Goal Recognition Design: Tutorial

Sarah Keren & William Yeoh

ICAPS 2019

- Present the *GRD* framework and its purpose
- Motivate *GRD* using simple examples and real-world applications
- Show the relationship between *GRD* and planning
- Specify open challenges

Did you get a chance to play the playGRounD game?

A little bit about us

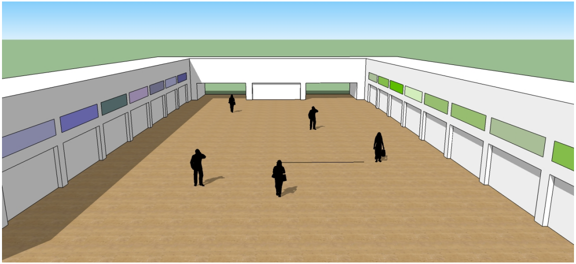
Sarah

- Now a post-doc at Harvard's school of engineering and applied sciences, working on various variations of Utility Maximizing Design
- GRD was the topic of my PhD thesis at the Technion
- email: sarah.e.keren@gmail.com or skeren@seas.harvard.edu
- website: https://sarahkeren.wixsite.com/sarahkeren-academics

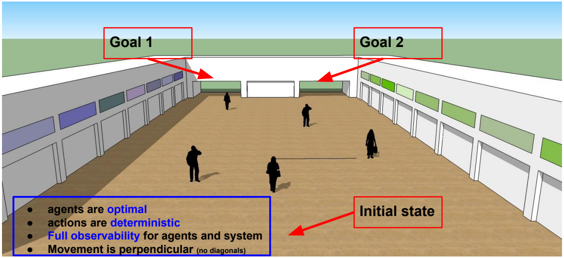
William

- An assistant professor at Washington University in St. Louis
- Primary research area is in multiagent systems, but got excited about GRD after learning about it from Sarah :)
- email: wyeoh@wustl.edu
- website: https://sites.wustl.edu/wyeoh/

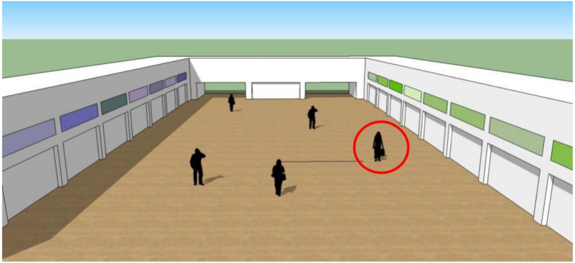
A goal recognition setting

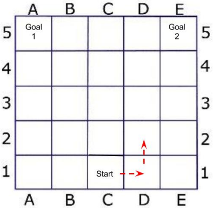


A goal recognition setting

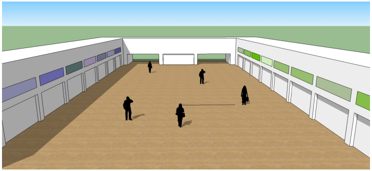


A goal recognition setting



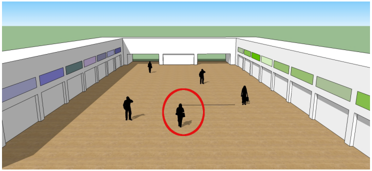


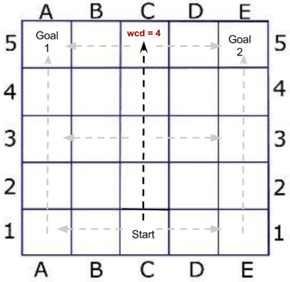
Offline design as a way to facilitate **online** goal recognition



Worst case distinctiveness (wcd) as a measure of model quality

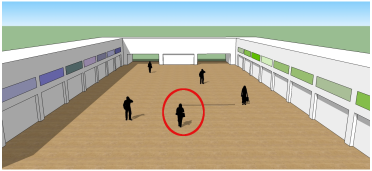
Offline design as a way to facilitate **online** goal recognition

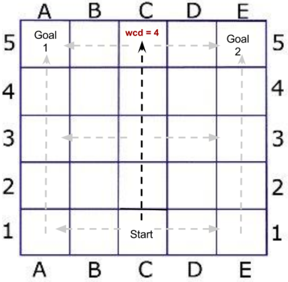


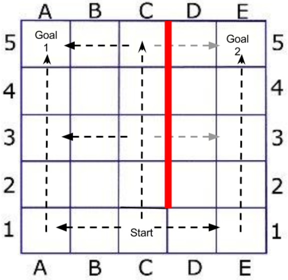


Worst case distinctiveness (wcd) as a measure of model quality

Offline design as a way to facilitate online goal recognition

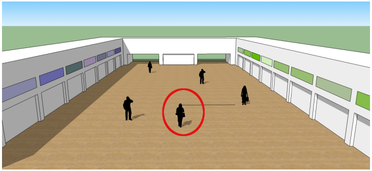


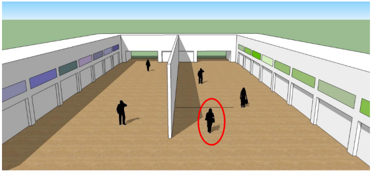


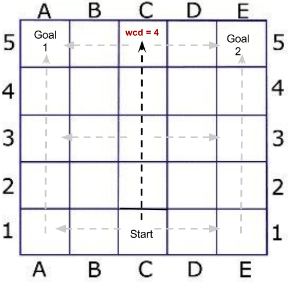


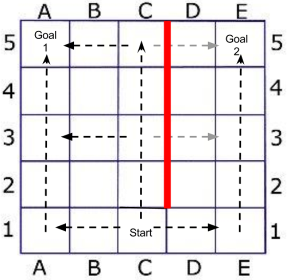
Worst case distinctiveness (wcd) as a measure of model quality

Offline design as a way to facilitate **online** goal recognition





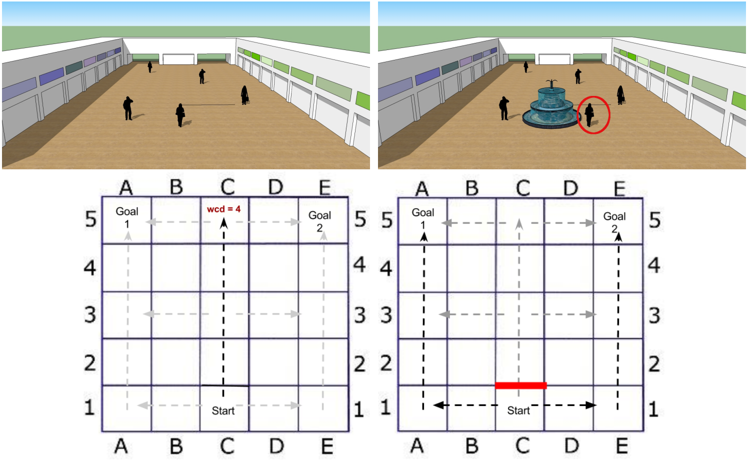




Worst case distinctiveness (wcd) as a measure of model quality

Goal Recognition Design (GRD)

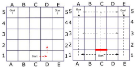
Offline design as a way to facilitate **online** goal recognition



Worst case distinctiveness (wcd) as a measure of model quality

Goal Recognition vs. Goal Recognition Design

- Goal Recognition — **Online**
  - Recognize : Given an observation sequence - what are the possible goals?
- Goal Recognition Design — **Offline**
  Design for early recognition
  - Evaluate : Worst Case Distinctiveness(*WCD*) - maximal number of steps an agent can take before his goal is revealed?
  - Optimize : How can we modify the model to reduce *WCD* without increasing agent cost?

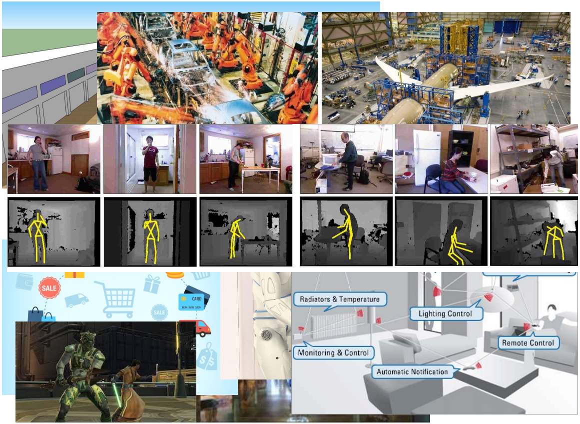


Goal recognition is to planning what inverse reinforcement learning is to reinforcement learning — GRD aims to facilitate the inverse-planning problem

Goal Recognition Design
Applications

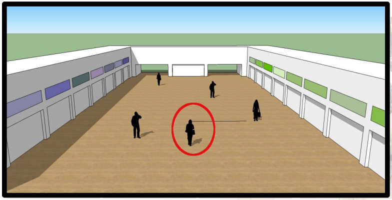
Applies to any goal recognition setting that can be controlled.

Extremely relevant to our 'big data' world



Applies to any goal recognition setting that can be controlled.

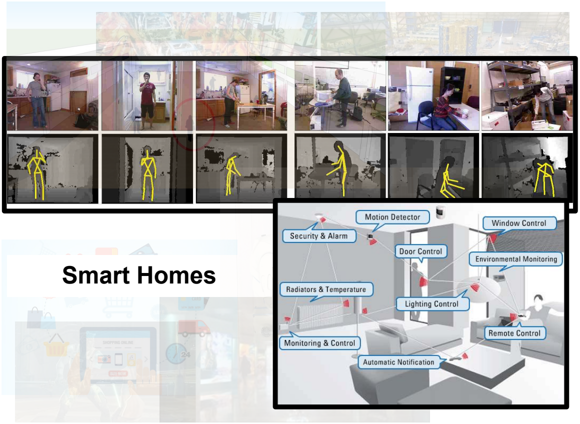
Extremely relevant to our 'big data' world



Intrusion detection

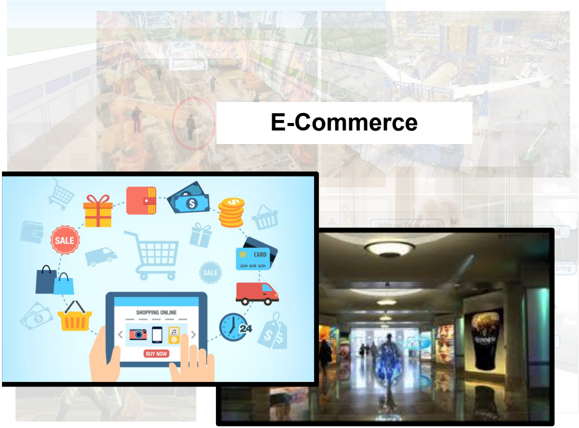
Applies to any goal recognition setting that can be controlled.

Extremely relevant to our 'big data' world



Applies to any goal recognition setting that can be controlled.

Extremely relevant to our 'big data' world



Applies to any goal recognition setting that can be controlled.

Extremely relevant to our 'big data' world

Goal Recognition Design (GRD)
Tutorial Outline

- Elements of a GRD problem (Sarah)
- GRD in deterministic environments (Sarah)
- GRD in stochastic environments (William)
- GRD for partially informed agents (Sarah)
- Conclusions

Elements of a GRD Problem

- The **Goal recognition setting** analyzed
  - Environment
  - Acting agent (actor)
  - Recognition system (recognizer, observer)
- A **Design model**
  - The possible ways to change the environment

Environment

- The setting in which agents act (a.k.a as the *domain theory*)
- Can be described as a state space
- Typically, the description includes
  - a set of features to describe a state
  - a set of possible initial states
  - a set of actions that can be performed at each state:
    - deterministic / non-deterministic / stochastic actions
    - temporal actions
  - a set of possible goals (states or conditions to be met)

An environment induces a set of possible behaviors

Actor

- The model of the *actor* (acting agent) specifies the **assumptions made** w.r.t. how an agent with a specific goal chooses to behave in a given environment
- We are assuming agents enter the environment and follow a policy / plan to achieve some goal
- **Note**: recognition in a multi-agent setting is an interesting extension but beyond scope for today!

Actor

In *GRD*, we need to account for the **set** of plans an actor may follow to achieve each of the possible goals

In particular, we need to answer the following questions:

- **How does the actor make decisions?**
- **What does the actor know and how does it perceive its surrounding ?**
- **What is the actor's relationship to the recognizer?**
- **What is the best formalism to represent the actor?**

We are representing the actor from the recognizer's point of view

Actor

- **How does the actor make decisions?**
  - For example: actors are optimal or sub-optimal
- **What does the actor know and how does it perceive its surrounding ?**
  - For example: when partially informed, we need to account for the actor's sensor model.
  - Typically, a **belief state** is used to represent the states an agent deems as possible / a probability distribution over states.

Actor

- **What is the actor's relationship to the recognizer?**
  - Agnostic - the actor is agnostic to / unaware of the recognition process
  - Adversarial - the actor wants to deceive the recognizer (given its own constraints)
  - Intended - the actor wants to implicitly communicate its goal / plan to the recognizer

Strongly related to the topic of **explainable/ privacy preserving planning** - which assumes the role of an agent that chooses to behave in a way that **reveals / obfuscates** its objective

Actor

- **What is the best formalism to represent the actor?**
  - There are many possible ways to represent the actor.
  - Two commonly used representations are plan libraries and domain theories.
  - Today we are going to focus on **Domain theory (planning)** to represent the actor (Ramirez and Geffner 2010).

Recognizer (Recognition System)

- The actor's model specified how the recognizer expects the actor to behave w.r.t each goal
- For the recognizer, we need to specify the **Observability** - How does the recognizer perceive the actor's behavior? What is the recognizer's sensor model
- The recognizer's sensor model is a mapping from executions/ plans / sequences to observation sequences
- The observation sequence is the entity that is analyzed

Recognizer's Objective

Three types of Recognition

- **Plan recognition** - identify the sequence of actions the actor follows to achieve it's goal
- **Goal recognition** - identify the end conditions the actor wishes to a achieve
- **Activity recognition** - identify a specific action that is being performed by the actor

Today we will focus on goal recognition and the way to facilitate it via design

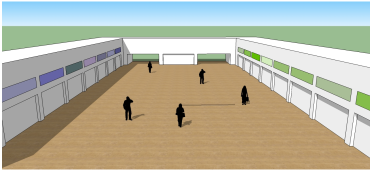
Design Model

It may be possible to affect the actor's behavior

- Online
  - Provoking the actor to behave in a specific way by setting the value of environment feature (Bisson, Kabanza, Benaskeur & Irandoust 2011)
  - Direct communication- Asking the actor questions about its plans / goals ((Mirsky, Stern, Gal, Kalech 2018)
- Offline - Goal Recognition Design
  - Facilitating online goal recognition via design
  - **Our focus today**

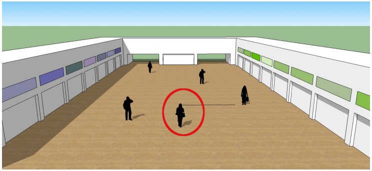
Goal Recognition Design (GRD) in Deterministic Environments

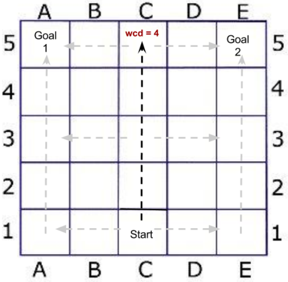
Offline design as a way to facilitate **online** goal recognition



Worst case distinctiveness (wcd) as a measure of model quality

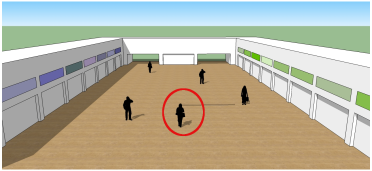
Offline design as a way to facilitate **online** goal recognition

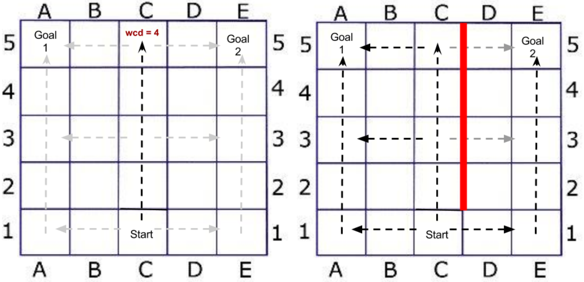




Worst case distinctiveness (wcd) as a measure of model quality

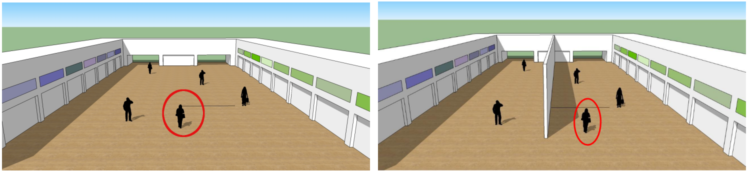
Offline design as a way to facilitate online goal recognition

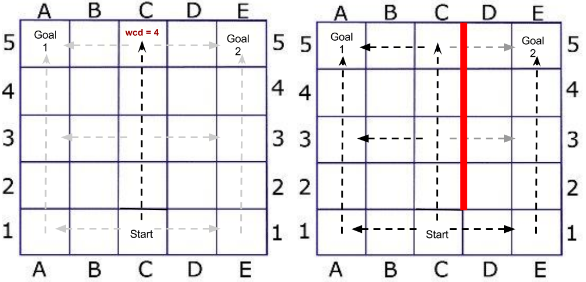




Worst case distinctiveness (wcd) as a measure of model quality

Offline design as a way to facilitate online goal recognition





Worst case distinctiveness (wcd) as a measure of model quality

- Background: automated planning
- Problem definition
- Computing *WCD* using compilations to classical planning
- Minimizing *WCD* using strong stubborn sets for safe pruning

Background: Domain Independent Planning

- A domain independent *classical* planning problem contains:
  - Initial world state
  - Desired goal condition
  - Set of deterministic actions
- Compact representation of the state space using STRIPS
  - A set P of boolean propositions are used to represent the world state. The goal condition is a subset of p.
  - Each action is a triple $a = \langle \text{pre}(a) \subseteq p, \text{add}(a) \subseteq p, \text{del}(a) \subseteq p \rangle$, specifying the conditions, add effects and delete effects of each transition.
- A solution is a sequence of actions:
  - Transforms the initial world state into a goal state
  - It is optimal if it minimizes sum of action costs
- Other models for planning account for various forms of uncertainty: (stochastic actions, conformant, contingent, partially observable Markov decision processes (POMDP), etc)

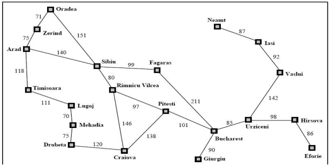
Background: Domain Independent Planning
Representation: STRIPS

- A STRIPS planning problem with action costs is a 5-tuple

$$\Pi = \langle P, s_0, G, A, cost \rangle$$

  where
  - P is a set of boolean propositions
  - $s_0 \subseteq P$ is the initial state
  - $G \subseteq P$ is the goal
  - A is a set of actions
  - Each action is a triple $a = \langle \text{pre}(a), \text{add}(a), \text{del}(a) \rangle$
  - $cost : A \rightarrow \mathbb{R}^{0+}$ assigns a cost to each action
- Applying action sequence $\pi = \langle a_0, a_1, \ldots, a_n \rangle$ at state s leads to $s\{\pi\}$
- The cost of action sequence π is $\sum_{i=0}^{n} cost(a_i)$

Background: Domain Independent Planning



- A *classical* planning problem:
  - Initial world state
  - Desired goal condition
  - Set of (deterministic) actions
- Seeking a minimal plan to goal
- Any planning problem **implicitly** defines a directed graph
- In theory, Dijkstra's algorithm can solve the planning problem

Background: Domain Independent Planning

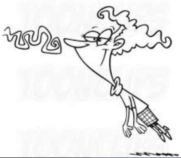
- A *classical* planning problem:
  - Initial world state
  - Desired goal condition
  - Set of (deterministic) actions
- Seeking a minimal plan to goal
- Any planning problem **implicitly** defines a directed graph
- In theory, Dijkstra's algorithm can solve the planning problem
- Actual graphs are too big to be solved exhaustively
- We use different strategies to efficiently find solutions
- Other models of planning account for various forms of uncertainty: (stochastic actions (MDP), conformant, contingent, partially observable MDP, etc)
- Domain Independent!

- **Pruning:**
  - Ignore part of the search tree
  - Safe pruning guarantees at least one desired solution is not pruned

- **Heuristic function: estimates cost to goal**
  - Admissible - underestimate the cost to goal
  - Automatic extraction from problem descriptions
  - Used with heuristic search algorithms (e.g. A^*)

We want informative and easy-to-compute admissible heuristics

Many **domain-independent** solvers and heuristics developed in the past 20 years



Goal Recognition Design in Deterministic Environments

Goal Recognition Design

- An initial goal recognition model $R \in \mathcal{R}$
  - Environment $\epsilon = \langle s_0, \mathbb{G}, A, cost \rangle$
  - Agents α - bounded sub-optimal agents
  - Recognition System σ - with sensor model S by which agents are observed
- A **Design Model: Δ**
  - Possible modifications $\mathcal{M}$
  - Modification transition function $\Theta : \mathcal{M} \times \mathcal{R} \rightarrow \mathcal{R}$
  - Design constraints $\Phi : \vec{\mathcal{M}} \times \mathcal{R} \rightarrow \{0, 1\}$

Observable Projection

The way a path is observed via the sensor model S

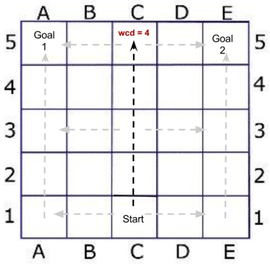
Non-distinctive Path

A path is non-distinctive if it has an observable projection, which is also the observable projection of a path leading to a different goal and distinctive otherwise

Worst Case Distinctiveness

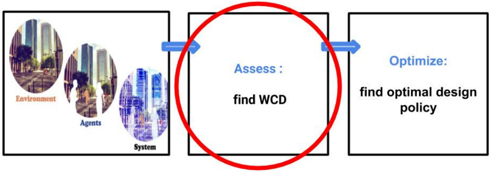
The Worst Case Distinctivenss (*WCD*) is the maximal non-distinctive path

Goal Recognition Design in Deterministic Environments
Example



- Observable Projection
- Non-distinctive path
- Worst Case Distinctivenss (*WCD*)

Goal Recognition Design: Assessing a Model

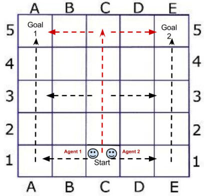


- We are seeking the Worst Case Distinctiveness (*WCD*)- the maximal non-distinctive agent path
- Basic approach: check all possible behaviors and select the maximal one.
- Does not scale !

WCD Computation via Compilation to Planning

WCD Computation via Compilation to Planning

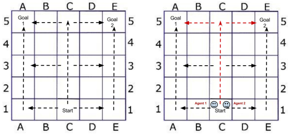
- *WCD* found by compiling a 2-goals goal recognition design problem into a 2-agent planning problem when $n > 2$ done for all pairs
- *latest-split*
  - agents can act separately or together
  - agent 'encouraged' to act together by a small discount ϵ



WCD Computation via Compilation to Planning

Goal Recognition Design - Compilation

- A goal recognition design problem is a 6-tuple $\Pi = \langle P, s_0, \mathbb{G}, A, cost \rangle$
  - P
  - $s_0 \subseteq P$
  - $\mathbb{G} \subseteq 2^P$
  - A is a set of actions
  - Each action is a triple $a = \langle \text{pre}(a), \text{add}(a), \text{del}(a) \rangle$
  - $cost : A \rightarrow \mathbb{R}^{0+}$



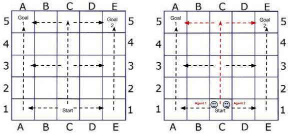
WCD Computation via Compilation to Planning

Goal Recognition Design - Compilation

- A goal recognition design problem is a 6-tuple $\Pi = \langle P, s_0, \mathbb{G}, A, cost \rangle$
  - $P \rightarrow P_i$ for each agent
  - $s_0 \subseteq P \rightarrow$ both agents start at the init state
  - $\mathbb{G} \subseteq 2^P \rightarrow$ each agent aiming at one goal
  - A is a set of actions $\rightarrow$ acting separately or together
  - Each action is a triple $a = \langle \text{pre}(a), \text{add}(a), \text{del}(a) \rangle$
  - $cost : A \rightarrow \mathbb{R}^{0+} \rightarrow \epsilon$ discount for acting together

The optimal solution (produced by any off–the–shelf optimal planner) reveals *WCD*

Later versions accounted for the recognition system's partial observability and for bounded sub-optimal agents



WCD Computation via Compilation to Planning

domain.pddl

```
;; simple Grid-navigation
(define (domain navigator)
(:requirements :strips :typing)
(:types place)
(:predicates
(at ?p - place)
(connected ?p1 ?p2 - place)
)
(:action MOVE
:parameters (?src - place ?dst - place)
:precondition (and (at ?src) (connected ?src ?dst) )
:effect (and (at ?dst) (not (at ?src)))
)
)
```

WCD Computation via Compilation to Planning

template.pddl

```
(define (problem simple5_5)

(:domain navigator)
(:objects
place_0_0 place_0_1 place_0_2 place_0_3 place_0_4
place_1_0 place_1_1 place_1_2 place_1_3 place_1_4
place_2_0 place_2_1 place_2_2 place_2_3 place_2_4
place_3_0 place_3_1 place_3_2 place_3_3 place_3_4
place_4_0 place_4_1 place_4_2 place_4_3 place_4_4
- place
)
(:init
(connected place_0_0 place_1_0) (connected place_0_0 place_0_1)
(connected place_0_1 place_1_1) (connected place_0_1 place_0_0)
...
(at place_2_0)
)
(:goal
(and
<HYPOTHESIS>
))
```

WCD Computation via Compilation to Planning
hyps.dat

```
(at place_0_4)
(at place_4_4)
```

WCD Computation via Compilation to Planning

compiled domain

```
(define (domain navigator)
...
(:constants  agent_0 agent_1 - agent)
...
(:predicates
(at ?p - place ?a - agent)
(connected ?p1 - place ?p2 - place ?a - agent)
(split)
(ag0_done)
)
(:functions (total-cost) - number)
(:action split-agents
 :parameters ()
 :precondition (and (not(split) ))
 :effect (and (split )(increase (total-cost) 0 )) )
(:action agent-0-done
 :parameters ()
 :precondition (and (not(ag0_done) )(split ))
 :effect (and (ag0_done )(increase (total-cost) 0 ))
))
```

WCD Computation via Compilation to Planning

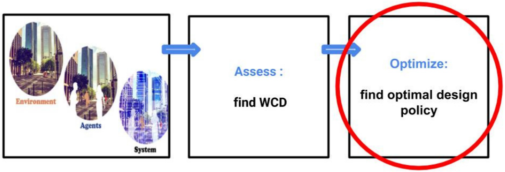
compiled domain -contd.

```
(:action move_together
 :parameters ( ?src - place ?dst - place)
 :precondition (and (at  ?src agent_0)(at  ?src agent_1)
                    (connected  ?src ?dst agent_0)
                    (connected  ?src ?dst agent_1)
                    (not(split) ))
 :effect (and (at  ?dst agent_0)(at  ?dst agent_1)(increase (total-cost) 19980)
         (not (at  ?src agent_0))(not (at  ?src agent_1))))
(:action move_seperate_#0
 :parameters ( ?src - place ?dst - place)
 :precondition (and (at  ?src agent_0)(connected  ?src ?dst agent_0)
               (split)(not(ag0_done) ))
 :effect (and (at  ?dst agent_0)(increase (total-cost) 10000 )
         (not (at  ?src agent_0))))
(:action move_seperate_#1
 :parameters ( ?src - place ?dst - place)
 :precondition (and (at  ?src agent_1)(connected  ?src ?dst agent_1)
               (split )(ag0_done ))
 :effect (and (at  ?dst agent_1)(increase (total-cost) 10000 )
         (not (at  ?src agent_1))))
```

WCD Computation via Compilation to Planning
compiled problem

```
(define (problem simple5_5)
(:domain navigator)
(:objects
place_0_0 place_0_1 place_0_2 place_0_3 place_0_4
...)
(:init
(connected place_0_0 place_1_0  agent_0)
(connected place_0_0 place_1_0  agent_1)
...
(at place_2_0 agent_0)(at place_2_0 agent_1)
( not ( split ) )(= (total-cost) 0)
)
(:goal
(and
(at place_4_4 agent_0)(at place_0_4 agent_1)))
(:metric minimize (total-cost)))
```

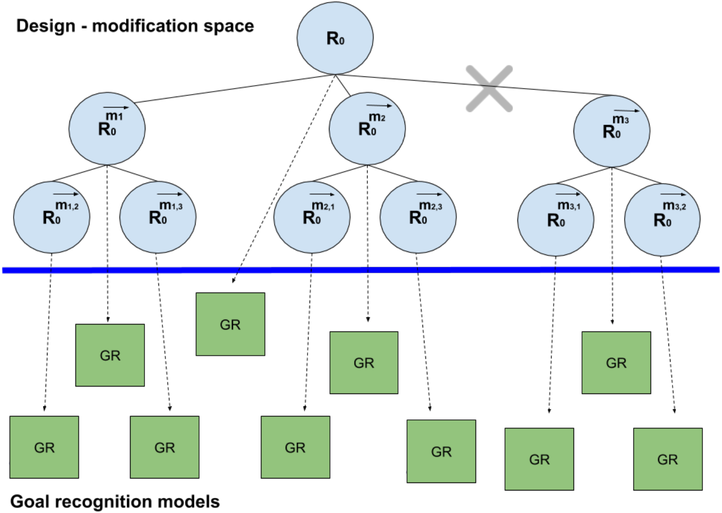
Goal Recognition Design: Minimizing WCD



- We are seeking an optimal sequence of modifications
- Basic approach: check all possible modification sequences and select the best one
- Does not scale !

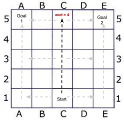
Goal Recognition Design: Minimizing WCD

Searching for an optimal redesign sequence



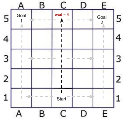
Goal Recognition Design: Minimizing WCD
Safe Pruning for GRD using Generalized Strong Stubborn Sets

- At each stage, we prune modifications that have no effect on the *WCD paths*, the maximal non-distinctive paths
  - found as part of the *WCD* calculation (no extra cost!)



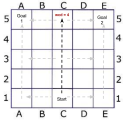
Goal Recognition Design: Minimizing WCD
Safe Pruning for GRD using Generalized Strong Stubborn Sets

- At each stage, we prune modifications that have no effect on the *WCD paths*, the maximal non-distinctive paths
  - found as part of the *WCD* calculation (no extra cost!)



Why is this working ?

- At each stage, we prune modifications that have no effect on the *WCD paths*, the maximal non-distinctive paths
  - found as part of the *WCD* calculation (no extra cost!)



Why is this working ?

- The unpruned modifications form a Generalized Strong Stubborn Set for Independent, Persistent, Monotonic-nd models

Goal Recognition Design: Minimizing WCD
Safe Pruning for GRD using Generalized Strong Stubborn Sets

- Generalized Strong Stubborn Sets for safe pruning

  (Valmari 1989, Wehrle and Helmert 2014)
  - Original: in a solvable state, for at least one strongly optimal plan, there exists a permutation which is not pruned.
  - GRD: in a non-terminal node, for at least one strongly optimal modification sequence, there exists a permutation which is not pruned.

  Equivalently:

  For every node, the modifications not pruned include the first modification in a sequence that minimizes the wcd of the goal recognition model represented by the node.
- Independent, Persistent, Monotonic-nd models
  - Independent - application order is not important
  - Persistent - valid sequences can't have invalid prefixes
  - Monotonic-nd - non-distinctive paths are not added

Allows us not only to account for all existing *GRD* models, but also to define new modification methods!

Modifications

Single Action Sensor Refinement (SAR)

- Improves the recognition system's sensor model
- Special case: Sensor Placement(SP)

Even with full observability, goal recognition may be delayed

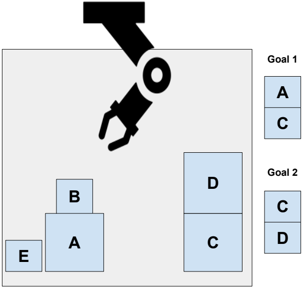
Modifications

Single Action Sensor Refinement (SAR)

- Improves the recognition system's sensor model
- Special case: Sensor Placement(SP)

Action Conditioning (AC)

- Force a partial order between actions
- Special case: Action Removal(AR)



Even with full observability, goal recognition may be delayed

Goal Recognition Design
Optimal Agents

	Agent		Environment		Metrics		Designs		
	Suboptimal Plans	Partial Obs.	Partial Obs.	Stochastic Actions	*wcd*	*ecd*	Action Removal	Sensor Refinement	Action Conditioning
Keren *et al.* (ICAPS 2014)					✓		✓		
Son *et al.* (AAAI 2016)					✓		✓		
Keren *et al.* (AAAI 2015)	✓				✓		✓		
Keren *et al.* (AAAI 2016)	✓		✓		✓		✓		
Keren *et al.* (IJCAI 2016)	✓		✓		✓		✓	✓	
Wayllace *et al.* (IJCAI 2016)				✓	✓		✓		
Wayllace *et al.* (IJCAI 2017)				✓	✓	✓	✓		
Wayllace *et al.* (HSDIP 2018)			✓	✓	✓		✓	✓	
Keren *et al.* (ICAPS 2018)	✓		✓		✓		✓	✓	✓
Keren *et al.* (JAIR 2019)	✓		✓		✓		✓	✓	✓
Keren *et al.* (HSDIP 2019)		✓			✓			✓	

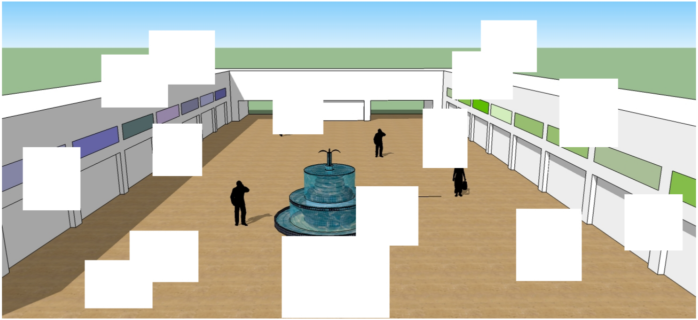
	Agent		Environment		Metrics		Designs		
	Suboptimal Plans	Partial Obs.	Partial Obs.	Stochastic Actions	*wcd*	*ecd*	Action Removal	Sensor Refinement	Action Conditioning
Keren *et al.* (ICAPS 2014)					✓		✓		
Son *et al.* (AAAI 2016)					✓		✓		
Keren *et al.* (AAAI 2015)	✓				✓		✓		
Keren *et al.* (AAAI 2016)	✓		✓		✓		✓		
Keren *et al.* (IJCAI 2016)	✓		✓		✓		✓	✓	
Wayllace *et al.* (IJCAI 2016)				✓	✓		✓		
Wayllace *et al.* (IJCAI 2017)				✓	✓	✓	✓		
Wayllace *et al.* (HSDIP 2018)			✓	✓	✓		✓	✓	
Keren *et al.* (ICAPS 2018)	✓		✓		✓		✓	✓	✓
Keren *et al.* (JAIR 2019)	✓		✓		✓		✓	✓	✓
Keren *et al.* (HSDIP 2019)		✓			✓			✓	

Goal Recognition Design
From Optimal to Bounded Sub-Optimal Agents

- In (Keren, Gal and Karpas ICAPS 2014) - agents are optimal
- In (Keren, Gal and Karpas AAAI15), we account for sub-optimal agents (still using classical planning techniques!)

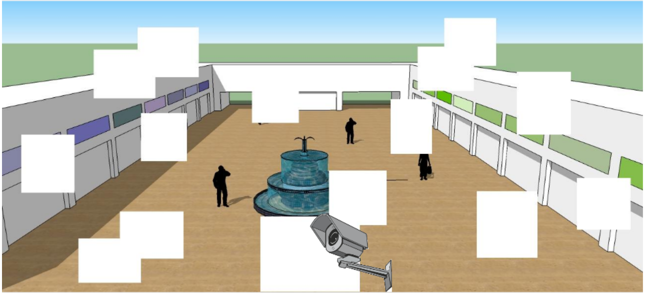
	Agent		Environment		Metrics		Designs		
	Suboptimal Plans	Partial Obs.	Partial Obs.	Stochastic Actions	*wcd*	*ecd*	Action Removal	Sensor Refinement	Action Conditioning
Keren *et al.* (ICAPS 2014)					✓		✓		
Son *et al.* (AAAI 2016)					✓		✓		
Keren *et al.* (AAAI 2015)	✓				✓		✓		
Keren *et al.* (AAAI 2016)	✓		✓		✓		✓		
Keren *et al.* (IJCAI 2016)	✓		✓		✓		✓	✓	
Wayllace *et al.* (IJCAI 2016)				✓	✓		✓		
Wayllace *et al.* (IJCAI 2017)				✓	✓	✓	✓		
Wayllace *et al.* (HSDIP 2018)			✓	✓	✓		✓	✓	
Keren *et al.* (ICAPS 2018)	✓		✓		✓		✓	✓	✓
Keren *et al.* (JAIR 2019)	✓		✓		✓		✓	✓	✓
Keren *et al.* (HSDIP 2019)		✓			✓			✓	

In (Keren, Gal and Karpas AAAI 2016, IJCAI 2016, ICAPS 2018, JAIR 2019) we account for noisy and partial sensors by which agent are observed (and still use compilations to classical planning!)



In (Keren, Gal and Karpas AAAI 2016, IJCAI 2016, ICAPS 2018, JAIR 2019) we account for noisy and partial sensors by which agent are observed (and still use compilations to classical planning!)

Sensor refinement as a way to reduce *WCD*.



Goal Recognition Design
Design for Improved Privacy

	Agent		Environment		Metrics		Designs		
	Suboptimal Plans	Partial Obs.	Partial Obs.	Stochastic Actions	*wcd*	*ecd*	Action Removal	Sensor Refinement	Action Conditioning
Keren *et al.* (ICAPS 2014)					✓		✓		
Son *et al.* (AAAI 2016)					✓		✓		
Keren *et al.* (AAAI 2015)	✓				✓		✓		
Keren *et al.* (AAAI 2016)	✓		✓		✓		✓		
Keren *et al.* (IJCAI 2016)	✓		✓		✓		✓	✓	
Wayllace *et al.* (IJCAI 2016)				✓	✓		✓		
Wayllace *et al.* (IJCAI 2017)				✓	✓	✓	✓		
Wayllace *et al.* (HSDIP 2018)			✓	✓	✓		✓	✓	
Keren *et al.* (ICAPS 2018)	✓		✓		✓		✓	✓	✓
Keren *et al.* (JAIR 2019)	✓		✓		✓		✓	✓	✓
Keren *et al.* (HSDIP 2019)		✓			✓			✓	

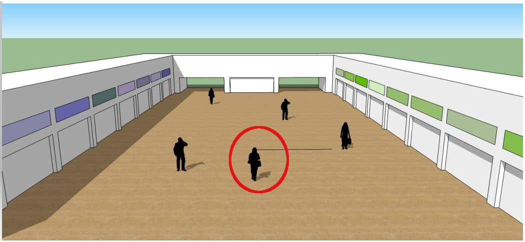
Cloaking : How long can an agent keep his goal ambiguous?

(Keren, Gal and Karpas IJCAI 2016)

A user can choose a path that potentially maximizes its privacy

The *WCD*-path that allows him to stay ambiguous for at most *WCD* steps

Cloaking : How long can an agent keep his goal ambiguous?



A user can choose a path that potentially maximizes its privacy

The *WCD*-path that allows him to stay ambiguous for at most *WCD* steps

Goal Recognition Design (GRD) in Stochastic Domains

Goal Recognition Design Models

	Agent		Environment		Metrics		Designs		
	Suboptimal Plans	Partial Obs.	Partial Obs.	Stochastic Actions	*wcd*	*ecd*	Action Removal	Sensor Refinement	Action Conditioning
Keren *et al.* (ICAPS 2014)					✓		✓		
Son *et al.* (AAAI 2016)					✓		✓		
Keren *et al.* (AAAI 2015)	✓				✓		✓		
Keren *et al.* (AAAI 2016)	✓		✓		✓		✓		
Keren *et al.* (IJCAI 2016)	✓		✓		✓		✓	✓	
Wayllace *et al.* (IJCAI 2016)				✓	✓		✓		
Wayllace *et al.* (IJCAI 2017)				✓	✓	✓	✓		
Wayllace *et al.* (HSDIP 2018)			✓	✓	✓		✓	✓	
Keren *et al.* (ICAPS 2018)	✓		✓		✓		✓	✓	✓
Keren *et al.* (JAIR 2018)	✓		✓		✓		✓	✓	✓
Keren *et al.* (HSDIP 2019)		✓			✓			✓	

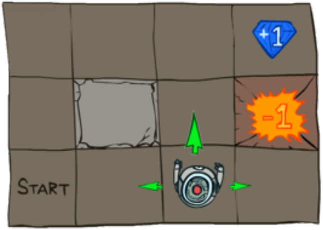
Model: Stochastic Goal Recognition Design (S-GRD)

Outline

- background
- problem definition
- *wcd* evaluation - using augmented MDPs and VI
- minimizing *wcd* - using heuristics for safe pruning
- sample results

A Markov Decision Process (MDP) is a tuple $\langle \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{C}, \mathbf{G}, s_0 \rangle$:

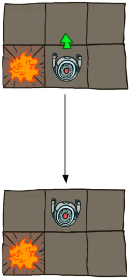
- $\mathbf{S}$ is a set of states.
- $\mathbf{A}$ is a set of actions.
- $\mathbf{T} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow [0, 1]$ is a transition function.
- $\mathbf{C} : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow \mathbb{R}$ is a cost function.
- $\mathbf{G}$ is a set of goal states.
- s_0 is an initial starting state.



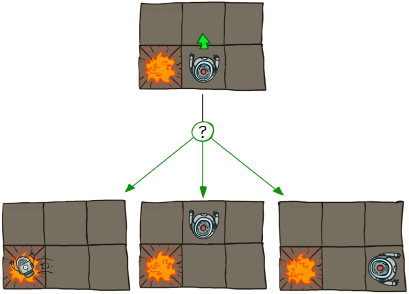
Note: Using rewards instead of costs for these examples. Cost = -Reward

- States = locations; Actions = movements (i.e., N, S, E, W).
- Transitions = successful movement with probability 0.8, left and right with 0.1 each.
- Rewards = +1 at goal, -1 at pit, and -0.1 at every other state.
- Goal state = top right cell; Initial state = bottom left cell.

Deterministic Environment



Stochastic Environment



A solution to an MDP is a policy $\pi : \mathbf{S} \rightarrow \mathbf{A}$.

The expected cost function V^π for a policy π is:

$$V^\pi(s) = \sum_{s \in \mathbf{S}} T(s, \pi(s), s')[C(s, \pi(s), s') + V^\pi(s')]$$

The optimal policy π^* is the one with the minimum expected cost:

$$\pi^*(s) = \operatorname*{argmin}_{a \in \mathbf{A}} \sum_{s \in \mathbf{S}} T(s, a, s')[C(s, a, s') + V^\pi(s')]$$

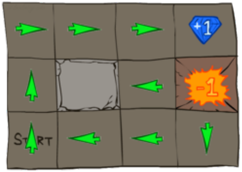
To find an optimal MDP policy, the most popular way is through the Value Iteration (VI) algorithm.

It iteratively updates the value of each state using the Bellman update equation:

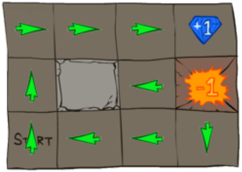
$$V_k(s) = \min_{a \in \mathbf{A}} \sum_{s \in \mathbf{S}} T(s, a, s')[C(s, a, s') + V_{k-1}(s')]$$

until convergence, where

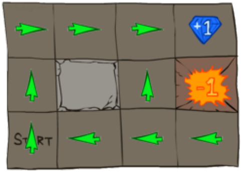
$$\forall s \in \mathbf{S} : V_k(s) - V_{k-1}(s) < \epsilon$$



R(s) = -0.01

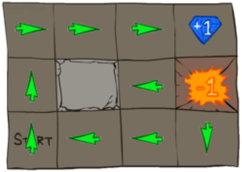


$R(s) = -0.01$

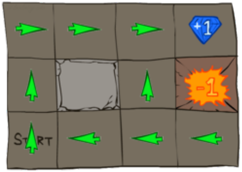


$R(s) = -0.03$

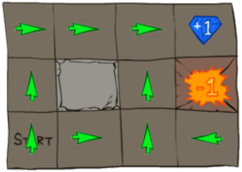
Goal Recognition Design in Stochastic Environments
Background: MDP Policies



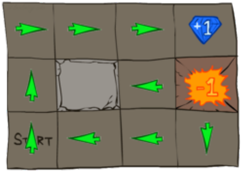
R(s) = -0.01



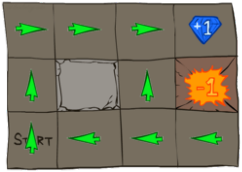
R(s) = -0.03



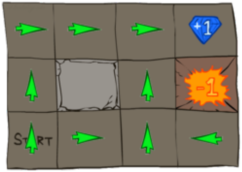
R(s) = -0.40



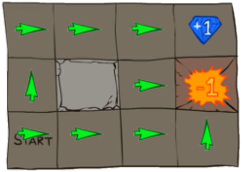
$R(s) = -0.01$



$R(s) = -0.03$



$R(s) = -0.40$



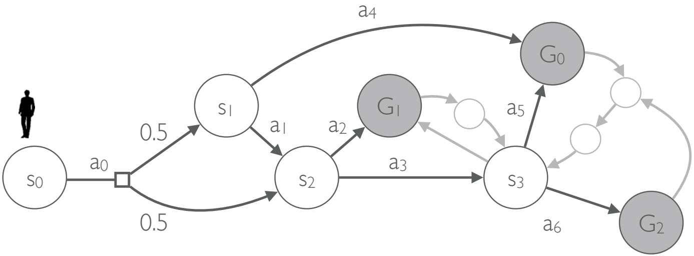
$R(s) = -2.00$

- Marauder's Map:
  - `https://www.youtube.com/watch?v=vNc43oKqQzg`
  - Time: 1:04 – 1:40
- Moving Stairs in Hogwarts:
  - `https://www.youtube.com/watch?v=uFvizAQHJz8`
  - Time: 0:00 – 0:30

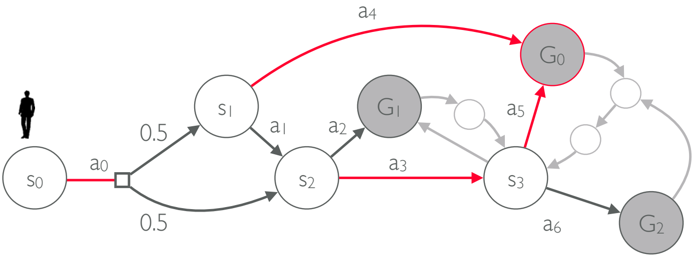
- Marauder's Map:
  - https://www.youtube.com/watch?v=vNc43oKqQzg
  - Time: 1:04 – 1:40
- Moving Stairs in Hogwarts:
  - https://www.youtube.com/watch?v=uFvizAQHJz8
  - Time: 0:00 – 0:30
- Combined together:
  How do we recognize where is Harry Potter trying to go if we observe him on the Marauder's map in Hogwarts?

Stochastic Goal Recognition Design (S-GRD):

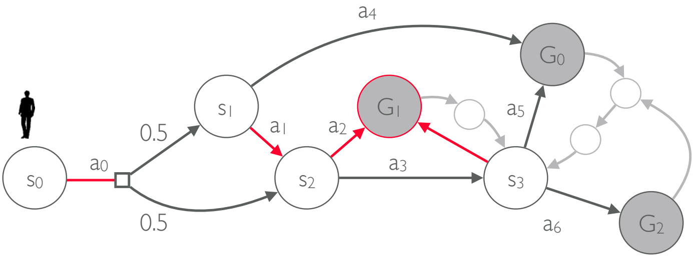
- An initial goal recognition model $R \in \mathcal{R}$
  - MDP without goals $\langle \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{C}, s_0 \rangle$
  - Possible goals $\mathbb{G}$
  - Agents α - optimal agents
  - Recognition System σ - with sensor model S by which agents are observed
- A **Design Model** Δ
  - Possible modifications $\mathcal{M}$
  - Modification transition function $\Theta : \mathcal{M} \times \mathcal{R} \rightarrow \mathcal{R}$
  - Design constraints $\Phi : \vec{\mathcal{M}} \times \mathcal{R} \rightarrow \{0, 1\}$



a4
G0
s1
G1
a5
0.5
a1
a2
a0
a3
s0
s2
s3
0.5
a6
G2



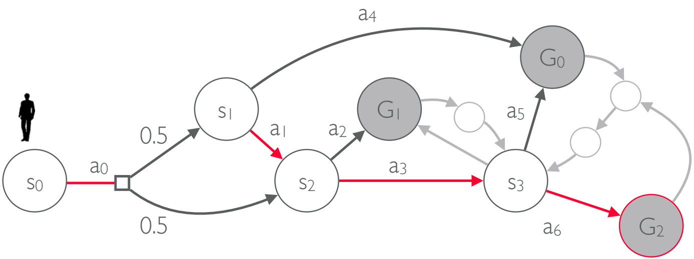
- $\Pi_{leg}(g)$: Set of *legal* (partial) policies for a goal g:
  - A (partial) policy $\pi \in \Pi_{leg}(g)$ if it is a subset of an optimal policy for goal g
  - Note: A partial policy can be a legal policy for multiple goals. Set of all goals for a policy π is $G(\pi)$



- $\Pi_{leg}(g)$: Set of *legal* (partial) policies for a goal g:
  - A (partial) policy $\pi \in \Pi_{leg}(g)$ if it is a subset of an optimal policy for goal g
  - Note: A partial policy can be a legal policy for multiple goals. Set of all goals for a policy π is $G(\pi)$

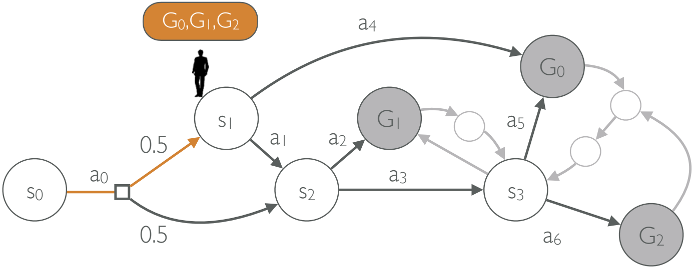
Goal Recognition Design in Stochastic Environments

Problem Definition (cont.)

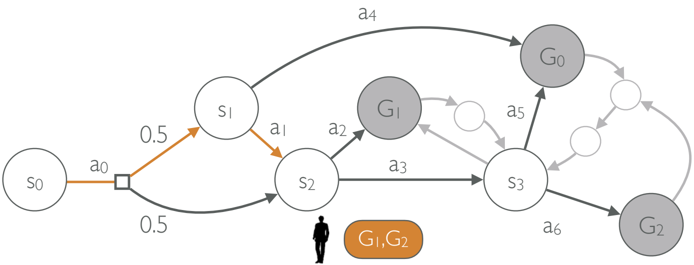


- $\Pi_{leg}(g)$: Set of *legal* (partial) policies for a goal g:
  - A (partial) policy $\pi \in \Pi_{leg}(g)$ if it is a subset of an optimal policy for goal g
  - Note: A partial policy can be a legal policy for multiple goals. Set of all goals for a policy π is $G(\pi)$

- Recognizing goals of agents:
  - Ideal if we can observe policies and infer the goal of the agent through their policy
  - But we only observe the agent's trajectory $\tau = \langle s_0, a_1, s_1, \ldots \rangle$
  - Or worse, the agent's state trajectory only $\tau = \langle s_0, s_1, \ldots \rangle$
  - $G(\tau)$: Set of possible goals of trajectory τ:
    - Goal g is a possible goal of τ iff $\exists \pi \in \Pi_{leg}(g)$ such that τ is a possible trajectory for.

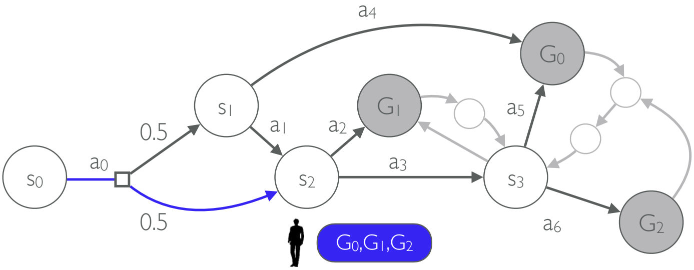


- A trajectory τ is *non-distinctive* if $|G(\tau)| > 1$.

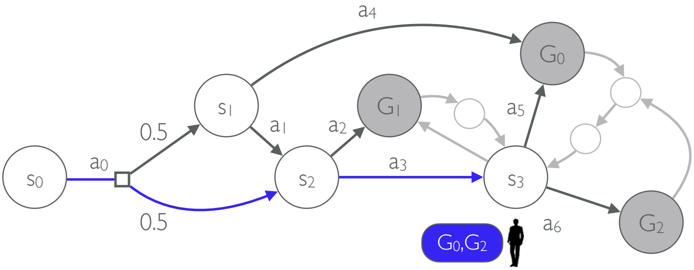


- A trajectory τ is *non-distinctive* if $|G(\tau)| > 1$.

Goal Recognition Design in Stochastic Environments
Problem Definition (cont.)

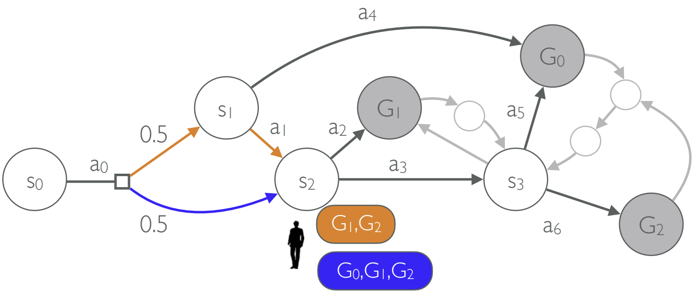


- A trajectory τ is *non-distinctive* if $|G(\tau)| > 1$.



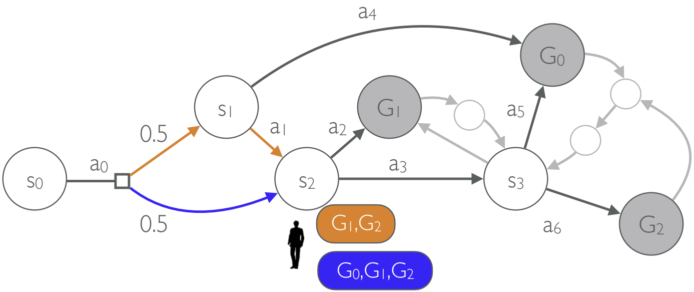
- A trajectory τ is *non-distinctive* if $|G(\tau)| > 1$.

Goal Recognition Design in Stochastic Environments
Problem Definition (cont.)



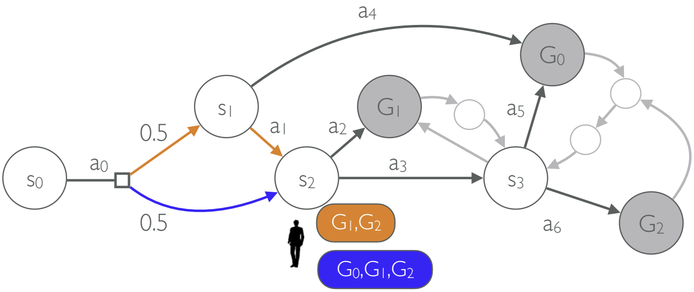
- A trajectory τ is *non-distinctive* if $|G(\tau)| > 1$.

- *Cost* of a trajectory $C(\tau)$:
  - If the trajectory is a state-action trajectory, then $C(\tau)$ is the sum of the cost of all actions in that sequence.
  - If the trajectory is a state trajectory, then $C(\tau)$ is the maximum cost across all possible state-action trajectories that could have resulted in the observed state trajectory.



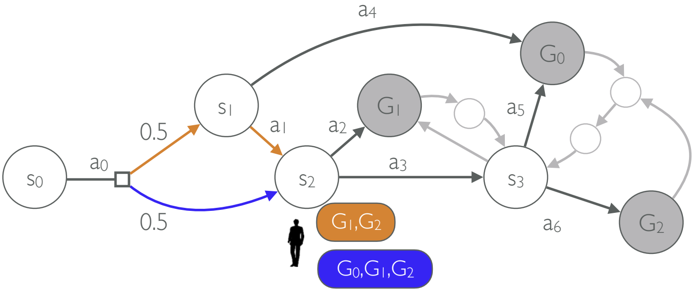
- Cost of $(s_0, a_0, s_1, a_1, s_2) = 2$
- Cost of $(s_0, a_0, s_2) = 1$

- *Cost* of a trajectory $C(\tau)$:
  - If the trajectory is a state-action trajectory, then $C(\tau)$ is the sum of the cost of all actions in that sequence.
  - If the trajectory is a state trajectory, then $C(\tau)$ is the maximum cost across all possible state-action trajectories that could have resulted in the observed state trajectory.
- *Cost* of a partial policy $C(\pi) = \sum_{\tau} P_{\pi}(\tau) C(\tau)$ is the expected cost of all possible trajectories of that policy



- Cost of $(s_0, a_0, s_1, a_1, s_2) = 2$
- Cost of $(s_0, a_0, s_2) = 1$
- Cost of policy $\langle s_0 \rightarrow a_0, s_1 \rightarrow a_1 \rangle = 0.5 \cdot 2 + 0.5 \cdot 1 = 1.5$

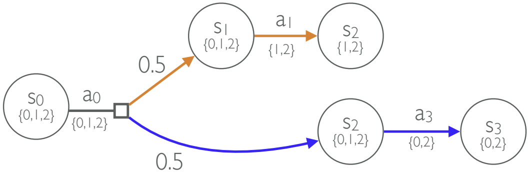
- *Worst-case distinctiveness wcd* $= \max_{\pi \in \Pi_{leg}(G)} C(\pi)$
  - Is the maximum expected cost incurred before an agent *must* reveal its goal.
  - Doesn't use any prior information on the goals; Assumes all goals are equally likely.



- Key observation: Set of possible goals depend on the observed trajectory to a state. *wcd* computation isn't Markovian.
  - If trajectory is $(s_0, a_0, s_1, a_1, s_2)$, next action will reveal its goal
  - If trajectory is (s_0, a_0, s_2), then it can still take a_3 without revealing its goal

Goal Recognition Design in Stochastic Environments

Metrics (cont.)



- Approach: Model the problem using augmented MDPs.
  - *wcd* computation is now Markovian in the augmented state space
  - $wcd = 0.5 \cdot 2 + 0.5 \cdot 2 = 2$

- Compute *wcd* using a modified version of VI on the augmented MDP graph:

$$V_k(s) = \max_{a \in \mathbf{A}} \sum_{s \in \mathbf{S}} T(s, a, s')[C(s, a, s') + V_{k-1}(s')]$$

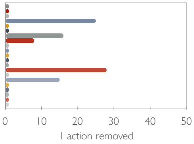
- Is a problem if there are loops in the graph, but our augmented MDP graphs don't have loops

Goal Recognition Design in Stochastic Environments
Metrics (cont.)

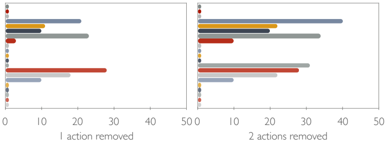
- *Worst-case distinctiveness wcd* $= \max_{\pi \in \Pi_{leg}(G)} C(\pi)$
  - Is the maximum expected cost incurred before an agent *must* reveal its goal.
  - Doesn't use any prior information on the goals; Assumes all goals are equally likely.
- *Expected-case distinctiveness ecd* $= \sum_g P(g) \sum_{\pi \in \Pi_{leg}(g)} \frac{1}{Z} C(\pi)$
  - Uses prior information on the likelihood of each goal being the true goal
  - Is like *wcd*, but weighted by the prior
  - Useful when *wcd* is on trajectories to goals with small probabilities

- General idea: Enumerate through all combinations of design options (e.g., all combinations of actions to remove)
- To improve scalability:
  - Pruning: E.g., if removing action $\hat{a}$ results in lengthening the optimal plan to a goal, then no need to consider combinations of $\hat{a}$ with other actions
  - Ordering heuristics: E.g., consider removing actions closer to the agent first

Percentage of *wcd* reduction



Percentage of *ecd* reduction



- The larger the modification, the larger the *wcd* and *ecd* reduction
- *ecd* can be reduced in some problems where *wcd* cannot be reduced
- In some instances, *wcd* and *ecd* cannot be reduced at all

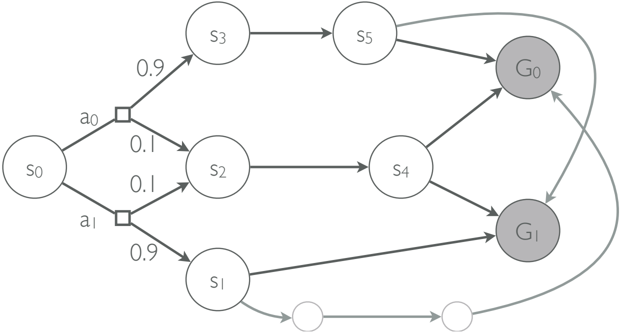
Goal Recognition Design Models

	Agent		Environment		Metrics		Designs		
	Suboptimal Plans	Partial Obs.	Partial Obs.	Stochastic Actions	*wcd*	*ecd*	Action Removal	Sensor Refinement	Action Conditioning
Keren *et al.* (ICAPS 2014)					✓		✓		
Son *et al.* (AAAI 2016)					✓		✓		
Keren *et al.* (AAAI 2015)	✓				✓		✓		
Keren *et al.* (AAAI 2016)	✓		✓		✓		✓		
Keren *et al.* (IJCAI 2016)	✓		✓		✓		✓	✓	
Wayllace *et al.* (IJCAI 2016)				✓	✓		✓		
Wayllace *et al.* (IJCAI 2017)				✓	✓	✓	✓		
Wayllace *et al.* (HSDIP 2018)			✓	✓	✓		✓	✓	
Keren *et al.* (ICAPS 2018)	✓		✓		✓		✓	✓	✓
Keren *et al.* (JAIR 2018)	✓		✓		✓		✓	✓	✓
Keren *et al.* (HSDIP 2019)		✓			✓			✓	

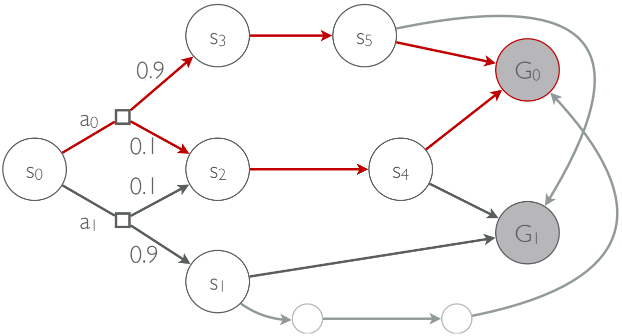
Model: Partially-Observable Stochastic Goal Recognition Design (POS-GRD)

Goal Recognition Design in Stochastic Environments

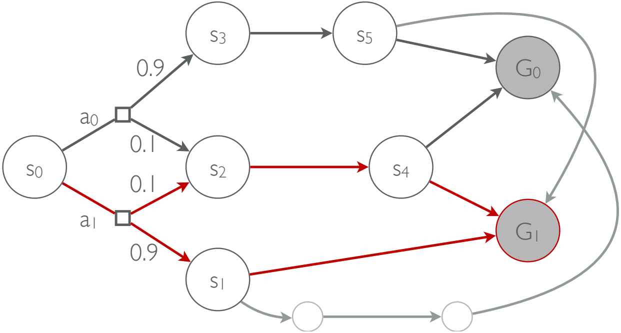
Partially-Observable S-GRD



Setting: Observable actions, fully-observable states

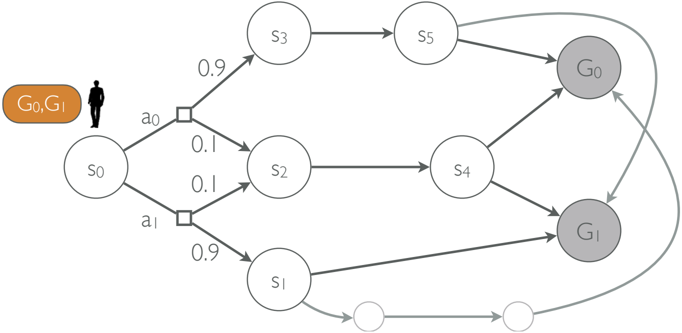


Setting: Observable actions, fully-observable states



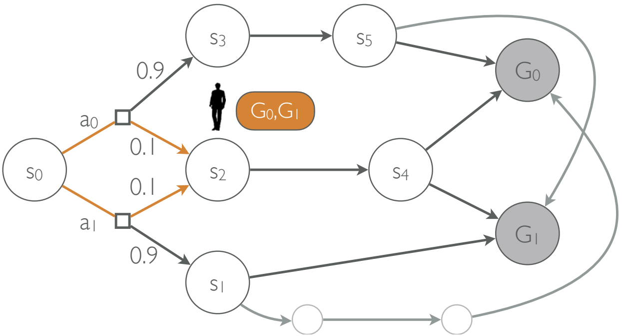
Setting: Observable actions, fully-observable states

wcd $= 0$; first action will reveal the goal of the agent



Setting: **Unobservable** actions, fully-observable states

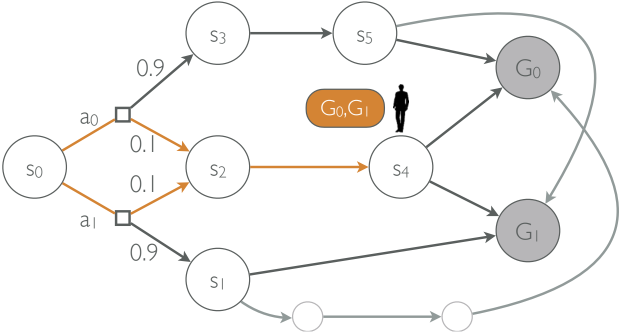
Goal Recognition Design in Stochastic Environments
Partially-Observable S-GRD



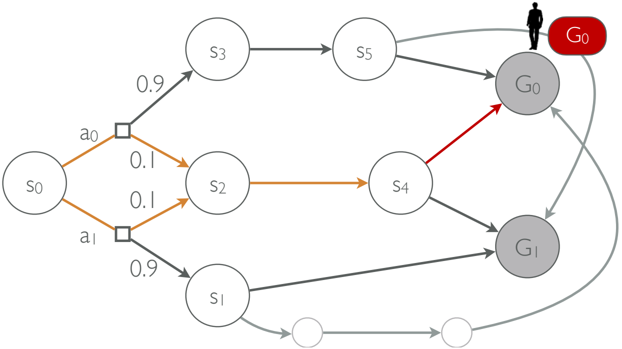
Setting: **Unobservable** actions, fully-observable states

Goal Recognition Design in Stochastic Environments

Partially-Observable S-GRD

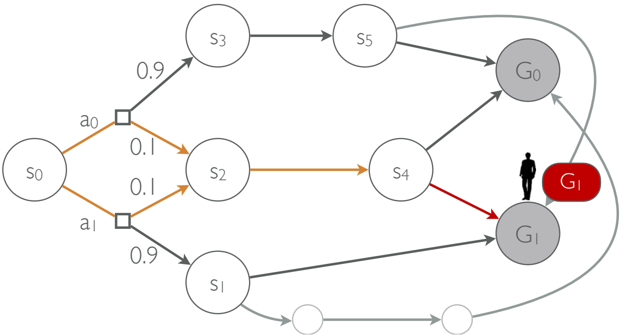


Setting: **Unobservable** actions, fully-observable states



Setting: **Unobservable** actions, fully-observable states

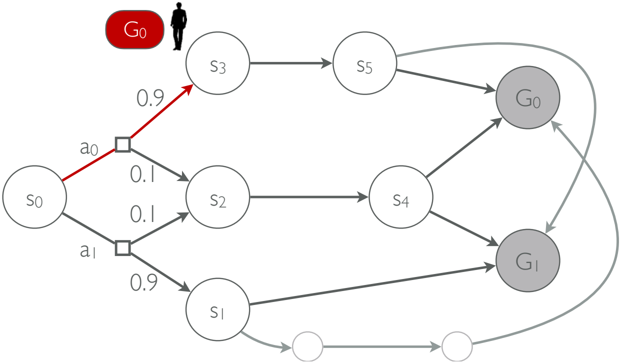
Goal Recognition Design in Stochastic Environments
Partially-Observable S-GRD



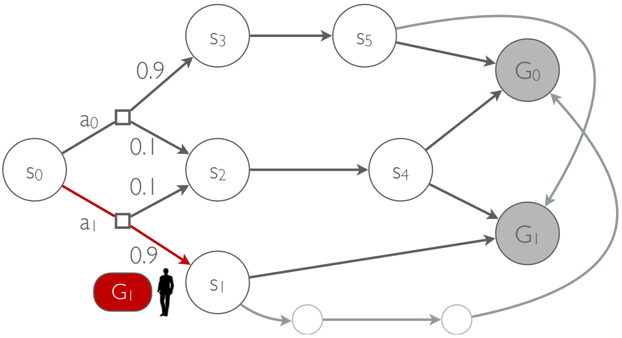
Setting: **Unobservable** actions, fully-observable states

Goal Recognition Design in Stochastic Environments

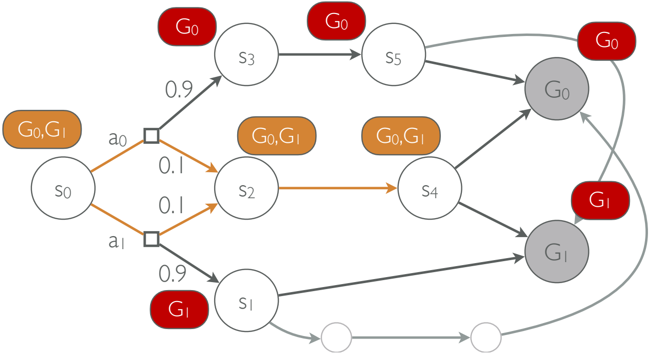
Partially-Observable S-GRD



Setting: **Unobservable** actions, fully-observable states

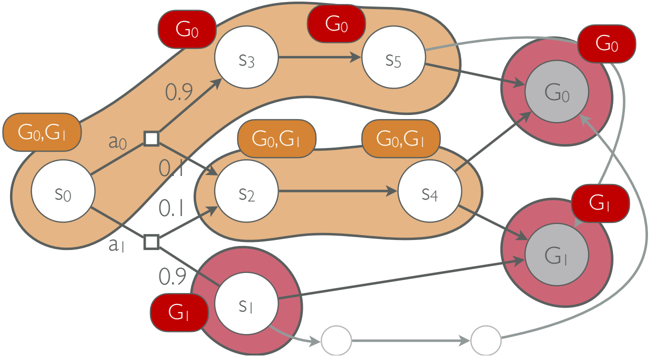


Setting: **Unobservable** actions, fully-observable states

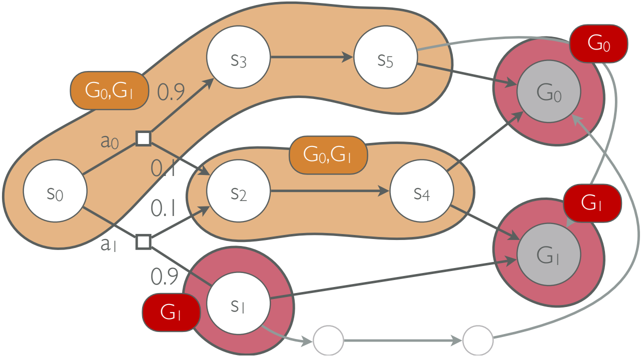


Setting: **Unobservable** actions, fully-observable states

$$wcd = \max(0.9 \cdot 0 + 0.1 \cdot 2 \textbf{ for } a_0, \quad 0.9 \cdot 0 + 0.1 \cdot 2 \textbf{ for } a_1) = 0.2$$



Setting: **Unobservable** actions, **partially-observable** states
Can't differentiate the states that map to the same observation



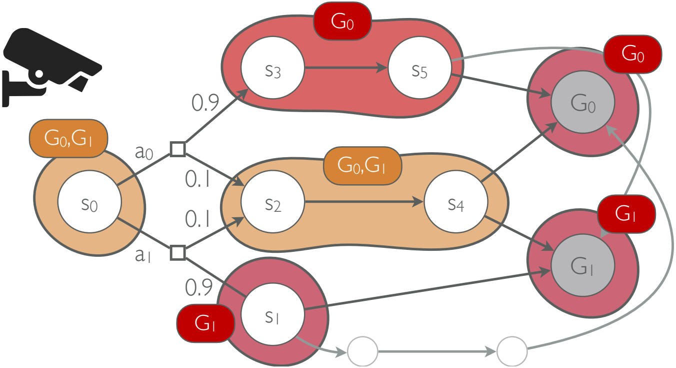
Setting: **Unobservable** actions, **partially-observable** states

$$wcd = \max(0.9 \cdot \mathbf{2} + 0.1 \cdot 2 \textbf{ for } a_0, \quad 0.9 \cdot 0 + 0.1 \cdot 2 \textbf{ for } a_1) = \mathbf{2}$$

Key takeaway: Uncertainty increases *wcd* of the problem.

In our example settings:

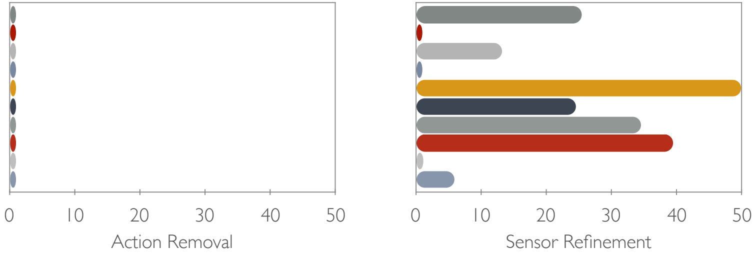
- Observable actions and fully-observable states: $wcd = 0.0$
- Unobservable actions and fully-observable states: $wcd = 0.2$
- Unobservable actions and partially-observable states: $wcd = 2.0$



Partially-Observable S-GRD designs include **sensor placements**!

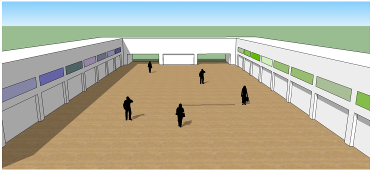
$wcd = \max(0.9 \cdot \mathbf{0} + 0.1 \cdot 2 \textbf{ for } a_0, \quad 0.9 \cdot 0 + 0.1 \cdot 2 \textbf{ for } a_1) = \mathbf{0.2}$

Preliminary results showing percentage of *wcd* reductions:



- Sensor refinement is significantly more effective at reducing *wcd* in partially-observable environments
- Action removal is empirically ineffective
- Future work: Combine both modifications

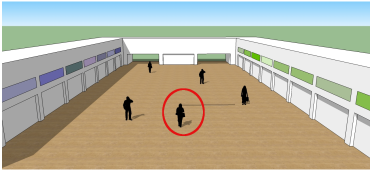
Offline design as a way to facilitate **online** goal recognition

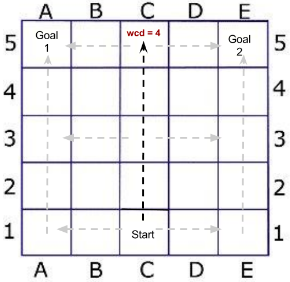


Worst case distinctiveness (wcd) as a measure of model quality

Goal Recognition Design (GRD)

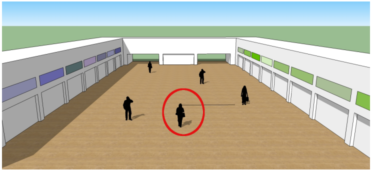
Offline design as a way to facilitate online goal recognition

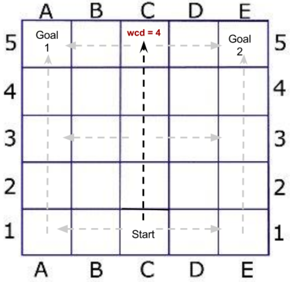


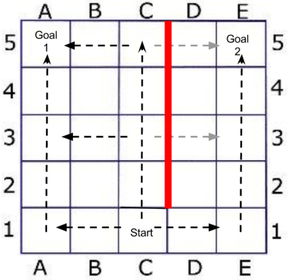


Worst case distinctiveness (wcd) as a measure of model quality

Offline design as a way to facilitate **online** goal recognition

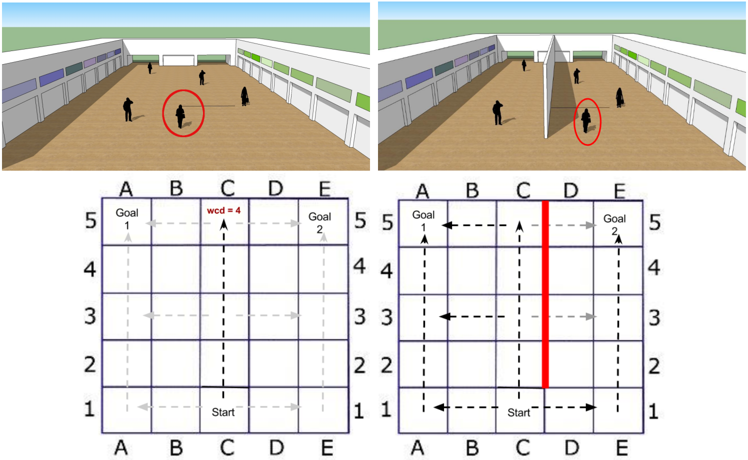






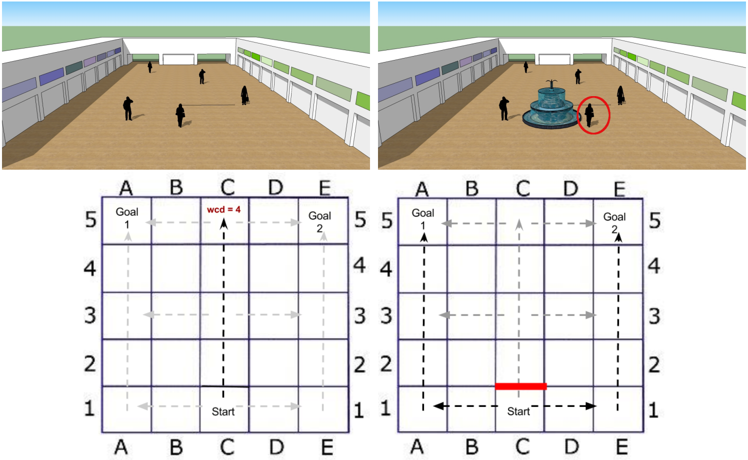
Worst case distinctiveness (wcd) as a measure of model quality

Offline design as a way to facilitate online goal recognition



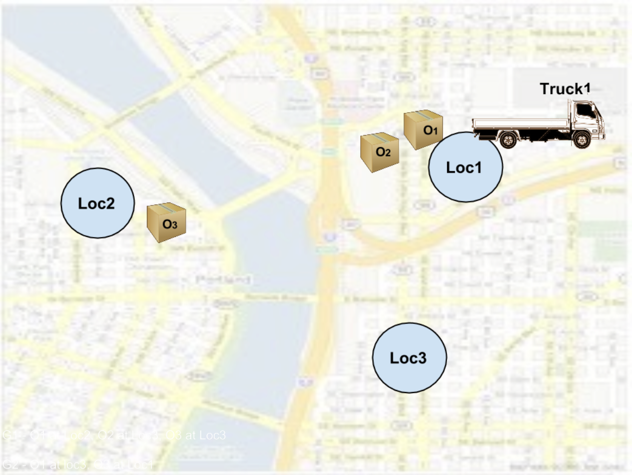
Worst case distinctiveness (wcd) as a measure of model quality

Offline design as a way to facilitate online goal recognition



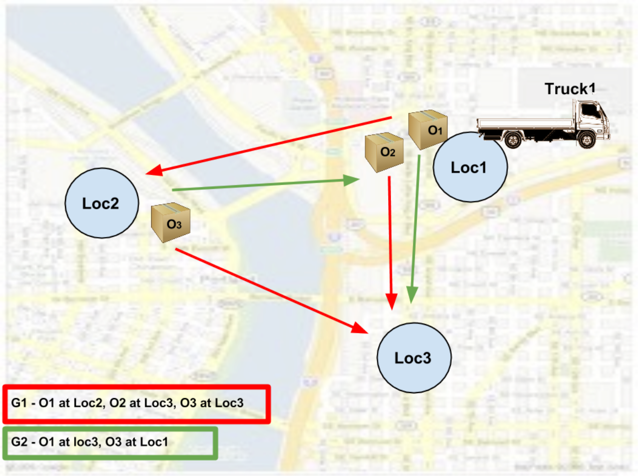
Worst case distinctiveness (wcd) as a measure of model quality

Goal Recognition Design
Fully observable setting - logistics

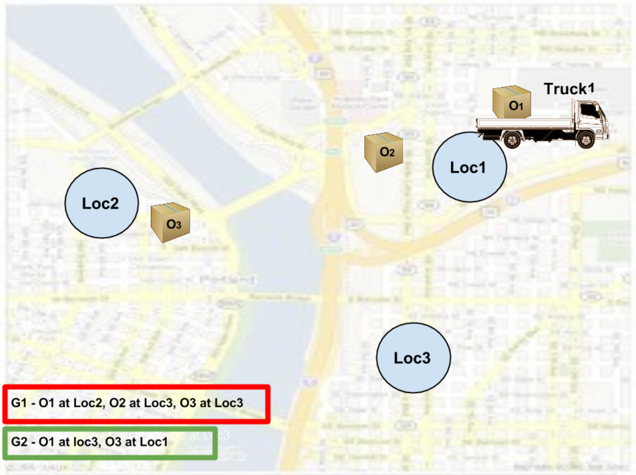


Goal Recognition Design

Fully observable setting - logistics

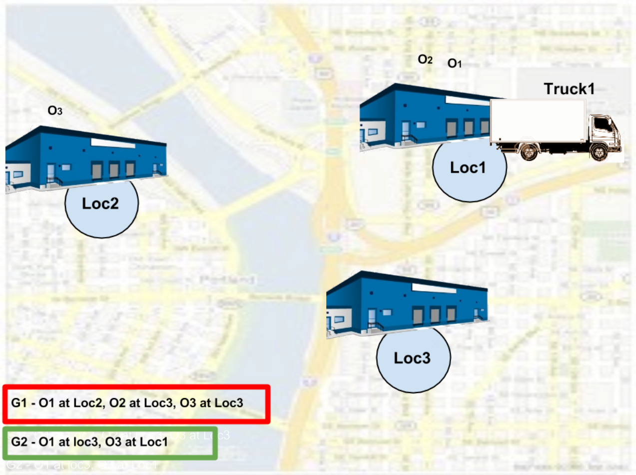


Goal Recognition Design with Non observable actions
WCD=1

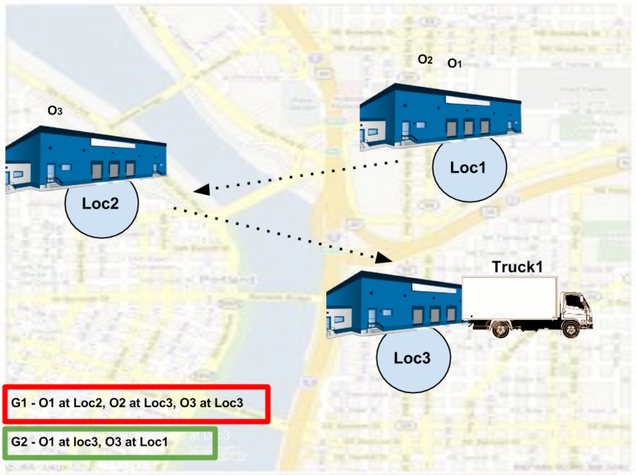


Goal Recognition Design with Non observable actions

Load and Unload actions are not observed

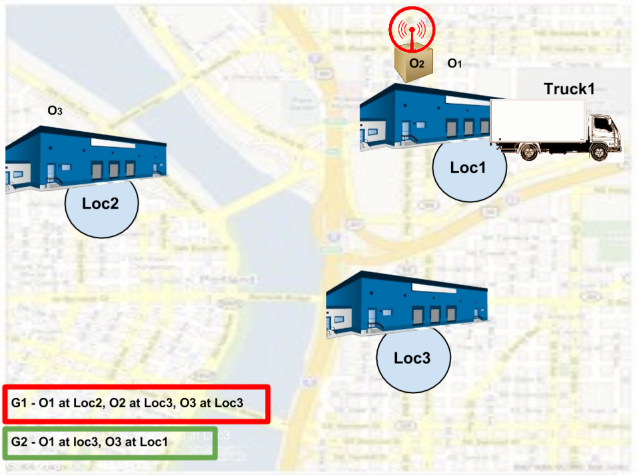


Goal Recognition Design with Non observable actions
WCD=8



Goal Recognition Design with Non observable actions

Sensor placement: WCD=1

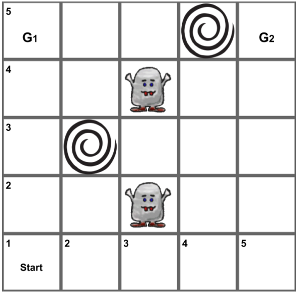


Goal Recognition Design (GRD) for Agents with Partial Knowledge (GRD-APK)

Goal Recognition Design Models

	Agent		Environment		Metrics		Designs		
	Suboptimal Plans	Partial Obs.	Partial Obs.	Stochastic Actions	*wcd*	*ecd*	Action Removal	Sensor Refinement	Action Conditioning
Keren *et al.* (ICAPS 2014)					✓		✓		
Son *et al.* (AAAI 2016)					✓		✓		
Keren *et al.* (AAAI 2015)	✓				✓		✓		
Keren *et al.* (AAAI 2016)	✓		✓		✓		✓		
Keren *et al.* (IJCAI 2016)	✓		✓		✓		✓	✓	
Wayllace *et al.* (IJCAI 2016)				✓	✓		✓		
Wayllace *et al.* (IJCAI 2017)				✓	✓	✓	✓		
Wayllace *et al.* (HSDIP 2018)			✓	✓	✓		✓	✓	
Keren *et al.* (ICAPS 2018)	✓		✓		✓		✓	✓	✓
Keren *et al.* (JAIR 2018)	✓		✓		✓		✓	✓	✓
Keren *et al.* (HSDIP 2019)		✓			✓			✓	

GRD for Agents with Partial Knowledge (GRD-APK)



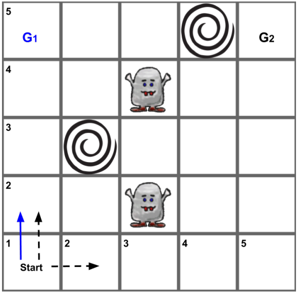
Optimistic (optimal) planning under uncertainty:

- follow a minimal-cost plan to goal
- make as few **assumptions** as possible about unknown variables

Conservative Acting:

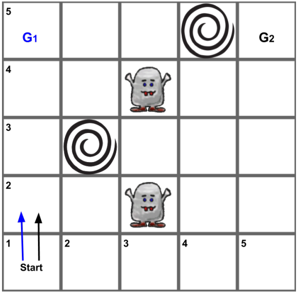
- act only when outcome is known

GRD for Agents with Partial Knowledge (GRD-APK)



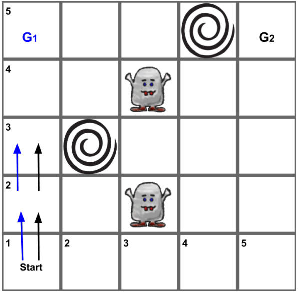
An agent to G_1 goes up
but an agent to G_2 can go either way.

GRD for Agents with Partial Knowledge (GRD-APK)



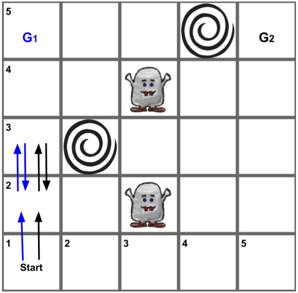
An agent to G_1 goes up
but an agent to G_2 can go either way.

GRD for Agents with Partial Knowledge (GRD-APK)

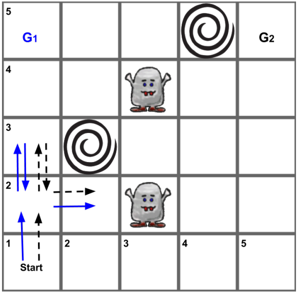


An agent to G_1 goes up
but an agent to G_2 can go either way.

GRD for Agents with Partial Knowledge (GRD-APK)



An agent to G_1 goes up
but an agent to G_2 can go either way.



Goal recognition cannot occur before the actor terminates execution (WCD=4)

GRD for Agents with Partial Knowledge (GRD-APK)

Actor:

- Partially informed
- Modeled as a **contingent planner** (Bonet and Geffner 2012)
- Information as sensors (C, L): the conditions C under which the true value of L is revealed

Recognizer:

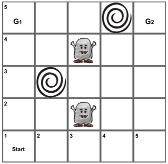
- Has perfect information
- Can selectively reveal information to the actor to recognize its goal as quickly as possible
- Applies **sensor extensions**- add sensors to the actor

Information Shaping

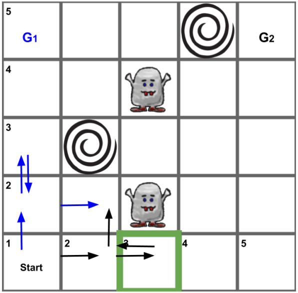
Selecting which information to reveal to minimize *WCD*

Corresponds both to (direct) communication and (indirect) sensor distribution.

- Direct: $(C = True, L = \text{Safe(2,2)})$
- Indirect: $(C = \text{AgentAt(1,2)}, L = \text{StenchAt(2,2)})$



GRD for Agents with Partial Knowledge (GRD-APK)



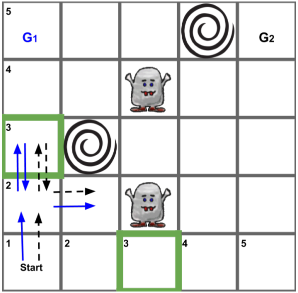
"Cell (3,1) is safe"

Information shaping - reveal safe cells

The first step reveals the actor's goal

(WCD=0)

GRD for Agents with Partial Knowledge (GRD-APK)



"(3,1)&(1,3) are safe"

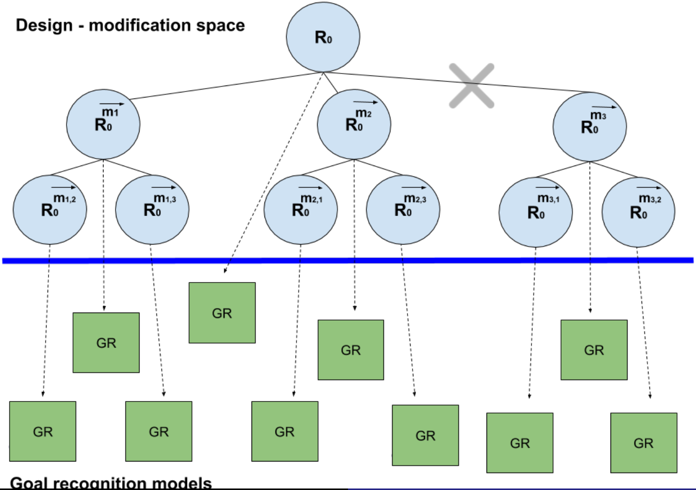
Information shaping - reveal safe cells

We are back to the initial situations

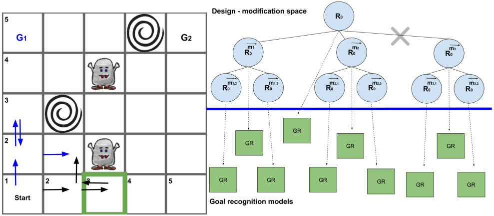
(WCD=4)

Information shaping is non-monotonic and needs to be applied carefully

Searching for a design solution that minimizes *WCD*



Searching for a design solution that minimizes *WCD*



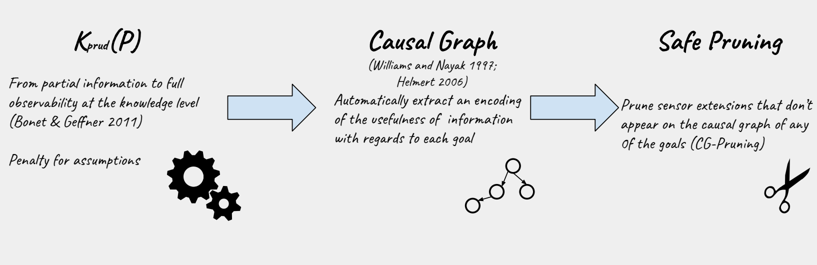
Information shaping is challenging because:

- it's **non-monotonic** - more information doesn't guarantee earlier recognition
- the space of options is too large to explore exhaustively
- previous approaches for **safe** pruning don't hold here

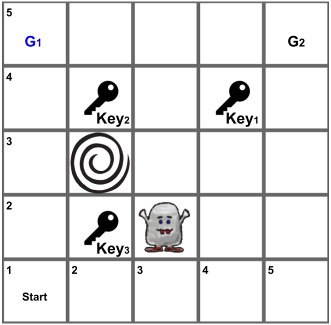
New techniques are needed !

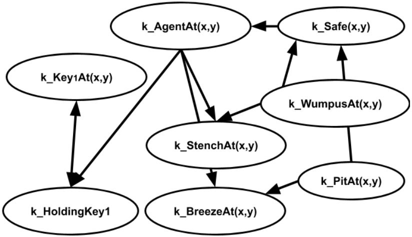
GRD for Agents with Partial Knowledge (GRD-APK)
Solution Approach

We use techniques from classical planning to **automatically** find sensor extensions that can be **safely** pruned



GRD for Agents with Partial Knowledge (GRD-APK)
Solution Approach





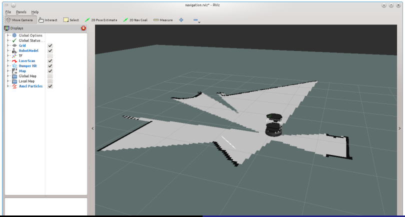
Causal graph analysis automatically detects information that is (ir)relevant to each goal

Applies to any goal recognition setting that can be controlled, and in which agents are only partially informed.

Example applications:

- Assistive cognition
- Intrusion detection
- Human-robot collaboration

Current focus: a robotic navigation setting, in which the map (occupancy grid) used by the robot can be manipulated



Conclusions

Goal Recognition Design

Summary: What has been done?

	Agent		Environment		Metrics		Designs		
	Suboptimal Plans	Partial Obs.	Partial Obs.	Stochastic Actions	*wcd*	*ecd*	Action Removal	Sensor Refinement	Action Conditioning
Keren *et al.* (ICAPS 2014)					✓		✓		
Son *et al.* (AAAI 2016)					✓		✓		
Keren *et al.* (AAAI 2015)	✓				✓		✓		
Keren *et al.* (AAAI 2016)	✓		✓		✓		✓		
Keren *et al.* (IJCAI 2016)	✓		✓		✓		✓	✓	
Wayllace *et al.* (IJCAI 2016)				✓	✓		✓		
Wayllace *et al.* (IJCAI 2017)				✓	✓	✓	✓		
Wayllace *et al.* (HSDIP 2018)			✓	✓	✓		✓	✓	
Keren *et al.* (ICAPS 2018)	✓		✓		✓		✓	✓	✓
Keren *et al.* (JAIR 2019)	✓		✓		✓		✓	✓	✓
Keren *et al.* (HSDIP 2019)		✓			✓			✓	

Many other related framework exists, for example:

- Plan Recognition Design (Mirsky *et al.*, PAIR-AAAI 2017)
- Deceptive Path Planning (Masters and Sardina, IJCAI 2017)
- Game-Theoretic Goal Recognition Models with Applications to Security Domains (Ang *et al.*, GameSec 2017)

- Goal Recognition Design: Offline design for efficient online recognition
- Relevant to a variety of applications

Source code for GRD for deterministic environments:
https://github.com/sarah-keren/goal-recognition-design

Goal Recognition Design
Conclusions

- Goal Recognition Design: Offline design for efficient online recognition
- Relevant to a variety of applications

Source code for GRD for deterministic environments:
https://github.com/sarah-keren/goal-recognition-design

Open Challenges:

- Extensions are orthogonal, but not effectively combined yet.
- Evaluation with actual online goal recognition algorithms.
- Mapping and deployment to practical real-world applications.
- ...?