Using State-Based Planning Heuristics for Partial-Order Causal-Link Planning

Pascal Bercher and Thomas Geier and Susanne Biundo

Institute of Artificial Intelligence

September 19th, 2013

ulm university universität ulm

Outline

(**POCL** Planning == **P**artial-**O**rder **C**ausal-**L**ink Planning)

A planning domain is a tuple $\mathcal{D} = \langle \mathcal{V}, \mathcal{A} \rangle$ with:

- $\mathcal{V}$ is a finite set of state variables, $s \in 2^{\mathcal{V}}$ being a state,
- $\mathcal{A}$ is a finite set of actions,
  an action $a := \langle pre, add, del \rangle \in \mathcal{A}$ consists of:
  - $pre \subseteq \mathcal{V}$, the precondition of a,
  - $add \subseteq \mathcal{V}$, the add list of a,
  - $del \subseteq \mathcal{V}$, the delete list of a

A planning domain is a tuple $\mathcal{D} = \langle \mathcal{V}, \mathcal{A} \rangle$ with:

- $\mathcal{V}$ is a finite set of state variables, $s \in 2^{\mathcal{V}}$ being a state,
- $\mathcal{A}$ is a finite set of actions,
  an action $a := \langle \mathit{pre}, \mathit{add}, \mathit{del} \rangle \in \mathcal{A}$ is applicable:
  - in a state $s \in 2^{\mathcal{V}}$ iff $\mathit{pre} \subseteq s$,
  - and generates the state $(s \setminus \mathit{del}) \cup \mathit{add}$
  - (applicability of action sequences is defined as usual)

A STRIPS planning problem is a tuple $\mathcal{P} = \langle \mathcal{D}, s_{init}, g \rangle$ with:

- $\mathcal{D}$ is the planning domain,
- s_{init} is the initial state,
- g is the goal description

A solution to $\mathcal{P}$ is an action sequence $\bar{a}$, s.t.

- $\bar{a}$ is applicable in s_{init},
- $\bar{a}$ generates a state $s' \supseteq g$

A POCL planning problem is a tuple $\mathcal{P} = \langle \mathcal{D}, P_{init} \rangle$ with:

- $\mathcal{D}$ is the planning domain
- P_{init} is the initial plan (actions; partially ordered)

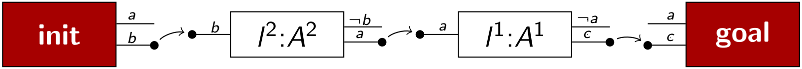
A solution to $\mathcal{P}$ is a plan P, s.t.:

- for every action sequence $\bar{a}$ induced by P holds:
  - $\bar{a}$ is applicable in s_{init},
  - $\bar{a}$ generates a state $s' \supseteq g$

A partial plan is a tuple $P = (PS, \prec, CL)$ with:

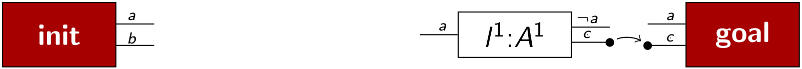
- PS is a finite set of plan steps,
  a plan step $l{:}a \in PS$ is a labeled action,
- $\prec$ is a strict partial order on PS,
- CL is a set of causal links between the plan steps in PS

Example for a partial plan:

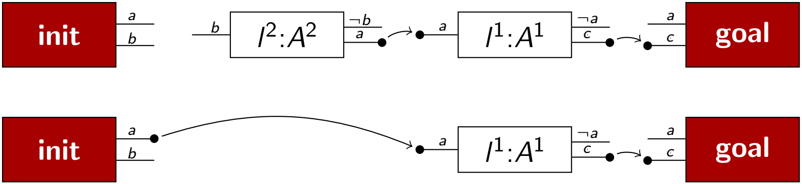


$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$

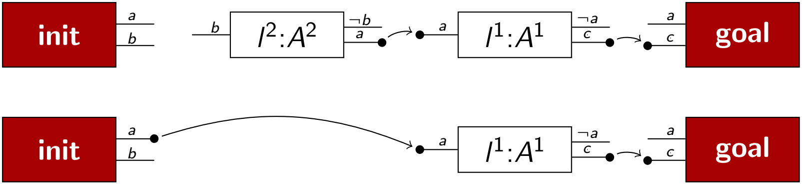
$$\mathcal{A} = \{A^1, A^2\} \text{ with: } A^1 = (a, \neg a \wedge c) \text{ and } A^2 = (b, \neg b \wedge a)$$
$$\text{initial state } s_{init} = a \wedge b, \quad \text{goal description } g = a \wedge c$$



$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$



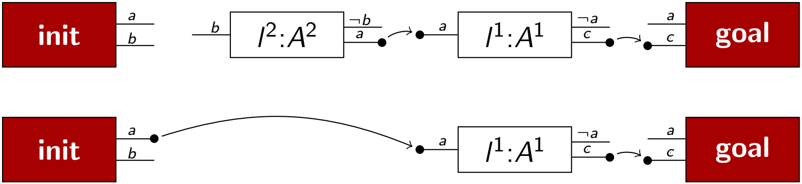
$\mathcal{A} = \{A^1, A^2\}$ with: $A^1 = (a, \neg a \wedge c)$ and $A^2 = (b, \neg b \wedge a)$
initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$



Which plan to select further?

$$\mathcal{A} = \{A^1, A^2\} \text{ with: } A^1 = (a, \neg a \wedge c) \text{ and } A^2 = (b, \neg b \wedge a)$$

initial state $s_{init} = a \wedge b$, goal description $g = a \wedge c$



Which plan to select further?

→ heuristic judges plan!

Problem:

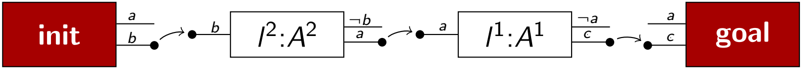
- Desired: goal-distance estimates for *plans*
- Already available: goal-distance estimates for *states*

Idea:

Encode a *plan* P by means of new planning problem $\mathcal{P}'$ s.t.:

$$\begin{array}{rcc} \text{solutions of } \mathcal{P}' & \equiv & \text{solutions reachable from } P \\ \rightarrow \text{goal distance of } P & \equiv & \text{goal distance of the initial state of } \mathcal{P}' \end{array}$$

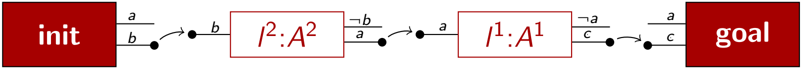
Consider the following plan P for $\mathcal{P} = \langle\langle \mathcal{V}, \mathcal{A}, \rangle, s_{init}, g\rangle$



The encoding of P is given by $\mathcal{P}' = \langle\langle \mathcal{V}', \mathcal{A}'\rangle, s'_{init}, g'\rangle$ with:

- $\mathcal{V}' := \mathcal{V} \cup \{l^1, l^2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(l^1{:}A^1), enc(l^1{:}A^1)\}$ with
  - $enc(l^1{:}A^1) = \langle a \wedge \neg l^1 \wedge l^2, \neg a \wedge c \wedge l^1\rangle$
  - $enc(l^2{:}A^2) = \langle b \wedge \neg l^2, \neg b \wedge a \wedge l^2\rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{l^1, l^2\}$

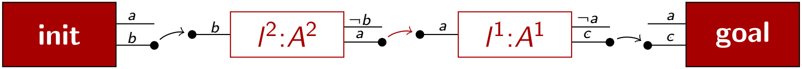
Consider the following plan P for $\mathcal{P} = \langle\langle\mathcal{V}, \mathcal{A},\rangle, s_{init}, g\rangle$



The encoding of P is given by $\mathcal{P}' = \langle\langle\mathcal{V}', \mathcal{A}'\rangle, s'_{init}, g'\rangle$ with:

- $\mathcal{V}' := \mathcal{V} \cup \{I^1, I^2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(I^1{:}A^1), enc(I^1{:}A^1)\}$ with
  - $enc(I^1{:}A^1) = \langle a \wedge \neg I^1 \wedge I^2, \neg a \wedge c \wedge I^1\rangle$
  - $enc(I^2{:}A^2) = \langle b \wedge \neg I^2, \neg b \wedge a \wedge I^2\rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{I^1, I^2\}$

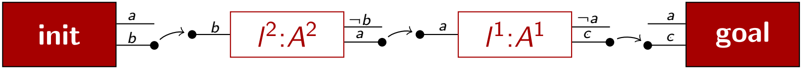
Consider the following plan P for $\mathcal{P} = \langle\langle\mathcal{V}, \mathcal{A}, \rangle, s_{init}, g\rangle$



The encoding of P is given by $\mathcal{P}' = \langle\langle\mathcal{V}', \mathcal{A}'\rangle, s'_{init}, g'\rangle$ with:

- $\mathcal{V}' := \mathcal{V} \cup \{l^1, l^2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(l^1{:}A^1), enc(l^1{:}A^1)\}$ with
  - $enc(l^1{:}A^1) = \langle a \wedge \neg l^1 \wedge l^2, \neg a \wedge c \wedge l^1\rangle$
  - $enc(l^2{:}A^2) = \langle b \wedge \neg l^2, \neg b \wedge a \wedge l^2\rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{l^1, l^2\}$

Consider the following plan P for $\mathcal{P} = \langle\langle \mathcal{V}, \mathcal{A}, \rangle, s_{init}, g\rangle$



The encoding of P is given by $\mathcal{P}' = \langle\langle \mathcal{V}', \mathcal{A}'\rangle, s'_{init}, g'\rangle$ with:

- $\mathcal{V}' := \mathcal{V} \cup \{l^1, l^2\}$
- $\mathcal{A}' := \mathcal{A} \cup \{enc(l^1{:}A^1), enc(l^1{:}A^1)\}$ with
  - $enc(l^1{:}A^1) = \langle a \wedge \neg l^1 \wedge l^2, \neg a \wedge c \wedge l^1\rangle$
  - $enc(l^2{:}A^2) = \langle b \wedge \neg l^2, \neg b \wedge a \wedge l^2\rangle$
- $s'_{\text{init}} := s_{\text{init}}$
- $g' := g \cup \{l^1, l^2\}$

Computational complexity:

- a plan $P = (PS, \prec, \emptyset)$ can be encoded in $O(|\prec|) = O(|PS|^2)$
- a plan $P = (PS, \prec, CL)$, $CL \neq \emptyset$ can be be encoded in **P**
- incremental encoding can be done in $O(|P|)$ and $\Omega(1)$ if $P = (PS, \prec, \emptyset)$ (cf. KEPS 2013)

Admissibility:

- if the used state-based heuristic is admissible, so is the "new" one!

We evaluated (almost) all problems from the IPC 1 to IPC 5.

We used our POCL planner PANDA with weighted A* and:

- The **Add Heuristic** for POCL Planning (Younes & Simmons)
- The **Relax Heuristic** (Nguyen & Kambhampati)
- **Transformation + Lm-Cut** (Helmert & Domshlak)
- **Transformation + Merge & Shrink** (Helmert et al.)

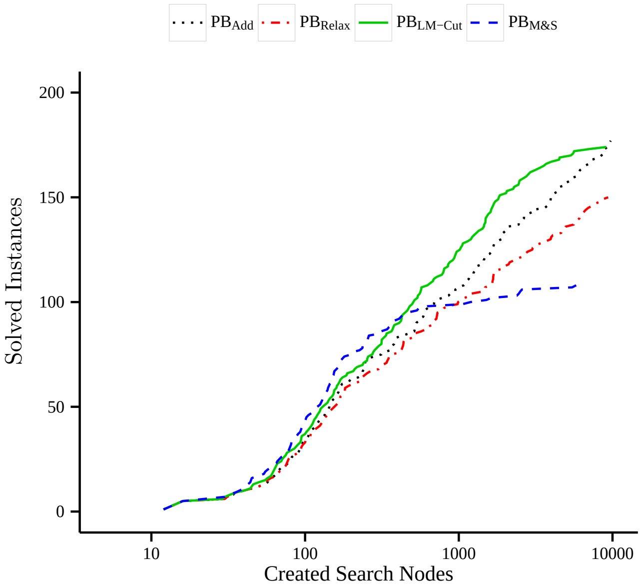
Heuristic calculation:

- We generate a PDDL domain and problem file for each plan P
- Fast Downward calculates heuristic h of the initial state
- PANDA uses h - cost(P) as heuristic estimate

We measured:

- number of solved problem instances per domain.
- search space size when a solution was found.

Results show many timeouts (due to file communication)
But: the state-based heuristics are quite informed if not timed-out!


PB$_{Add}$
PB$_{Relax}$
PB$_{LM-Cut}$
PB$_{M\&S}$
Solved Instances
200
150
100
50
0
10
100
1000
10000
Created Search Nodes

Summary:

- Encoding can be done in **P**, even with causal links
- Encoding provides the *first admissible heuristics* for POCL planning
- Lm-cut seems to work well with our encoding

Outlook:

- Native implementation of Lm-cut for the encoding
- Native *adaptation* of Lm-cut for POCL planning

Domain	n	PB_{Add}	PB_{Relax}	$PB_{LM\text{-}Cut}$	$PB_{M\&S}$	$SB_{LM\text{-}Cut}$	$SB_{M\&S}$
grid	5	**0**	**0**	**0**	**0**	2	2
gripper	20	14	**20**	1	1	20	8
logistics	20	**16**	15	6	0	16	1
movie	30	**30**	**30**	**30**	**30**	30	30
mystery	20	8	**10**	5	5	13	13
mystery-prime	20	3	**4**	2	1	12	12
blocks	21	2	3	3	**5**	21	21
logistics	28	**28**	**28**	27	5	28	15
miconic	100	**100**	53	65	29	100	68
depot	22	**2**	**2**	1	1	11	7
driverlog	20	7	**9**	3	3	15	12
freecell	20	**0**	**0**	**0**	**0**	6	6
rover	20	**20**	19	9	5	18	8
zeno-travel	20	4	4	3	**5**	16	13
airport	20	**18**	15	6	5	20	18
pipesworld-noTankage	20	**8**	5	1	1	18	19
pipesworld-Tankage	20	**1**	**1**	**1**	**1**	11	14
satellite	20	**18**	**18**	4	3	15	7
pipesworld	20	**1**	**1**	**1**	**1**	11	14
rover	20	**0**	**0**	**0**	**0**	18	8
storage	20	7	**9**	5	5	17	15
tpp	20	**9**	8	5	5	9	7
total	526	296	254	178	111	427	318

Let $\mathcal{P} = \langle\langle\mathcal{V}, \mathcal{A}\rangle, s_{init}, g\rangle$ be a planning problem and $P = (PS, \prec, CL)$ a plan.

The encoding of P is given by $\mathcal{P}' = \langle\langle\mathcal{V}', \mathcal{A}'\rangle, s'_{init}, g'\rangle$ with:

- $\mathcal{A}' := \mathcal{A} \cup \mathcal{A}_{new}$, with $\mathcal{A}_{new} := \{\ enc(l{:}A) \mid l{:}A \in PS\ \}$
- $\mathcal{A}_{new}$ encodes the plan steps in PS, s.t.:
  - each $a \in \mathcal{A}_{new}$ is executable exactly once
  - the actions in $\mathcal{A}_{new}$ can only be inserted in an order consistent with the one in PS
  - no action in $\mathcal{A}'$ can violate causal links in CL
- The goal description is altered, s.t. all actions in $\mathcal{A}_{new}$ have to be executed