' \rangle, \langle a, \mathbf{G}'' \rangle, \langle s', \mathbf{G}' \cap \mathbf{G}'' \rangle) = C(s, a, s')$ of executing augmented action $\langle a, \mathbf{G}'' \rangle$ in augmented state $\langle s, \mathbf{G}' \rangle$ and arriving in augmented state $\langle s', \mathbf{G}' \cap \mathbf{G}'' \rangle$; and*
- *the set of augmented goals $\tilde{\mathbf{G}} \subset \tilde{\mathbf{S}} = \{\langle s, \mathbf{G}' \rangle \mid \forall \tilde{T}(\langle s, \mathbf{G}' \rangle, \langle a, \mathbf{G}'' \rangle, \langle s', \mathbf{G}' \cap \mathbf{G}'' \rangle) > 0 \Rightarrow (|\mathbf{G}'| \geq 2 \wedge |\mathbf{G}' \cap \mathbf{G}''| < 2)\}$.*

Figure 3 shows the reachable portion of the augmented MDP that corresponds to the regular MDP shown in Figure 1, where each node corresponds to an augmented state and each hyper-edge corresponds to an augmented action. The elements in the sets for a node/hyper-edge are the possible goals for that augmented state/action.

Finally, the objective here is to find an augmented policy (i.e., a policy that maps augmented states to augmented actions) with the *largest* expected cost. Note that this objective is different from that in regular MDPs, which seek to find the policy with the *smallest* expected cost.

**Definition 2 (All-Goals *wcd*)** *The new* All-Goals *wcd ($wcd_{ag}$) of an S-GRD problem $P$ is defined as:*

$$wcd_{ag}(P) = \max_{\tilde{\pi} \in \tilde{\mathbf{\Pi}}} V_{\tilde{\pi}}(\tilde{s}_0) \tag{3}$$

$$V_{\tilde{\pi}}(\tilde{s}) = \sum_{\tilde{s}' \in \tilde{\mathbf{S}}} \tilde{T}(\tilde{s}, \tilde{\pi}(\tilde{s}), \tilde{s}') \big[ \tilde{C}(\tilde{s}, \tilde{\pi}(\tilde{s}), \tilde{s}') + V_{\tilde{\pi}}(\tilde{s}') \big] \tag{4}$$

*where $\tilde{\mathbf{\Pi}}$ is the set of augmented policies in the augmented MDP and $V_{\tilde{\pi}}(\tilde{s}_0)$ is the expected cost for $s_0$ with augmented policy $\tilde{\pi}$ computed recursively using Equation 4.*

Observe that Equation 3 is analogous to the brute force algorithm to solve an MDP [Mausam and Kolobov, 2012] that performs a policy evaluation over all enumerated policies to return the best policy. As *Value Iteration* (VI) is faster in regular MDPs, we aim to also optimize over the value function space in augmented MDPs. This value function optimization can be done using a Bellman-like equation:

$$V^*(\tilde{s}) = \max_{\tilde{a} \in \tilde{A}} \sum_{\tilde{s}' \in \tilde{\mathbf{S}}} \tilde{T}(\tilde{s}, \tilde{a}, \tilde{s}') \big[ \tilde{C}(\tilde{s}, \tilde{a}, \tilde{s}') + V^*(\tilde{s}') \big] \tag{5}$$

but note that it uses the maximization operator instead of the minimization operator for regular MDPs. This difference will cause an issue if there are *infinite cost cycles*, which are cycles in the graph where the optimal policy is to stay in the cycles and accumulate an infinite cost. Fortunately, our augmented MDP does *not* have infinite cost cycles, and Equation 5 will thus return the correct finite value upon convergence. These properties are formalized in Lemma 1 and Theorem 1, where proof sketches are provided.

**Lemma 1** *The augmented MDP does not have infinite cost cycles.*

**Proof Sketch:** We prove that an infinite cost cycle between two augmented states cannot exist by contradiction. Assume that such a cycle exists. Thus, $\tilde{T}(\tilde{s}, \tilde{a}, \tilde{s}') + \tilde{T}(\tilde{s}, \tilde{a}, \tilde{s}) = 1$ and $\tilde{T}(\tilde{s}', \tilde{a}, \tilde{s}') + \tilde{T}(\tilde{s}', \tilde{a}, \tilde{s}) = 1$ for some augmented states $\tilde{s} = \langle s, \mathbf{G}_1 \rangle$ and $\tilde{s}' = \langle s', \mathbf{G}_2 \rangle$, and augmented actions $\tilde{a} = \langle a, \mathbf{G}_3 \rangle$ and $\tilde{a}' = \langle a', \mathbf{G}_4 \rangle$. Since they form an infinite cost cycle, then $\mathbf{G}_1 = \mathbf{G}_2$. Let $g \in \mathbf{G}_1$ be one of the possible goals. Then, there must exist the actions $\pi^*_g(s) = a$ that transitions from $s$ to $s'$, and $\pi^*_g(s') = a'$ that transitions from $s'$ to $s$ in the optimal policy $\pi^*_g$ for goal $g$, which forms an infinite cost cycle in the optimal policy. This is not possible for optimal policies of SSP-MDPs [Mausam and Kolobov, 2012], which contradicts our assumption. ■

**Theorem 1** *$wcd_{ag} = V^*_{\tilde{\pi}}(\tilde{s}_0)$, which can be computed recursively via Equation 5.*

**Proof Sketch:** Equation 5 is, like the original Bellman equation [Bellman, 1957], a contracting operator. As such, it will eventually converge to the true optimal value. The only exception is if there are infinite cost cycles, which will cause the value of some states to converge to infinity. However, since there are no infinite cost cycles in our augmented MDP (Lemma 1), they will converge to finite values. ■

**Theorem 2** *$wcd_{ag} \geq wcd$ as defined by Wayllace* et al. *[2016].*

**Proof Sketch:** One can view the augmented MDP for $wcd_{ag}$ as a union of all the augmented MDPs for regular $wcd$. For example, the subgraph formed by the actions in bold and their predecessor and successor states of Figure 1 is a union of all three subgraphs in Figure 2. Since both definitions of $wcd_{ag}$ and $wcd$ maximizes the expected cost in their respective subgraphs, it must be the case that $wcd_{ag} \geq wcd$ since the subgraph of the former is a union of all the subgraphs for the latter. ■

**Corollary 1** *If there are only two possible goals in the S-GRD, then $wcd_{ag} = wcd$.*

## 4 Expected-Case Distinctiveness (*ecd*)

An implicit assumption made by the *worst-case distinctiveness* ($wcd$) metric is that there is no prior information on which is the true agent's goal. While this assumption is reasonable in many problems, it may be the case that some information is available. For example, in human-computer interaction applications, user profiles may be used to assign different weights to each goal, where the weights correspond to the prior probabilities of an agent choosing its goal.

Further, it may often be the case where the $wcd$ cannot be reduced (i.e., the longest non-distinctive path cannot be shortened). However, other shorter non-distinctive paths can be shortened. Thus, intuitively, one should prefer the solution that shortens the shorter non-distinctive paths. In such a scenario, the $wcd$ metric fails to distinguish between these solutions as the $wcd$ remains the same in both cases. This situation is further exacerbated when the longest path are to goals with low weights!

Therefore, in response to these two observations, we propose a new metric, called the *expected-case distinctiveness* ($ecd$), for S-GRDs that weighs the length of a path to a goal by the probability of an agent choosing that goal and takes the sum of all the weighted path lengths.

**Definition 3 (expected-case distinctiveness)**

$$ecd(P) = E(\tilde{s}_0) \tag{6}$$

$$E(\tilde{s}) = \sum_{\tilde{a},\tilde{s}'} P(\tilde{a})\tilde{T}(\tilde{s},\tilde{a},\tilde{s}')\big[\tilde{C}(\tilde{s},\tilde{a},\tilde{s}') + E(\tilde{s}')\big] \tag{7}$$

$$P(\tilde{a} = \langle a, \mathbf{G}' \rangle) = \frac{1}{Z} \sum_{g_i \in \mathbf{G}'} w_i \tag{8}$$

*where $w_i > 0$ is the probability of the agent choosing goal $g_i$ and $Z$ is the normalization constant such that $\sum_{\tilde{a} \in \tilde{\mathbf{A}}} P(\tilde{a}) = 1$.*

Intuitively, Equation 8 associates a probability $P(\tilde{a})$ to each augmented action $\tilde{a}$ based on the number of goals for which that action is an optimal action as well as the probabilities of those goals being the true goal. Then, Equation 7 recursively defines the expected cost of each augmented state $\tilde{s}$ based on the probability of each augmented action and the expected cost of each successor state. Finally, the $ecd$ of the problem is the expected cost of the augmented start state.

This definition also has the added benefit that the $ecd$ computation is straightforward and efficient. The runtime to compute it is similar to the runtime of VI as it requires multiple iterations of a Bellman-like equation until the expected costs of all states have converged.

**Theorem 3** $ecd(P) \leq wcd_{ag}(P)$

**Proof Sketch:** $wcd_{ag}$ is the expected cost of the *worst* policy to reach an augmented goal state in the augmented MDP, while $ecd$ is the expected cost over *all* policies. Therefore, $ecd(P) \leq wcd_{ag}(P)$ ■

**Corollary 2** *If there is exactly one policy in the augmented MDP, then $ecd(P) = wcd_{ag}(P)$.*

## 5 Algorithms

To compute the $wcd_{ag}$ and $ecd$ of each problem, one can use a VI-like algorithm that runs iterations of the Bellman-like update of Equations 5 and 7, respectively. Additionally, we make the observation that the augmented state space of the augmented MDP can often be segmented into *strongly connected components* (SCCs) – each SCC contains the augmented states with the same set of possible goals, and the set of possible goals is non-increasing. Therefore, we also propose a TVI-like algorithm that uses Tarjan's algorithm [Tarjan, 1972] to segment the augmented state space into SCCs first before running VI on each SCC in reverse topological order. This should significantly speed up the solving time if there are large numbers of SCCs, but may have the opposite effect if there are few SCCs due to the overhead incurred by Tarjan's algorithm.

Optimally computing $wcd_{ag}$ is P-complete in the number of reachable augmented states as it is equivalent to optimally solving an augmented MDP, which is P-complete [Mausam and Kolobov, 2012]. Optimally computing $ecd$ is easier as it is equivalent to evaluating a single policy in the augmented MDP. It is thus in P, but not P-hard.

Similar to Wayllace *et al.* [2016], to reduce the $wcd_{ag}$ or $ecd$ of a problem, we enumerate through all possible combinations of actions to remove, compute the resulting $wcd_{ag}$ or $ecd$, and store the best solution. Additionally, we can use Theorem 4 to make useful inferences.

**Theorem 4** *If there are no combination of $k$ actions in an S-GRD that can reduce its $ecd$, then there are also no such combinations that can reduce its $wcd_{ag}$.*

| Domain | Instances | $wcd_{ag}$ Reduction | Runtime (s) ENUM | VI | TVI | $wcd$ Reduction | Runtime (s) R-W(O) | $ecd$ Reduction | Runtime (s) VI | TVI |
|---|---|---|---|---|---|---|---|---|---|---|
| ROOM | 8-8-3 | 6.2 → 6.2 | 1 | 1 | 1 | 6.0 → 6.0 | 1 | 6.2 → 6.2 | 1 | 1 |
| | 32-32-2 | 86.6 → 86.6 | 184 | 101 | 85 | 86.6 → 86.6 | 18,526 | 86.6 → 86.6 | 98 | 86 |
| | 32-32-3 | 86.6 → 86.6 | 324 | 164 | 140 | 86.6 → 86.6 | 38,367 | 86.6 → 86.6 | 161 | 156 |
| | 44-44-5 | 73.8 → 73.8 | timeout | 946 | 878 | 73.8 → 73.8 | 16,959 | 73.8 → 73.8 | 923 | 928 |
| GRID-NA-VIGATION | 5-5-3 | 4.4 → 3.3 | 1 | 1 | 1 | 4.4 → 3.3 | 1 | 3.7 → 2.9 | 1 | 1 |
| | 4-12-3 | 10 → 10 | timeout | 3 | 3 | 10 → 10 | 789 | 10 → 8.9 | 3 | 3 |
| | 4-12-3 | 5.6 → 5.6 | 13 | 3 | 3 | 5.6 → 5.6 | 4 | 4.9 → 4.4 | 3 | 3 |
| | 4-12-3 | 6.7 → 5.6 | 18 | 3 | 3 | 6.7 → 5.6 | 4 | 6.7 → 5.1 | 3 | 3 |
| | 4-12-6 | 13.3 → 12.2 | timeout | 3 | 3 | 13.3 → 12.2 | 25 | 9.4 → 9.1 | 4 | 4 |
| BLOCKS-WORLD | 5-3 | 1.3 → 1.3 | 1 | 2 | 1 | 1.3 → 1.3 | 1 | 1.3 → 1.3 | 1 | 1 |
| | 6-5 | 4.9 → 4.9 | 582 | 575 | 626 | — | timeout | 3.7 → 3.7 | 644 | 659 |
| | 6-3 | 3.4 → 3.4 | 374 | 351 | 353 | — | timeout | 3.1 → 3.1 | 379 | 378 |
| COLORED BLOCKS-WORLD | 3-2-2 | 2.8 → 2.8 | 1 | 1 | 1 | 2.8 → 2.8 | 1 | 2.8 → 2.8 | 1 | 1 |
| | 5-2-3 | 4.1 → 4.1 | 20 | 17 | 17 | 4.1 → 4.1 | 19 | 4.1 → 4.1 | 16 | 17 |
| | 5-3-3 | 4.9 → 3.5 | timeout | 16 | 17 | — | timeout | 4.9 → 3.5 | 18 | 18 |
| | 6-2-3 | 14.6 → 14.6 | timeout | 390 | 390 | — | timeout | 10.2 → 8.3 | 406 | 414 |
| | 6-3-3 | 6.3 → 5.3 | timeout | 408 | 420 | — | timeout | 5.9 → 5.3 | 410 | 420 |
| BOXWORLD | 2-1-1-5-3 | 4.1 → 4.1 | 29 | 29 | 32 | 4.1 → 4.1 | 30 | 4.1 → 4.1 | 33 | 32 |
| | 2-1-1-6-3 | 4.3 → 4.3 | 10 | 10 | 10 | 4.1 → 4.1 | 10 | 4.3 → 4.3 | 12 | 9 |
| | 3-1-1-6-3 | 5.2 → 5.2 | 413 | 414 | 402 | 4.9 → 4.9 | 414 | 5.2 → 5.2 | 398 | 478 |
| | 3-1-1-6-5 | 8.1 → 8.1 | 653 | 642 | 404 | 7.3 → 7.3 | 593 | 7.8 → 7.8 | 400 | 593 |
| | 3-2-1-6-3 | 5.2 → 5.2 | 50,204 | 49,028 | 48,955 | 5.2 → 5.2 | 49,082 | 5.2 → 5.2 | 49,003 | 48,861 |

Table 1: Experimental Results for $k = 1$

**Proof Sketch:** Since the $ecd$ is the expected cost over *all* policies in the augmented MDP, if it cannot be reduced, then it must mean that every policy in the augmented MDP for the original problem cannot be modified; modifying them would affect the optimal cost to one of the possible goals in the regular MDP. And since all policies cannot be modified, the $wcd_{ag}$, which is the expected cost of the *worst* policy, will also remain unchanged. ■

## 6 Experimental Results

The first domain is called ROOM, which is used in the Non-Deterministic Track of the 2006 ICAPS International Planning Competition.[3] It is a grid world where the actions as well as the transition probabilities are defined individually for each state. The actions allow the agent to move in the four cardinal directions but after taking one action the agent has some probability to end up in other adjacent cells including a diagonal one. Each instance of this domain is defined by the $x$- and $y$-dimensions of the room and the number of possible goals.

The second domain, GRID-NAVIGATION, is also a grid world where the transitions are defined equally for all the states; the agent has a 90% probability to move to a cell indicated by the deterministic outcome of the action and a 10% probability to stay in the same cell. The instances are defined in the same way as in the ROOM domain.

The remaining domains were used in the Probabilistic Track of 2004 ICAPS International Planning Competition:[4] BLOCKSWORLD is the traditional domain with a 25% probability of slippage each time a block is picked up or put down; the goal state defines the last position of every block. Each instance is defined by the number of blocks and the number of possible goals. COLORED-BLOCKSWORLD is a modification of BLOCKSWORLD where each block has a color and the goal is specified in terms of colors. Thus, more than one state can represent the same goal. Instances in this domain are defined by the number of blocks, number of colors, and number of goals. BOXWORLD is a modified LOGISTICS domain where the only action that introduces noise is "drive-truck" and there is a 20% probability that the truck ends up in one of three wrong cities. Each instance in this domain is defined by the number of boxes, trucks, airplanes, cities and goals. Tables 1 and 2 tabulate the results when the number of actions that can be blocked ($k$) is 1 and 2, respectively. The experiments were conducted on a 3.1GHz quad-core machine with 6GB of RAM and we imposed a timeout of 2 days.

We compared three algorithms to compute the $wcd_{ag}$: ENUM, which explicitly enumerates through all policies using Equation 4 and both VI and TVI as described in Section 5. We also computed the $wcd$ using the REDUCE-WCD algorithm with optimization [Wayllace *et al.*, 2016] (labeled as R-W(O)). To compute the $ecd$, we only compared VI and TVI because explicit enumeration of policies is not needed as it is equivalent to evaluating a single policy. Finally, we

[3] http://idm-lab.org/wiki/icaps/ipc2006/probabilistic/

[4] http://www.cs.rutgers.edu/∼mlittman/topics/ipc04-pt/

| Domain | Instances | $wcd_{ag}$ Reduction | Runtime (s) ENUM | Runtime (s) VI | Runtime (s) TVI | $wcd$ Reduction | Runtime (s) R-W(O) | $ecd$ Reduction | Runtime (s) VI | Runtime (s) TVI |
|---|---|---|---|---|---|---|---|---|---|---|
| ROOM | 8-8-3 | 6.2 → 6.2 | 7 | 8 | 8 | 6.0 → 6.0 | 6.4 | 6.2 → 6.2 | 8 | 8 |
| | 32-32-2 | 86.6 → 86.6 | 234,126 | 50,609 | 40,303 | — | timeout | 86.6 → 86.6 | 47,690 | 42,057 |
| | 32-32-3 | 86.6 → 86.6 | timeout | timeout | 78,245 | — | timeout | 86.6 → 86.6 | 85,417 | 84,134 |
| | 44-44-5 | — | timeout | timeout | timeout | — | timeout | — | timeout | timeout |
| GRID-NA-VIGATION | 5-5-3 | 4.4 → 2.2 | 2 | 1 | 1 | 4.4 → 2.2 | 1 | 3.7 → 2.2 | 1 | 1 |
| | 4-12-3 | 10 → 8.9 | timeout | 14 | 5 | 10 → 8.9 | 13,739 | 10 → 7.8 | 12 | 5 |
| | 4-12-3 | 5.6 → 4.4 | 246 | 5 | 5 | 5.6 → 4.4 | 8 | 4.9 → 3.9 | 6 | 6 |
| | 4-12-3 | 6.7 → 4.4 | 418 | 5 | 5 | 6.7 → 4.4 | 32 | 6.7 → 4.4 | 5 | 6 |
| | 4-12-6 | 13.3 → 12.2 | timeout | 7 | 7 | 13.3 → 12.2 | 8,095 | 9.4 → 8.4 | 8 | 8 |
| BLOCKS-WORLD | 5-3 | 1.3 → 1.3 | 2 | 2 | 1 | 1.3 → 1.3 | 2 | 1.3 → 1.3 | 1 | 1 |
| | 6-5 | 4.9 → 4.9 | 10,883 | 10,836 | 9,832 | — | timeout | 3.7 → 3.7 | 9,586 | 10,313 |
| | 6-3 | 3.4 → 3.4 | 6,776 | 6,721 | 5,027 | — | timeout | 3.1 → 3.1 | 4,464 | 4,368 |
| COLORED BLOCKS-WORLD | 3-2-2 | 2.8 → 2.8 | 1 | 1 | 1 | 2.8 → 2.8 | 1 | 2.8 → 2.8 | 1 | 1 |
| | 5-2-3 | 4.1 → 2.8 | timeout | 448 | 429 | — | timeout | 4.1 → 2.8 | 453 | 426 |
| | 5-3-3 | 4.9 → 3.5 | timeout | 380 | 385 | — | timeout | 4.9 → 3.5 | 383 | 393 |
| | 6-2-3 | 14.6 → 14.6 | timeout | 17,539 | 16,553 | — | timeout | 10.2 → 7.9 | 16,665 | 17,378 |
| | 6-3-3 | 6.3 → 5.3 | timeout | 16,778 | 16,132 | — | timeout | 5.9 → 5.3 | 15,667 | 16,888 |
| BOXWORLD | 2-1-1-5-3 | 4.1 → 4.1 | 43 | 37 | 128 | 4.1 → 4.1 | 56 | 4.1 → 4.1 | 133 | 132 |
| | 2-1-1-6-3 | 4.3 → 4.3 | 51 | 51 | 14 | 4.1 → 4.1 | 17 | 4.3 → 4.3 | 14 | 14 |
| | 3-1-1-6-3 | 5.2 → 5.2 | 1,011 | 976 | 482 | 4.9 → 4.9 | 519 | 5.2 → 5.2 | 667 | 584 |
| | 3-1-1-6-5 | 8.1 → 8.1 | 12,003 | 1,164 | 735 | 7.3 → 7.3 | 823 | 7.8 → 7.8 | 738 | 736 |
| | 3-2-1-6-3 | 5.2 → 5.2 | 57,807 | 53,001 | 55,788 | 5.2 → 5.2 | 75,546 | 5.2 → 5.2 | 59,778 | 59,715 |

Table 2: Experimental Results for $k = 2$

assumed that all goals have equal weights when computing the $ecd$ in order to have a fair comparison with $wcd_{ag}$.

We make the following observations:

- As expected, when $k$ increases, the runtimes increased but the $wcd_{ag}$ and $ecd$ were reduced in more instances. In the ROOM, BLOCKSWORLD, and BOXWORLD domains, the $wcd_{ag}$ and $ecd$ remained unchanged. The reason is that, in all three domains, each goal has only very few (either 1 or 2) optimal policies and removing any action that affects these policies will increase the expected cost to that goal, which is prohibited by the definition of S-GRDs.
- In general, consistent with the trends in regular MDPs with multiple SCCs [Dai *et al.*, 2011], TVI is often faster than VI, which is faster than ENUM. The reason is that TVI does not need to perform Bellman updates on states outside the SCC being considered, thereby reducing the runtime of each iteration.
- Computing $wcd_{ag}$ is up to three orders of magnitude faster than computing $wcd$. The reason is that the $wcd$ computation needs to iterate through all possible pairs of goals and all possible enumerated policies, each of which must be solved via VI, while the $wcd_{ag}$ computation solves the whole problem in a single shot via VI or TVI. The $wcd_{ag}$ and $wcd$ values are the same in all instances except for four instances, indicating that the $wcd$ metric is intuitively correct except for those four instances.
- There are several instances where the $ecd$ is reduced but the $wcd_{ag}$ remains unchanged. The reason is that it is possible to shorten *some* of the paths to the goals before the agent is forced to reveal its action, but not the *longest* path. Therefore, one can use $ecd$ as a tie-breaking metric between two solutions with identical $wcd_{ag}$ values, which is important given that $wcd_{ag}$ usually remains unchanged for small values of $k$.

## 7 Conclusions and Future Work

The *Stochastic Goal Recognition Design* (S-GRD) model is an important variant of the GRD model. Unfortunately, the original $wcd$ metric proposed has a formal definition that is inconsistent with its intuitive definition. Further, the $wcd$ definitions for all the GRD variants make the implicit assumption that all goals have equal likelihood of being the true goal, which is unrealistic in many practical applications. Therefore, in this paper, we proposed a new $wcd$ metric, called *all-goals* $wcd$ ($wcd_{ag}$), that remedies the inconsistency above as well as a new *expected-case distinctiveness* ($ecd$) metric that weighs the possible goals based on their likelihood of being the true goal. We further show the complexity of computing these metrics and introduced efficient algorithms to compute them. Experimental results show that it is possible to reduce the $ecd$ in several cases where the $wcd_{ag}$ remained unchanged. This result highlights the need for more metrics that can be better used by practitioners to trade off the different possible solutions.

Future work includes the generalization of the $ecd$ metric to the other GRD variants as well as the investigation of heuristic search techniques to compute the $wcd_{ag}$ and $ecd$ values. We suspect that such techniques may be useful in cases one can prune significant portions of the search space due to the differences in the weights of the goals.

## References


[Bellman, 1957] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[Dai *et al.*, 2011] Peng Dai, Mausam, Daniel S Weld, and Judy Goldsmith. Topological value iteration algorithms. *Journal of Artificial Intelligence Research*, 42:181–209, 2011.

[Geffner and Bonet, 2013] Hector Geffner and Blai Bonet. *A Concise Introduction to Models and Methods for Automated Planning*. Morgan & Claypool Publishers, 2013.

[Jarvis *et al.*, 2005] Peter Jarvis, Teresa Lunt, and Karen Myers. Identifying terrorist activity with AI plan recognition technology. *AI Magazine*, 26(3):73–81, 2005.

[Johnson, 2010] W. Lewis Johnson. Serious use of a serious game for language learning. *International Journal of Artificial Intelligence in Education*, 20(2):175–195, 2010.

[Kautz, 1987] Henry A Kautz. *A Formal Theory of Plan Recognition*. PhD thesis, Bell Laboratories, 1987.

[Kelley *et al.*, 2012] Richard Kelley, Liesl Wigand, Brian Hamilton, Katie Browne, Monica Nicolescu, and Mircea Nicolescu. Deep networks for predicting human intent with respect to objects. In *Proceedings of the International Conference on Human-Robot Interaction (HRI)*, pages 171–172, 2012.

[Keren *et al.*, 2014] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 154–162, 2014.

[Keren *et al.*, 2015] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design for non-optimal agents. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 3298–3304, 2015.

[Keren *et al.*, 2016a] Sarah Keren, Avigdor Gal, and Erez Karpas. Goal recognition design with non-observable actions. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 3152–3158, 2016.

[Keren *et al.*, 2016b] Sarah Keren, Avigdor Gal, and Erez Karpas. Privacy preserving plans in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3170–3176, 2016.

[Keren *et al.*, 2017] Sarah Keren, Luis Pineda, Avigdor Gal, Erez Karpas, and Shlomo Zilberstein. Redesigning stochastic environments for maximized utility. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[Lee *et al.*, 2012] Seung Lee, Bradford Mott, and James Lester. Real-time narrative-centered tutorial planning for story-based learning. In *Proceedings of the International Conference on Intelligent Tutoring Systems (ITS)*, pages 476–481, 2012.

[Mausam and Kolobov, 2012] Mausam and Andrey Kolobov. *Planning with Markov Decision Processes: An AI Perspective*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

[McQuiggan *et al.*, 2008] Scott McQuiggan, Jonathan Rowe, Sunyoung Lee, and James Lester. Story-based learning: The impact of narrative on learning experiences and outcomes. In *Proceedings of the International Conference on Intelligent Tutoring Systems (ITS)*, pages 530–539, 2008.

[Min *et al.*, 2014] Wookhee Min, Eunyoung Ha, Jonathan Rowe, Bradford Mott, and James Lester. Deep learning-based goal recognition in open-ended digital games. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, pages 37–43, 2014.

[Oh *et al.*, 2010] Jean Oh, Felipe Meneguzzi, Katia Sycara, and Timothy Norman. ANTIPA: An agent architecture for intelligent information assistance. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pages 1055–1056, 2010.

[Oh *et al.*, 2011a] Jean Oh, Felipe Meneguzzi, Katia Sycara, and Timothy Norman. An agent architecture for prognostic reasoning assistance. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2513–2518, 2011.

[Oh *et al.*, 2011b] Jean Oh, Felipe Meneguzzi, Katia Sycara, and Timothy Norman. Probabilistic plan recognition for intelligent information agents: Towards proactive software assistant agents. In *Proceedings of the International Conference on Agents and Artificial Intelligence (ICAART)*, pages 281–287, 2011.

[Schmidt *et al.*, 1978] Charles Schmidt, N. Sridharan, and John Goodson. The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence*, 11(1–2):45–83, 1978.

[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)*, pages 3181–3187, 2016.

[Sukthankar *et al.*, 2014] Gita Sukthankar, Christopher Geib, Hung Hai Bui, David Pynadath, and Robert P Goldman. *Plan, activity, and intent recognition: Theory and practice*. Newnes, 2014.

[Tarjan, 1972] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[Tavakkoli *et al.*, 2007] Alireza Tavakkoli, Richard Kelley, Christopher King, Mircea Nicolescu, Monica Nicolescu, and George Bebis. A vision-based architecture for intent recognition. In *Proceedings of the International Symposium on Advances in Visual Computing*, pages 173–182, 2007.

[Wayllace *et al.*, 2016] Christabel Wayllace, Ping Hou, William Yeoh, and Tran Cao Son. Goal recognition design with stochastic agent action outcomes. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3279–3285, 2016.