Sound and Complete Landmarks for AND/OR Graphs

Emil Keyder[1], Silvia Richter[2], and Malte Helmert[3]

August 18th, 2010

[1]Universitat Pompeu Fabra, Spain
[2]Griffith University and NICTA, Australia
[3]Albert-Ludwigs-Universität Freiburg, Germany

What are landmarks?

Features common to *every solution* of a planning task:

- *Formulas over fluents* satisfied in some state
- *Actions* that appear in every plan

Two primary questions

1. How do we find them?
2. How do we use them?

What are landmarks?

Features common to *every solution* of a planning task:

- *Formulas over fluents* satisfied in some state
- *Actions* that appear in every plan

Two primary questions

1. How do we find them?
2. How do we use them?

This work focuses on finding landmarks.

Contributions

- *Declarative characterization* of delete-relaxation landmarks in terms of underlying AND/OR graphs
- Computation of *complete set of causal landmarks* for the delete relaxation
- Parametrized method analogous to the h^m heuristics that computes for $m > 1$:
  - Conjunctive landmarks of size m
  - Non-delete-relaxation landmarks

Planning tasks

We use the STRIPS formalism:

Definition (planning task)

A *planning task* is a 4-tuple $\Pi = \langle F, A, I, G \rangle$:

- F a finite set of fluents
- A a finite set of actions, each with:
  - $pre(a) \subseteq F$ a set of preconditions
  - $add(a) \subseteq F$ a set of add effects
  - $del(a) \subseteq F$ a set of delete effects
  - $cost(a) \in \mathbb{R}_0^+$ a cost
- $I \subseteq F$ the set of fluents true in the initial state
- $G \subseteq F$ the set of goal fluents

Landmarks

Definition (landmark)

A *landmark* is a logical formula λ over F such that every valid plan π makes λ true in some state during its execution.

- **Fact landmark** $|\lambda| = 1$
- **Disjunctive landmark** A disjunction over F
- **Conjunctive landmark** A conjunction over F

Definition (action landmark)

An *action landmark* is an action a that appears in every valid plan π.

Finding landmarks is difficult

Determining if a fact is a landmark is PSPACE-complete in general, but in P for problems with no deletes

Approach: Look for landmarks of the *delete relaxation* Π^+

Use *backchaining* (Porteous, Sebastia & Hoffmann, 2001, Hoffmann, Porteous & Sebastia, 2004):

1. Start from a fact p known to be a landmark
2. Check whether all a adding p have a common precondition q
3. If so, add q to set of landmarks and repeat

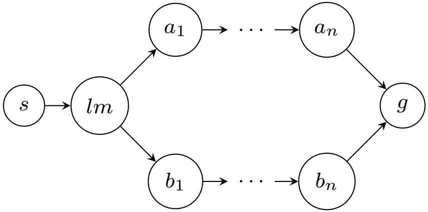
Extensions: Consider disjunctions and use them to discover additional landmarks (lookback), or use them directly (Richter, Helmert & Westphal, 2008)

Backchaining: drawbacks

Does not compute complete set of Π^+ landmarks

Requires arbitrary limits on size of disjunctions, lookback depth

For any n, we can construct a problem that requires n levels of lookback:



How to find the complete set?

Checking whether a fact is a landmark in Π^+ is in P, therefore so is finding the complete set of landmarks

The naive approach is not efficient

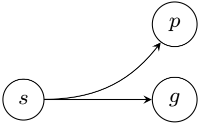
Zhu & Givan (2003) propose an efficient forward propagation approach that finds the complete set of *causal landmarks*

Causal Π^+ landmarks

Definition (causal landmark)

A *causal landmark* is a formula λ over F such that either λ is entailed by the goal, or every valid plan π contains an action a whose precondition *pre*(a) entails λ.

Example: p is a non-causal landmark



Forward propagation

Zhu & Givan's algorithm propagates *labels* representing landmark sets *forwards* in the relaxed planning graph

At level i:

1. Mark each action node with the *union* of the labels for its preconditions at level $i - 1$
2. Mark each fluent node with the *intersection* of the labels for the actions that add it at level $i - 1$

Intuition: Fact p is a landmark for action a if it is a landmark for *some* $q \in$ *pre*(a).
Fact p is a landmark for fact q if it is a landmark for *all* actions that achieve q.

AND/OR graphs

Π^+ defines an AND/OR graph:

Definition (AND/OR graph)

An *AND/OR graph* $\langle V_\mathsf{I}, V_\mathsf{AND}, V_\mathsf{OR}, E\rangle$ is a directed graph with

- vertices $V_\mathsf{I} \cup V_\mathsf{AND} \cup V_\mathsf{OR}$ and edges E, where
- V_I, V_AND and V_OR are disjoint sets called initial nodes, AND nodes, and OR nodes, respectively.

Intuition:

- initial nodes: things known to be true (initial state)
- AND nodes: true iff all predecessors true (actions)
- OR nodes: true iff some predecessor true (fluents)

AND/OR landmarks

Justifications in AND/OR graphs correspond to a plans in Π^+:

Definition (AND/OR graph justification)

Given an AND/OR graph $\mathcal{G} = \langle V_\mathsf{I}, V_\mathsf{AND}, V_\mathsf{OR}, E \rangle$, a *justification* for a set of nodes V_G is a subgraph $J = \langle V^J, E^J \rangle$ of $\mathcal{G}$ such that:

- $V_\mathsf{G} \subseteq V^J$
- For every AND node $v \in V^J$, *pred*$(v) \subseteq V^J$.
- For every OR node $v \in V^J$, $\exists u(u \in$ *pred*$(v) \wedge u \in V^J)$
- J is acyclic.

Definition (AND/OR landmark)

Given an AND/OR graph $\mathcal{G} = \langle V_\mathsf{I}, V_\mathsf{AND}, V_\mathsf{OR}, E \rangle$, a node n is a landmark for a node set V_G if $n \in V^J$ for all justifications J for V_G.

Landmark equations for AND/OR graphs

The fixpoint of the Zhu and Givan algorithm is the solution to the following set of equations for the AND/OR graph defined by Π^+:

$$\begin{aligned} \mathit{LM}(v) &= \{v\} && \text{if } v \in V_{\mathsf{I}} \\ \mathit{LM}(v) &= \{v\} \cup \bigcap_{u \in \mathit{pred}(v)} \mathit{LM}(u) && \text{if } v \in V_{\mathsf{OR}} \\ \mathit{LM}(v) &= \{v\} \cup \bigcup_{u \in \mathit{pred}(v)} \mathit{LM}(u) && \text{if } v \in V_{\mathsf{AND}} \end{aligned}$$

The landmarks for a set of nodes are given by the union of the landmarks for each of the nodes:

$$\mathit{LM}(V) = \bigcup_{v \in V} \mathit{LM}(v)$$

Discussion

Theorem (*LM* equations solution)

The LM(·) equations have a unique maximal solution with regard to set inclusion.

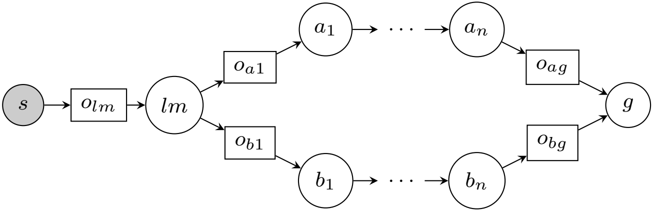
Computation: Initialize landmark sets as follows:

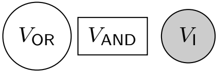
$$LM(v) = \{v\} \qquad \text{if } v \in V_\mathsf{I}$$
$$LM(v) = V_\mathsf{I} \cup V_\mathsf{AND} \cup V_\mathsf{OR} \qquad \text{otherwise}$$

and apply equations as update rules until fixpoint.

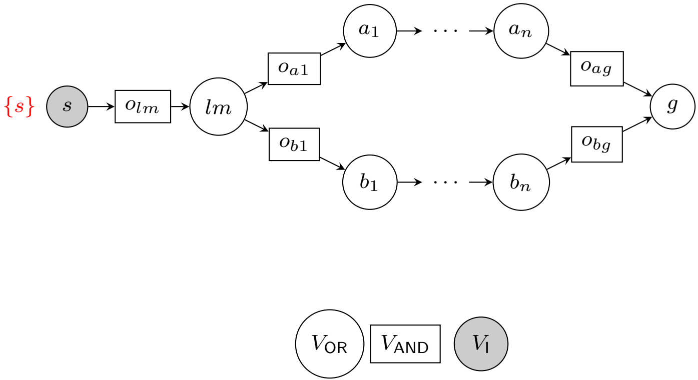
Second initialization rule ensures that maximal solution is found.

Propagation example

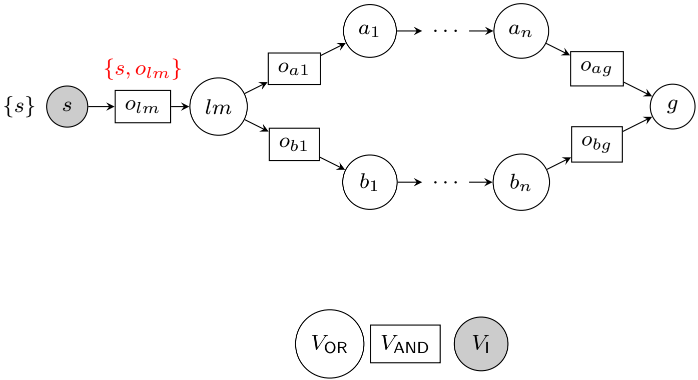




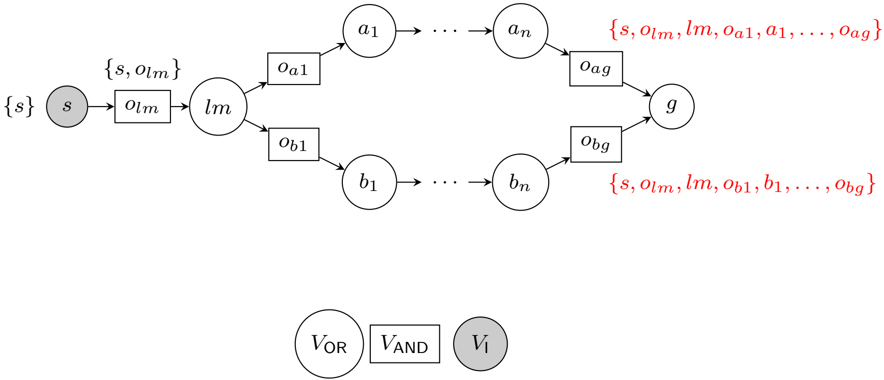
Propagation example



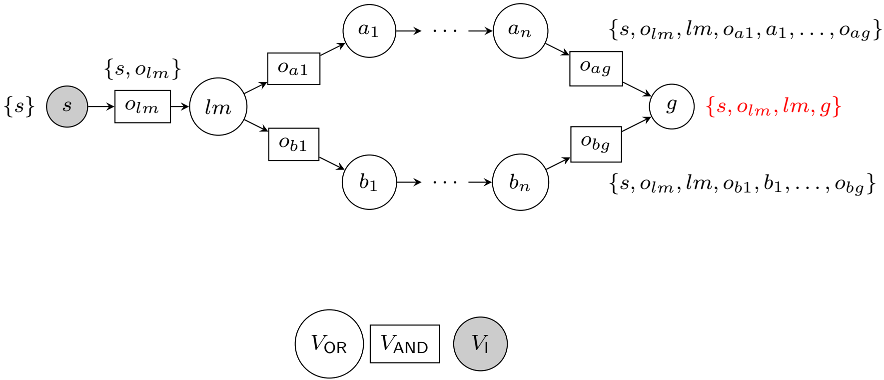
Propagation example



Propagation example



Propagation example



Landmarks beyond Π^+

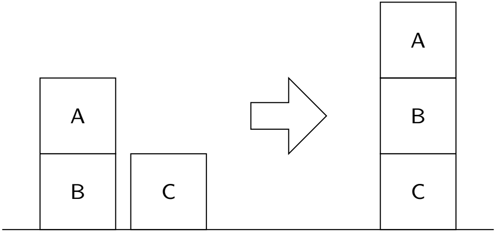
Observation: The equations are applicable to *any* planning problem with no delete effects

The Π^m compilation (Haslum, 2009) yields such a planning problem that encodes information about deletes in Π:

- Fluents of Π^m are subsets of F of size $\leq m$
- Π^m fluents can not be achieved by actions deleting some Π fluent in the set

Landmarks of Π^m are *conjunctive landmarks* in Π that take into account *delete information*

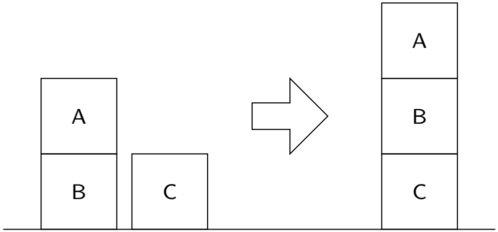
Blocksworld example



Π^+ Landmarks:

$$\textit{clear } B \prec_{\text{gn}} \textit{holding } B$$

Blocksworld example



(Some) Π^2 Landmarks:

$$(\mathit{clear}\ B \land \mathit{holding}\ A) \prec_{\mathrm{gn}} (\mathit{clear}\ B \land \mathit{ontable}\ A) \prec_{\mathrm{gn}}$$
$$(\mathit{holding}\ B \land \mathit{ontable}\ A) \prec_{\mathrm{gn}} (\mathit{on}\ B\ C \land \mathit{ontable}\ A) \prec_{\mathrm{gn}}$$
$$(\mathit{on}\ B\ C \land \mathit{holding}\ A)$$

We look at *optimal* planning

Planning framework: Admissible landmark cost-partitioning heuristic with LM-A^* search (Karpas & Domshlak, 2009)

Questions:

1. Do we find more causal landmarks?
2. Do these landmarks improve heuristic accuracy?
3. Do we improve planning performance?

We compare to Richter, Helmert & Westphal, 2008 (RHW)

Do we find more landmarks?

	RHW	$m = 1$	$m = 2$	$m = 2$ **conj**
elevators-opt	629 (30)	1.12	1.12	3.66
openstacks-opt	2925 (30)	1.03	1.03	11.37
parcprinter	2142 (30)	1.00	1.07	18.37
pegsol	1457 (30)	1.00	1.02	19.33
scanalyzer	673 (26)	1.00	1.26	9.65
sokoban-opt	605 (29)	2.73	4.80	36.95
transport-opt	390 (30)	1.00	1.00	3.44
woodworking-opt	1520 (30)	1.06	1.08	9.91

Table: Number of landmarks generated in IPC6 domains.

Yes! Up to 2.73 as many Π^+ landmarks, 4.80 as many fact landmarks with $m = 2$, and 36.95 as many landmarks when conjunctive landmarks are also counted.

Are they informative?

	RHW	$m = 1$	$m = 2$	$m = 2$ **conj**
elevators-opt	483982 (7)	1.00	1.00	1.35
openstacks-opt	649341 (10)	1.00	1.00	1.00
parcprinter	1118898 (12)	1.00	1.29	1.61
pegsol	1734655 (23)	1.00	1.04	1.20
scanalyzer	23029 (11)	1.00	1.00	1.46
sokoban-opt	3502115 (13)	1.02	1.03	1.10
transport-opt	929285 (9)	1.00	1.00	1.00
woodworking-opt	199666 (10)	1.41	1.41	2.35

Table: Nodes expanded to find a solution in IPC6 domains.

Yes! With optimal cost-partitioning, up to 2.10 times fewer nodes expanded when using fact landmarks, and up to 22.80 times fewer nodes expanded when using conjunctive landmarks with $m = 2$.

Do they improve overall performance?

Contrary to expectations, no.

Coverage with complete Π^+ landmarks is slightly better (584 vs. 576 out of 1116)

Using Π^m landmarks improves coverage in some domains, but decreases coverage in general:

- Computation of Π^m landmarks is costly
- Greater informativeness all but disappears with uniform cost partitioning, optimal cost partitioning is expensive
- More landmarks lead to greater overhead, slower heuristic computation

Conclusions & future work

Declarative characterisation gives us a better understanding of Π^+ landmarks

Separates *what* is computed from *how* it is computed

With Π^m, we obtain *conjunctive* landmarks and non-Π^+ landmarks

Future work:

- Cost-partitioning schemes intermediate between uniform and optimal
- Make Π^m compilation more efficient

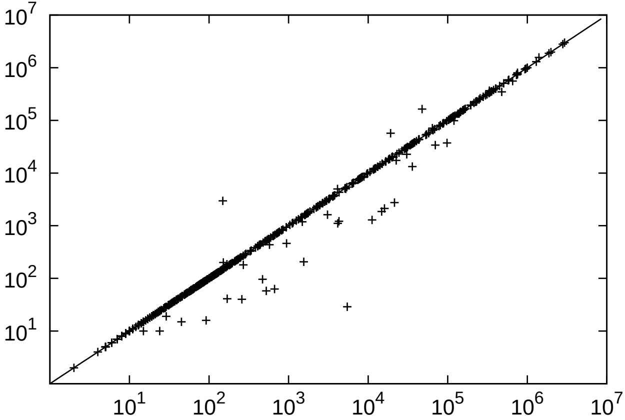
The Π^m compilation

Definition (Π^m problem)

For a planning problem $\Pi = \langle F, A, I, G \rangle$ and a value $m \geq 1$, $\Pi^m = \langle F^m, A^m, I^m, G^m \rangle$ is given by

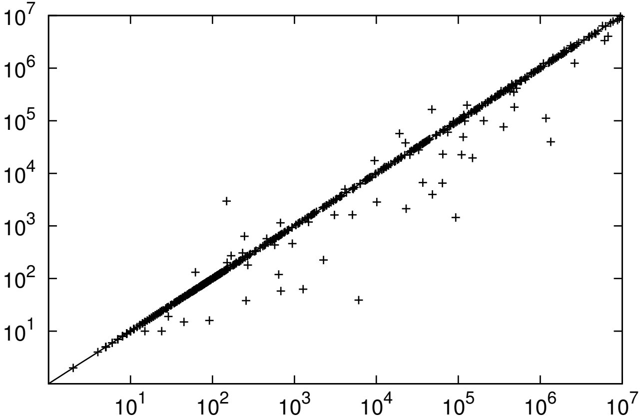
$$
\begin{array}{rl}
F^m & \{f \subseteq F \mid |f| \leq m\} \\
A^m & \bigcup_{a \in A} \bigcup_{c \in F^{m-1}} \{a_c \mid c \cap add(a) = c \cap del(a) = \emptyset\} \\
\textit{pre}(a_c) & \{f \in F \mid f \subseteq (pre(a) \cup c)\} \\
\textit{add}(a_c) & \{f \in F \mid f \subseteq (add(a) \cup c) \wedge f \cap add(a) \neq \emptyset\} \\
\textit{del}(a_c) & \emptyset \\
\textit{cost}(a_c) & \textit{cost}(a) \\
I^m & \{i \subseteq I \mid |i| \leq m\} \\
G^m & \{g \subseteq G \mid |g| \leq m\}
\end{array}
$$

Landmark comparison: Π^+ vs. Richter, Helmert & Westphal



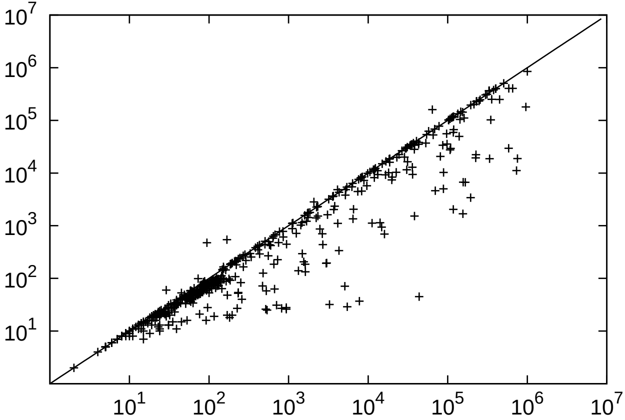
Expanded nodes (optimal cost partitioning):
Π^+ landmarks (y-axis) vs. RHW landmarks (x-axis)

Landmark comparison: Π^+ vs. Richter, Helmert & Westphal



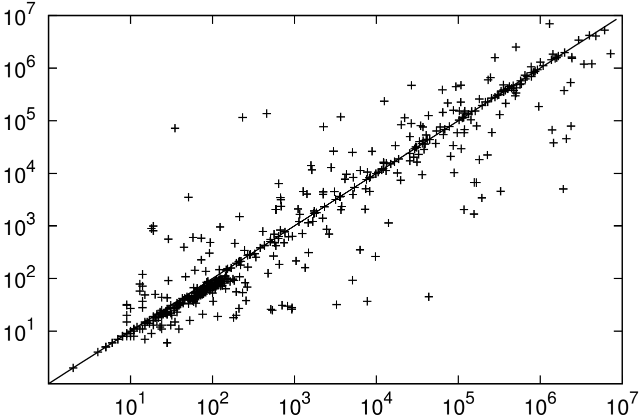
Expanded nodes (uniform cost partitioning):
Π^+ landmarks (y-axis) vs. RHW landmarks (x-axis)

Landmark comparison: Π^2 vs. Richter, Helmert & Westphal



Expanded nodes (optimal cost partitioning):
Π^2 landmarks (y-axis) vs. RHW landmarks (x-axis)

Landmark comparison: Π^2 vs. Richter, Helmert & Westphal



Expanded nodes (uniform cost partitioning):
Π^2 landmarks (y-axis) vs. RHW landmarks (x-axis)