Heuristics for Planning under Partial Observability with Sensing Actions

Shlomi Maliah Guy Shani Ronen Brafman Erez Karpas

ICAPS 2013 Workshop on Heuristic Search for Domain-Independent Planning

Outline

1. Motivation
2. Landmarks for PPOS
3. The Heuristic Contingent Planner
4. Empirical Evaluation

Setting

PPOS

Planning under Partial Observability with Sensing Actions

- Partial observability
- Uncertainty about the initial state
- Actions
  - Deterministic
  - Observation effects
  - Conditional effects
- $\Rightarrow$ Effects of actions during runtime are uncertain

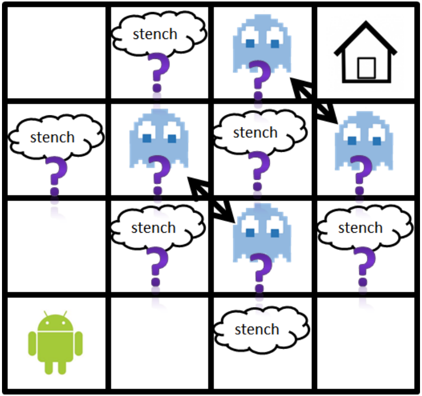
Setting

PPOS

Planning under Partial Observability with Sensing Actions

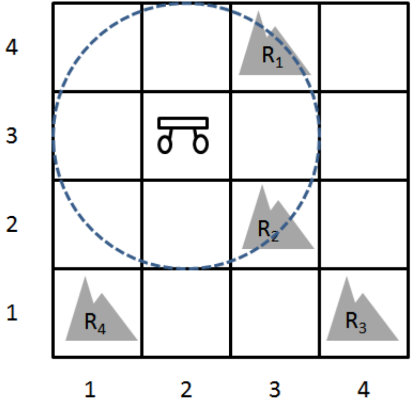
- Partial observability
- Uncertainty about the initial state
- Actions
  - Deterministic
  - Observation effects
  - Conditional effects
- $\Rightarrow$ Effects of actions during runtime are uncertain

Example PPOS Task 1: Wumpus



- Each Wumpus is in one of two possible locations
- Cells adjacent to a wumpus have stench
- Goal is to reach top right corner

Example PPOS Task 2: Mars Rover



- Rocks can be good/bad
- Activating sensor tells whether there are good rocks in range of the antenna
- Goal is to sample a good rock

Formal Setting

- PPOS task $\pi = \langle P, A, \varphi_I, G \rangle$
  - P is a set of propositions
  - A is a set of actions
  - φ_I is a formula that describes the set of possible initial states
  - $G \subseteq P$ is the goal
- Each action $a \in A$ consists of:
  - $pre(a) \subseteq P$ is a set of literals denoting the action's preconditions.
  - $effects(a)$ is a set of pairs (c, e) denoting conditional effects, where c is a conjunction of literals and e is a single literal
  - $obs(a) \subseteq P$ are the propositions whose value is observed when a is executed
- Assume actions either have observations or effects, but not both

PPOS Solution

- Offline:
  - Prepare for every possible outcome in advance
  - Contingent plan / policy — possibly very big
- Online
  - Choose the next action to execute online
  - Between every two sensing actions, there is a sequence of non-sensing actions

PPOS Solution

- Offline:
  - Prepare for every possible outcome in advance
  - Contingent plan / policy — possibly very big
- Online
  - Choose the next action to execute online
  - Between every two sensing actions, there is a sequence of non-sensing actions

PPOS Solution

- Offline:
  - Prepare for every possible outcome in advance
  - Contingent plan / policy — possibly very big
- Online
  - Choose the next action to execute online
  - Between every two sensing actions, there is a sequence of non-sensing actions

Key Insight

In simple domains, the sequence of non-sensing actions between every two sensing actions, can be obtained by solving a classical planning problem over the original state space of the problem

Heuristic Contingent Planner — High Level Control

- If we can achieve the goal without sensing — do so
  - Classical planning, assuming all unknown propositions are false
- Otherwise, choose a reachable sensing action *a*
- Plan to execute *a*, and execute *a*
- Repeat

Heuristic Contingent Planner — High Level Control

- If we can achieve the goal without sensing — do so
  - Classical planning, assuming all unknown propositions are false
- Otherwise, choose a reachable sensing action *a*
- Plan to execute *a*, and execute *a*
- Repeat

Main Contribution

A novel landmark-based heuristic for choosing the next sensing action in PPOS

Outline

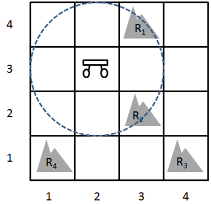
1. Motivation
2. Landmarks for PPOS
3. The Heuristic Contingent Planner
4. Empirical Evaluation

Landmarks

- A landmark is a logical formula over the facts, which must be satisfied by some state along every solution
- Landmark detection is hard even in classical planning
- Challenge for PPOS: must handle uncertainty and sensing
- Our solution:
  - Augment the problem with artificial reasoning actions
  - Join reasoning and observation actions
  - Relax the problem (as for classical planning)

Reasoning Actions: Example

- Suppose we know from φ_I that *good-rock*$_1 \vee$ *good-rock*$_2 \vee$ *good-rock*$_3 \vee$ *good-rock*$_4$
- Suppose we also know
  - $\neg$*good-rock*$_1$
  - $\neg$*good-rock*$_2$
  - $\neg$*good-rock*$_3$
- We create a reasoning action that can deduce that *good-rock*$_4$ holds

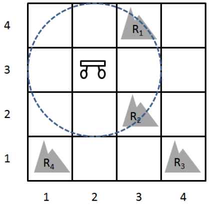


Reasoning Actions

- Proposition $p \in P$ is constant if its value never changes (Geffner and Palacios)
  - Easy to check that p does not appear in effects of any action
- Create reasoning actions from clauses of φ_I containing only constant propositions
  - For disjunctive clause $c = \bigvee_{i=1..k} l_i$, create actions which "reason" that if $k-1$ of the literals are false, then the remaining one is true $A_c = \{a_{l_i}\}_{i=1}^{k}$, with $pre(a_{l_i}) = \bigwedge_{j=1..k, j \neq i} \neg l_j$, and $effects(a_{l_i}) = l_i$
  - For oneof clause $c = oneof_{i=1..k} l_i$, create actions which "reason" that if one of the literals is true, then all the others are false $A_c = \{a_{l_i}\}_{i=1}^{k}$, with $pre(a_{l_i}) = l_i$, and $effects(a_{l_i}) = \bigwedge_{j=1..k, j \neq i} \neg l_j$
- Works only when initial state uncertainty is expressed using such clauses

Joining Immediate Reasoning and Observations: Example

- Action activate-sensor-at-2-3
  - Pre: at-2-3
  - CE: good-$rock_1$ $\longrightarrow$ good-rocks-in-range
  - CE: good-$rock_2$ $\longrightarrow$ good-rocks-in-range
- Observation action observe-good-rocks-in-range observes fact good-rocks-in-range
- The only actions which affect good-rocks-in-range are activate-sensor-at-x-y, which are all mutex
- Create two joined actions, for $i = 1$ and $j = 2$ and for $i = 2$ and $j = 1$, where:
  - Pre: at-2-3 $\wedge$ $\neg$good-$rock_j$
  - Obs: good-rocks-in-range
  - Eff: good-$rock_i$



Joining Immediate Reasoning and Observations

- Can split propositions into 3 sets:
  - Known (e.g., location of rover/android)
  - Unknown, but observable (e.g., stench/good-rocks-in-range)
  - Unknown and unobservable (e.g., location of wumpus/"goodness" of specific rock)

Joining Immediate Reasoning and Observations

- Let a be an action with conditional effects $\{(c_i, e)\}_{i=1}^{k}$ where
  - c_i is unknown and unobservable, and
  - e is observable, and
  - There is no other action that affects the value of e which is not mutually exclusive with a
- Let a_{obs} be an action that observes e
- We create k new actions $a_i \circ a_{obs}$ where:
  - $pre(a_i \circ a_{obs})$ = $pre(a) \wedge pre(a_{obs}) \wedge \bigwedge_{j \neq i} \neg c_j$
  - $obs(a_i \circ a_{obs}) = \{e\}$
  - $effects(a_i \circ a_{obs}) = effects_u(a) \wedge c_i$, where $effects_u(a)$ are the unconditional effects of a.
- Although this is ad-hoc and not complete, this works in many benchmarks

Joining Immediate Reasoning and Observations

- Let a be an action with conditional effects $\{(c_i, e)\}_{i=1}^{k}$ where
  - c_i is unknown and unobservable, and
  - e is observable, and
  - There is no other action that affects the value of e which is not mutually exclusive with a
- Let a_{obs} be an action that observes e
- We create k new actions $a_i \circ a_{obs}$ where:
  - $pre(a_i \circ a_{obs})$ = $pre(a) \wedge pre(a_{obs}) \wedge \bigwedge_{j \neq i} \neg c_j$
  - $obs(a_i \circ a_{obs}) = \{e\}$
  - $effects(a_i \circ a_{obs}) = effects_u(a) \wedge c_i$, where $effects_u(a)$ are the unconditional effects of a.
- Although this is ad-hoc and not complete, this works in many benchmarks

Action Relaxation

1. Ignore delete effects
2. Given action $a \in A$ with k conditional effects $\{(c_i, e_i) : i = 1..k\}$, generate k actions where $a_{(c_i,e_i)}$ is defined by
   - $pre(a_{(c_i,e_i)}) = pre(a) \wedge c_i$
   - $effects(a_{(c_i,e_i)}) = effects(a) \wedge e_i$

Landmark Detection

- We use a landmark detection algorithm for a classical task
- The classical task is generated by:
  - Adding reasoning actions
  - Joining reasoning and observation actions
  - Relaxing the actions in the original task
- One modification to classical landmark detection: “optimistic” sensing — we assume a sensing action will sense the required value

Properties of PPOS Landmark Detection

- Sound
- Not complete
- Only works for certain (common) types of problems
- Example: joining sensing and reasoning fails to capture cases with sequences of actions over unobservable propositions

Outline

1 Motivation

2 Landmarks for PPOS

3 The Heuristic Contingent Planner

4 Empirical Evaluation

Overall Scheme

- If we can achieve the goal without sensing — do so
- Otherwise, choose a reachable sensing action *a*
- Plan to execute *a*, and execute *a*
- Repeat

- Note: reachability is checked in the *relaxed* problem

Choosing the Next Sensing Action

- Denote by *s* what is reachable now
- For each reachable sensing action *a*
  - Assume *a* senses *true*, and denote by s'_+ what is reachable
  - Assume *a* senses *false* and denote by s'_- what is reachable
- Score for *a* is:
  - number of landmarks satisfied in s'_+ and s'_-, but not in *s*
- Tie-breaking by:
  1. number of literals achievable in s'_+ and s'_-, but not in *s*
  2. number of sensing actions achievable in s'_+ and s'_-, but not in *s*
  3. number of actions required from current state before *a* can be executed (in relaxed problem)

Outline

1. Motivation
2. Landmarks for PPOS
3. The Heuristic Contingent Planner
4. Empirical Evaluation

Empirical Results

	HCP		MPSR		SDR		CLG		K-Planner	
Name	A.	T.	A.	T.	A.	T.	A.	T.	A.	T.
cloghuge	55.48	**5.9**			61.17	117.13	**51.76**	8.25		
ebtcs-70	42.32	**1.12**	44.5	22.4	**35.52**	3.18	**36.52**	73.96		
elog7	**20**	**0.32**	23.5	1.4	21.76	0.85	**20.12**	1.4		
CB-9-5	**324**	158.9			392.16	505.48	CSU		358.08	**94.18**
CB-9-7	**425**	373			487.04	833.52	CSU		458.36	**116.63**
doors13	**96.68**	**30**	197.92	105.5	120.8	158.54	105.48	330.73	109.72	37.96
doors15	**137.9**	**52.6**	262.2	190	143.24	268.16			150.88	**55.24**
doors17	**170**	91	368.25	335.3	188	416.88			188.8	**79.24**
localize17			59.8	**230.4**	**45**	928.56	CSU			
unix3	**40.48**	**1.77**	69.7	5.2	56.32	5.47	51.32	18.56	**45.48**	16.87
unix4	94.56	**20.21**	158.6	30.4	151.72	35.22	**90.8**	189.41	87.04	38.81
Wumpus15	**65.08**	9.57	**65**	126.6	120.14	324.32	101.12	330.54	107.64	**7.17**
Wumpus20	90	34	**71.6**	261.1	173.21	773.01	155.32	1432	151.52	**16.03**
Rock 8-12	**105.76**	**6.3**			127.24	113.4				
Rock 8-14	**135**	**9**			142.08	146.75				

Summary

- Presented a method for discovering landmarks in PPOS
- Presented a landmark-based heuristic for choosing the next sensing action in online PPOS
- An online planner using this heuristic performs very well

Thank You