Minimizing Necessary Observations for Nondeterministic Planning

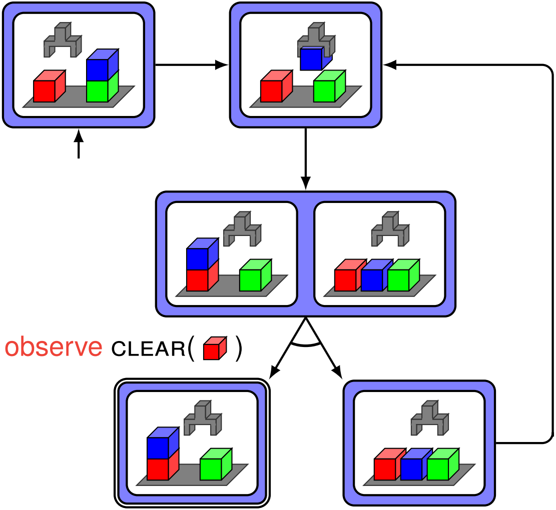
Robert Mattmüller Manuela Ortlieb Erik Wacker

Research Group Foundations of Artificial Intelligence
Department of Computer Science
University of Freiburg, Germany

September 25th, 2014

Motivation

Example: Partially Observable BLOCKSWORLD Domain with Nondeterministic PUT-DOWN Operator



UNI FREIBURG

Motivation

Example: Partially Observable BLOCKSWORLD Domain with Nondeterministic PUT-DOWN Operator

Observing CLEAR predicate sufficient to find solutions.

Initial state known $\Rightarrow$ overhead camera sufficient as a sensor.

Question: How to find minimal sets of variables sufficient for solution existence in arbitrary POND planning tasks?

Remark: Here, “solution” means strong cyclic plan:

- closed: defined for all belief states it may reach, and
- proper: no dead ends

Preliminaries

Formalism

Motivation

Theory

Practical Algorithms

Empirical Evaluation

Conclusion

Problem OBSERVEINCLMIN:

- Input: POND planning task $\Pi = \langle \mathcal{V}, B_0, B_\star, \mathcal{A}, \mathcal{W} \rangle$ with
  - state variables $\mathcal{V}$
  - initial belief state B_0
  - goal description $B_\star$
  - nondeterministic actions $\mathcal{A}$
  - possibly observable variables $\mathcal{W} \subseteq \mathcal{V}$
- Output: Inclusion-minimal set of variables $\mathcal{O} \subseteq \mathcal{W}$ such that there exists a strong cyclic plan for Π observing only variables from $\mathcal{O}$, or NONE if no such set $\mathcal{O}$ exists.

Theorem (Rintanen, 2004)

The strong cyclic plan existence problem for POND planning, PLANEXPOND*, is* 2-EXPTIME*-complete.* □

Theorem

OBSERVEINCLMIN *is* 2-EXPTIME*-complete.*

Proof.

- Trivial reduction from PLANEXPOND
  $\Rightarrow$ OBSERVEINCLMIN 2-EXPTIME-hard.
- Naive algorithm iterating over all subsets of $\mathcal{W}$
  $\Rightarrow$ OBSERVEINCLMIN $\in$ 2-EXPTIME. □

Theory

Question: Can we improve over the naive algorithm from the proof?

Assumption: No obviously irrelevant variables in $\mathcal{W}$.
Ignore variables known in B_0 and never made unknown by any action.

Baseline Algorithm

Simple Greedy Algorithm

function SIMPLEGREEDYSEARCH(Π):
  if Π is unsolvable **then**
    return NONE
  Compute some plan π for Π
  Let $\mathcal{O}$ be the set of variables actually observed in π
  while Π still solvable with some o removed from $\mathcal{O}$ **do**
    Remove o from $\mathcal{O}$
  return $\mathcal{O}$

Theorem

Function SIMPLEGREEDYSEARCH

- *runs in* 2-EXPTIME,
- *correctly solves* OBSERVEINCLMIN. □

Plan Reuse

Motivation

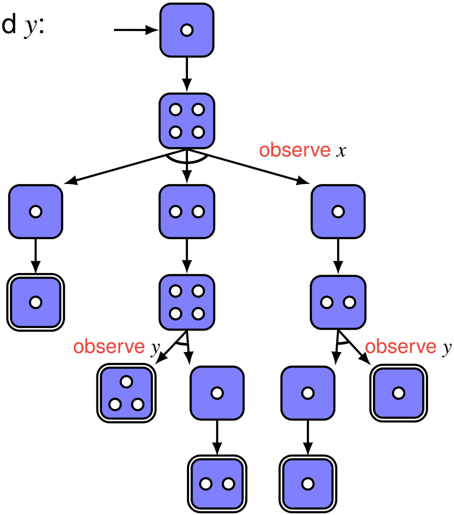
- ✓ SIMPLEGREEDYSEARCH correct
- ✓ SIMPLEGREEDYSEARCH asymptotically optimal
- ✗ SIMPLEGREEDYSEARCH naive and inefficient
  - ➠ Look for ways to speed up SIMPLEGREEDYSEARCH!
  - ➠ Reuse portions of plan not affected by dropping a variable.

UNI FREIBURG

Plan Reuse

Example, $\mathcal{W} = \{x, y\}$

Plan observing x and y:

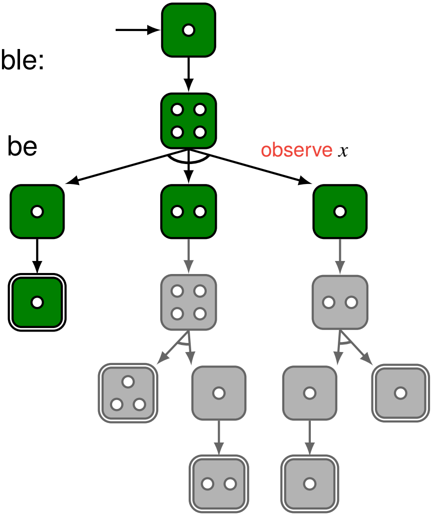


Plan Reuse

Example, $\mathcal{W} = \{x, y\}$

Reusable plan fragment if y unobservable:

- y known or
- never needs to be observed before goal

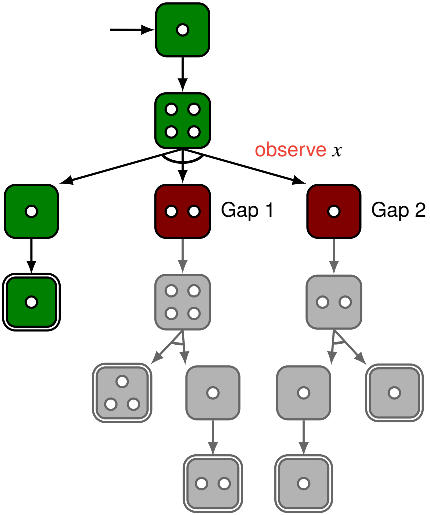


UNI FREIBURG

Plan Reuse

Example, $\mathcal{W} = \{x, y\}$

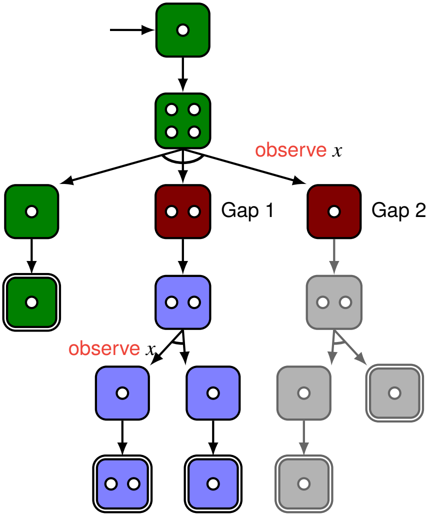
Gap states that need replanning



Plan Reuse

Example, $\mathcal{W} = \{x, y\}$

Plan to fill gap 1

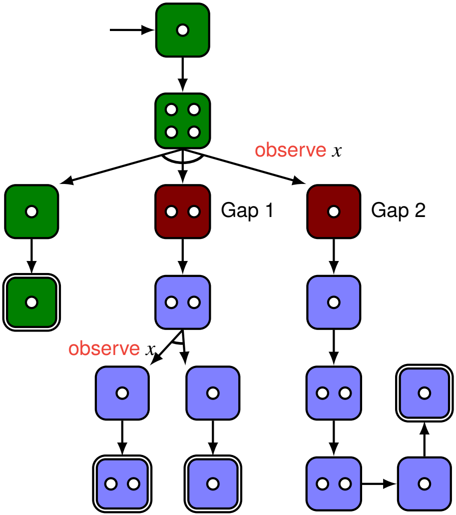


UNI FREIBURG

Plan Reuse

Example, $\mathcal{W} = \{x, y\}$

Plan to fill gap 2



UNI FREIBURG

Plan Reuse

Soundness

To eliminate variable y:

- Let π be the old plan still observing y.
- Identify gaps $1, \dots, n$ in π.
- Let π_y be the reusable fragment of π.
- Let π_j, $j = 1, \dots, n$, be the new sub-plans filling the gaps.
- Let $\pi' = \pi_y \oplus \pi_1 \oplus \cdots \oplus \pi_n$ ($\oplus =$ function overriding).

Lemma

If π and all π_j are strong cyclic plans, then so is π'.

Proof sketch.

In π', "last" subplan "wins". Thus, closedness and properness of π and all π_j carry over to π'. □

UNI FREIBURG

Plan Reuse

Soundness

To minimize set of observation variables:

- Eliminate variables one by one, if possible.

Theorem

Plan π resulting from successive elimination of variables is strong cyclic plan.

Proof sketch.

Base case + inductive application of previous lemma. □

Remarks:

- In induction, skip gaps filled/circumvented by chance when filling earlier gap.
- In elimination step, existence of π_j not guaranteed. Resulting π' not necessarily with inclusion minimal set $\mathcal{O}$.

UNI FREIBURG

Plan Reuse

Suboptimality

Remark:

Observation sets found with plan reuse can be suboptimal.

Example: Let Π with

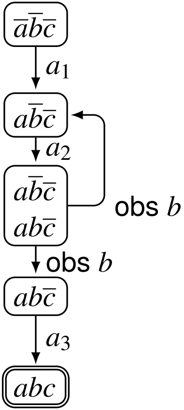
- Propositional variables $\mathcal{V} = \{a, b, c\}$,
- Initial belief state $B_0 = \overline{a}\overline{b}\overline{c}$,
- Goal belief state $B_\star = c$,
- Observable variables $\mathcal{W} = \{b\}$,
- Actions $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$, where
  - $a_1 = \langle \overline{a} \longrightarrow a \rangle$,
  - $a_2 = \langle \overline{b} \longrightarrow (b \text{ or } \top) \rangle$,
  - $a_3 = \langle b \longrightarrow c \rangle$,
  - $a_4 = \langle \overline{a}\overline{b}\overline{c} \longrightarrow c \rangle$.

UNI FREIBURG

Plan Reuse

Suboptimality

Example (ctd.): Possible plan for Π:

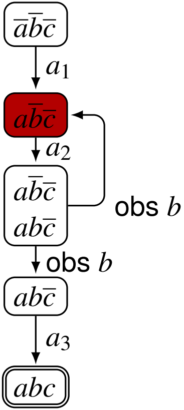


With observation of b.

Plan Reuse

Suboptimality

Example (ctd.): Possible plan for Π:



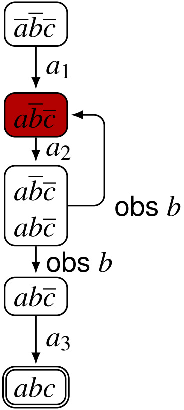
With observation of b.

Only gap state.

Plan Reuse

Suboptimality

Example (ctd.): Possible plan for Π:



With observation of b.

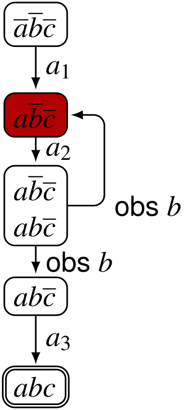
No plan for $a\bar{b}\bar{c}$ without observing b:

- only applicable action a_2 makes b unknown
- then all actions inapplicable

Plan Reuse

Suboptimality

Example (ctd.): Possible plan for Π:



With observation of b.

No plan for $a\bar{b}\bar{c}$ without observing b:

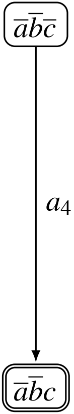
- only applicable action a_2 makes b unknown
- then all actions inapplicable

Variable b cannot be removed
$\Rightarrow$ solution $\mathcal{O} = \{b\}$

Plan Reuse

Suboptimality

Example (ctd.): Possible plan for Π:

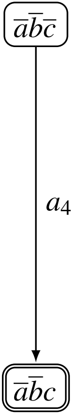


Plan for $\overline{a}\overline{b}\overline{c}$ without observation of b exists.

Plan Reuse

Suboptimality

Example (ctd.): Possible plan for Π:



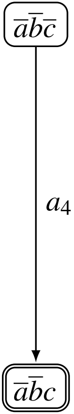
Plan for $\overline{a}\overline{b}\overline{c}$ without observation of b exists.

$\Rightarrow$ optimal solution $\mathcal{O}^* = \emptyset$

Plan Reuse

Suboptimality

Example (ctd.): Possible plan for Π:



Plan for $\overline{a}\overline{b}\overline{c}$ without observation of b exists.

$\Rightarrow$ optimal solution $\mathcal{O}^* = \emptyset$

$\Rightarrow$ solution $\mathcal{O} = \{b\}$ found with plan reuse was suboptimal

Functional Dependencies

Further enhancement: functional dependencies

- Idea: If value of variable o can be inferred from values of observed variables $o^1, \dots, o^n$, need not observe o.
- Identify such functional dependencies in plan π.
- Replace observations of o in π by observation of $o^1, \dots, o^n$.
- Remark: functional dependencies only have to hold in states reachable following π, not necessarily in all reachable states.
- Implemented: only exactly-one mutexes between propositional variables.

Empirical Evaluation

Runtimes

Motivation
Theory
Practical Algorithms
Empirical Evaluation
Conclusion

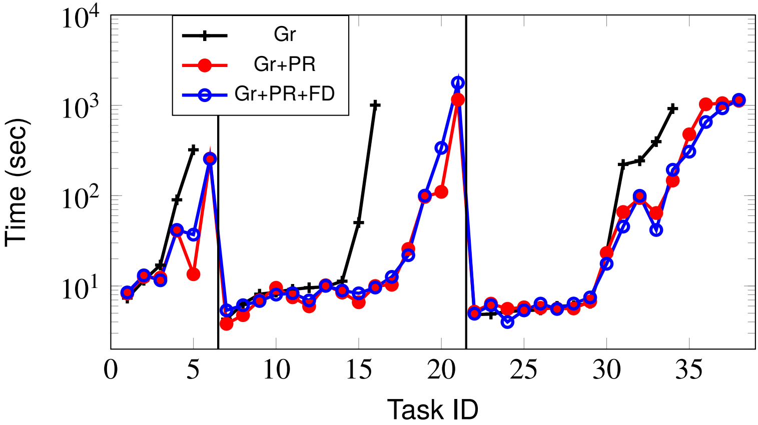
- Implementation on top of MYND planner[1].
- LAO* search [Hansen & Zilberstein, 2001] guided by FF heuristic [Hoffmann & Nebel, 2001].
- Domains:
  - BLOCKSWORLD
  - FIRSTRESPONDERS
  - TIDYUP
- Legend:
  - Gr = greedy
  - PR = plan reuse
  - FD = functional dependencies

[1] https://bitbucket.org/robertmattmueller/mynd

Empirical Evaluation

Runtimes

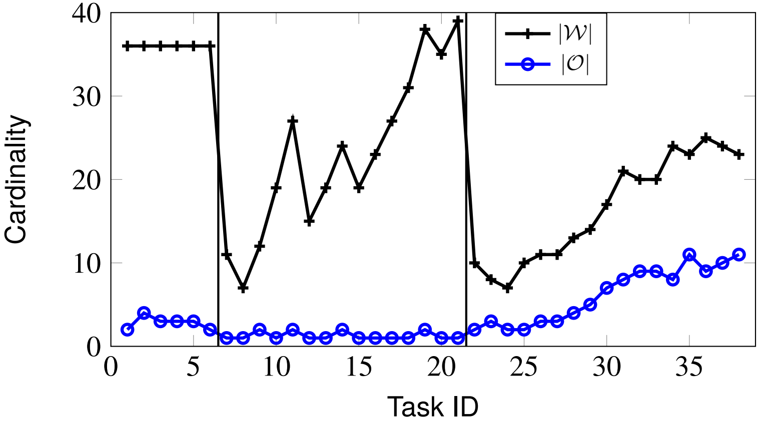
Overall runtime needed for finding final observation set.



Empirical Evaluation

Observation Set Cardinalities

Cardinalities of the observation sets before/after minimization.



Empirical Evaluation

Domain-specific Observations

Variables in resulting observation sets:

- BLOCKSWORLD: mostly
  - ONTABLE
  - CLEAR

  (either of them alone is sufficient.)
- FIRSTRESPONDERS:
  - FIRE (in all tasks, for relevant locations)
  - In one instance without road to hospital: VICTIMSTATUS – needs to be observed for applicability of TREATVICTIMONSCENE.
- TIDYUP: relevant instances of
  - GRIPPERSTATUS
  - TABLECLEAN
  - DOORSTATE
  - ROBOTLOCATION
  - CUPLOCATION

Conclusion and Future Work

Conclusion:

- Theory: OBSERVEINCLMIN is 2-EXPTIME-complete.
- Presented asymptotically optimal baseline greedy top-down algorithm for OBSERVEINCLMIN.
- Extended it with
  - plan reuse (pays off) and
  - functional dependencies (do not really pay off).

Future work:

- Complement top-down with bottom-up procedure.
- Investigate variable ordering heuristics for the iteration over candidate variables for removal.
- Study problem on domain instead of planning task level.