ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010
© 2010 The authors and IOS Press. All rights reserved.
doi:10.3233/978-1-60750-606-5-335

Sound and Complete Landmarks for And/Or Graphs

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

Abstract. Landmarks for a planning problem are subgoals that are necessarily made true at some point in the execution of any plan. Since verifying that a fact is a landmark is PSPACE-complete, earlier approaches have focused on finding landmarks for the delete relaxation Π^+. Furthermore, some of these approaches have *approximated* this set of landmarks, although it has been shown that the complete set of *causal delete-relaxation landmarks* can be identified in polynomial time by a simple procedure over the relaxed planning graph. Here, we give a declarative characterisation of this set of landmarks and show that the procedure computes the landmarks described by our characterisation. Building on this, we observe that the procedure can be applied to any delete-relaxation problem and take advantage of a recent compilation of the m-relaxation of a problem into a problem with no delete effects to extract landmarks that take into account delete effects in the original problem. We demonstrate that this approach finds strictly more causal landmarks than previous approaches and discuss the relationship between increased computational effort and experimental performance, using these landmarks in a recently proposed admissible landmark-counting heuristic.

1 INTRODUCTION

Landmarks in the context of planning are propositions, or more generally formulas over propositions, that are necessarily made true in some state during the execution of any plan. Landmark-based approaches to planning have recently enjoyed great success, with the winner of the most recent International Planning Competition (IPC6) employing a landmark-counting heuristic [11].

Planning techniques based on landmarks can be characterised in terms of two orthogonal properties: *landmark utilisation*, the methods used to take advantage of the knowledge that a given formula is a landmark, and *landmark generation*, the methods used to generate the set of landmarks. Earlier approaches to landmark utilisation focused on using landmark information to provide a control loop, feeding to a classical planner the next landmark to be achieved in a given order as an intermediate goal [5]. More recent techniques have used landmarks to generate heuristic functions for planning problems. In the satisficing setting this has taken the form of *landmark-counting* heuristics that count the number of landmarks that remain to be achieved or need to be reachieved [10]. In the optimal setting, two recent state-of-the-art admissible heuristics take advantage of landmark information. One of these is a cost-partitioning approach in which the minimal cost of achieving each landmark is summed over the set of landmarks of the problem [6]. This heuristic is complemented with an improved version of the optimal search algorithm A*, LM-A*, which checks whether the landmarks achieved along some path to a given state are necessarily achieved along *all* paths to that state, and uses this information to boost the set of landmarks that remain to be achieved and thus the heuristic estimate for the state, while maintaining admissibility. The second heuristic uses the delete-relaxation landmarks of a problem to closely approximate its optimal delete-relaxation cost [4].

The principal contribution of this paper is in the area of landmark generation. Since checking whether a given fact is a landmark for a problem is PSPACE-complete, approaches to landmark generation have generally concentrated on finding landmarks for the delete relaxation Π^+ of the planning problem, and provided no guarantees on the completeness of the set of landmarks that is found [5, 10]. However, one method has been proposed that guarantees completeness according to the well-defined criterion of *causality* [12]. Here, we give a set of equations whose solution describes the landmarks computed by this method, and show that the equations can be applied to the Π^m compilation of a planning problem [2] to obtain, for the first time, both *conjunctive landmarks* and *landmarks beyond the delete relaxation*. The method used is polynomial in the size of the compiled problem, which grows exponentially in m. Furthermore, for sufficiently large m the landmarks computed are the complete set of causal landmarks for Π.

2 PRELIMINARIES

STRIPS planning. We use the propositional STRIPS formalism augmented with non-negative actions costs (e. g., [7]).

Definition 1 (planning task)
A planning task *is a 4-tuple* $\Pi = \langle F, A, I, G\rangle$, *where*

- F *is a finite set of* propositional state variables*,*
- A *is a finite set of* actions*, each with associated* preconditions *pre*$(a) \subseteq F$*,* add effects *add*$(a) \subseteq F$*,* delete effects *del*$(a) \subseteq F$ *and* cost *cost*$(a) \in \mathbb{R}_0^+$*,*
- $I \subseteq F$ *is the* initial state*, and*
- $G \subseteq F$ *is the set of* goals.

State variables of planning tasks are also called *propositions* or *facts*. A *state* in our formalism is a subset of facts, representing the propositions which are currently true. States can alternatively be defined as assignments to state variables, but set notation is more convenient for the purposes of this paper. *Applying* an action a in s results in state $(s \setminus del(a)) \cup add(a)$, which we denote as $s[a]$. The notation is only defined if a is *applicable* in s, i. e., if $pre(a) \subseteq s$.

Applying an action sequence $a_1, \ldots, a_n$ to a state is defined inductively as $s[\epsilon] := s$ and $s[a_1, \ldots, a_{i+1}] := (s[a_1, \ldots, a_i])[a_{i+1}]$.

[1] Universitat Pompeu Fabra, Roc Boronat, 138, 08018 Barcelona, Spain, email: emil.keyder@upf.edu
[2] Griffith University and NICTA, Lvl 5, Axon (47), Staff House Road, St. Lucia, QLD 4072, Australia, email: silvia.richter@nicta.com.au
[3] Albert-Ludwigs-Universität Freiburg, Georges-Köhler-Allee 52, 79110 Freiburg, Germany, email: helmert@informatik.uni-freiburg.de

A plan for a state s (*s-plan*, or *plan* when s is clear from context) is an action sequence π such that $s[\pi]$ is defined and satisfies all goals (i. e., $G \subseteq s[\pi]$). The *cost* of plan $\pi = a_1, \ldots, a_n$ is $\mathit{cost}(\pi) := \sum_{i=1}^{n} \mathit{cost}(a_i)$. The objective of *optimal planning* is to find an I-plan of minimal cost (called an *optimal I-plan*) or prove that no plan exists.

Landmarks. A *landmark* is a logical formula L (possibly consisting of a single fact) over the set F such that for any I-plan $a_1, \ldots, a_n$ there exists a prefix $a_1, \ldots, a_i$ such that $s[a_1, \ldots, a_i] \models L$. An *action landmark* is an action a such that $a \in \pi$ for any I-plan π. Given two landmarks L_1 and L_2, there is a *natural ordering* $L_1 \prec_{\text{n}} L_2$ if for any I-plan $a_1, \ldots, a_n$, $s[a_1, \ldots, a_j] \models L_2$ implies that there exists $i < j$ such that $s[a_1, \ldots, a_i] \models L_1$. There is a *greedy-necessary* ordering $L_1 \prec_{\text{gn}} L_2$ if for any I-plan $a_1, \ldots, a_n$, $s[a_1, \ldots, a_j] \models L_2$ and $s[a_1, \ldots, a_i] \not\models L_2$ for all $i < j$ implies $s[a_1, \ldots, a_{j-1}] \models L_1$.

3 DELETE RELAXATION LANDMARKS

Given a planning task $\Pi = \langle F, A, I, G \rangle$, the delete relaxation $\Pi^+ = \langle F, A^+, I, G \rangle$ is obtained by removing from each action its set of delete effects. Formally, the modified action set A^+ of Π^+ is given by $A^+ = \{a^+ \mid a \in A\}$, where $\mathit{pre}(a^+) = \mathit{pre}(a)$, $\mathit{add}(a^+) = \mathit{add}(a)$, $\mathit{del}(a^+) = \emptyset$ and $\mathit{cost}(a^+) = \mathit{cost}(a)$.

The delete relaxation is a fundamental structure in recent approaches to planning, its main use being the extraction of a *relaxed plan* whose cost can be used as a heuristic for the original problem. The attraction of the delete relaxation stems from the fact that while finding the optimal relaxed plan is NP-hard, finding *some* plan is in P, and many good approximate approaches have been proposed. One tool used to perform various computations on the Π^+ problem is the *relaxed planning graph* (RPG), which represents facts and actions in alternating layers. The first layer of an RPG consists of the initial set of facts I, while subsequent layers are constructed based on two rules: an action a appears in layer i if all facts $f \in \mathit{pre}(a)$ are present in layer $i-1$, and a fact f appears in layer j if f is also present in layer $j-2$ (via a no-op action) or if $f \in \mathit{add}(a)$ for some a in layer $j-1$. These rules are applied until no new facts can be added.

In addition to the computation of heuristics, the RPG representation of the delete relaxation has also been widely used for the extraction of landmarks. Most work has focused on methods based on *backchaining*, beginning with a fact g known to be a landmark (e. g. a goal of the problem) and discovering further landmarks by analysing the actions A_g at the previous level that add g. One approach is to take the intersection $\bigcap_{a \in A_g} \mathit{pre}(a)$ of all precondition sets of actions in A_g, since if all actions adding g require f as a precondition, then f is also a landmark [5]. Since this method typically does not find many landmarks, the algorithm can be enhanced with a *lookahead* procedure which works as follows: first, a temporary *disjunctive* landmark is built by selecting, from each action in A_g, one of its preconditions and creating a disjunction over these facts. Since one of the actions in A_g must be applied, one of the facts in this disjunction will have to be made true. Next, the approach checks all actions that add any of the facts in the disjunction. If these actions share a precondition f', then f' is a landmark. Alternatively, the algorithm can simply be extended to handle disjunctive landmarks directly, rather than only as temporaries [10]. However, these techniques have their drawbacks. For any n-step lookahead procedure a problem can be designed such that a landmark appears $n+1$ steps before the known landmark we are backchaining from, while if disjunctive landmarks are admitted, an arbitrary upper limit on the size of disjunctions or some other restriction must be specified in order to avoid encoding all possible plans for the problem in the form of disjunctions.

However, there exists a simple algorithm due to Zhu & Givan [12] that, rather than applying a backchaining criterion recursively, computes landmarks via *forward* propagation in the RPG. This algorithm is sound and complete according to the simple and intuitive criterion of *causality*, which excludes "incidentally" achieved facts that are added by some action in the plan, but not necessarily used as preconditions by some other action:

Definition 2 (Causal Landmarks)
A fact f is a causal (fact) landmark *for a problem Π if it is a goal of Π or if for all valid plans π for Π, $f \in \mathit{pre}(a)$ for some $a \in \pi$.*

The algorithm works by associating with each action or fact node at every level of the RPG a *label* consisting of the set of facts that must be made true in order to reach it. In the first level of the RPG, each initial state fact is associated with a label containing only itself. The labels of the nodes appearing at following levels are obtained by combining the labels of the nodes in previous layers in two different ways:

- The label for an action node a at level i is the union of the labels of all its preconditions at level $i-1$.
- The label for a fact node f at level i is the intersection of the labels of all action nodes adding it at level $i-1$ (possibly including no-op actions), plus the fact itself.

Intuitively, these rules state that for a fact f to be a landmark for an action a, it is sufficient that f be a landmark for *some* precondition of a, and that for a fact f to be a landmark for another fact f' at a given level, either $f = f'$ or f must be a landmark for *all* action nodes that can achieve f' at that level.

Given these propagation rules, the label associated with a fact or action node at any level i is a *superset* of the set of causal landmarks for this fact or action in Π^+. If the RPG construction continues until a fixpoint is reached, i. e. until no further changes occur in the node labels from layer to layer, the landmarks for the goal nodes in the last layer are exactly the causal landmarks for Π^+ [12].

4 AND/OR LANDMARKS

In order to give a more general declarative characterisation of the landmarks computed above, we first discuss AND/OR graphs and how the delete relaxation can be understood as an instance of this type of graph. For a fuller treatment of the subject, see the paper by Mirkis and Domshlak [9].

An AND/OR graph $\mathcal{G} = \langle V_{\text{I}}, V_{\text{and}}, V_{\text{or}}, E \rangle$ is a directed graph with vertices $V := V_{\text{I}} \cup V_{\text{and}} \cup V_{\text{or}}$ and edges E, where V_{I}, V_{and} and V_{or} are disjoint sets called *initial nodes*, *AND nodes* and *OR nodes*, respectively. A subgraph $J = \langle V^J, E^J \rangle$ of $\mathcal{G}$ is said to *justify* $V_{\text{G}} \subseteq V$ if and only if the following are true of J:

1. $V_{\text{G}} \subseteq V^J$
2. $\forall a \in V^J \cap V_{\text{and}} : \forall \langle v, a \rangle \in E : v \in V^J \wedge \langle v, a \rangle \in E^J$
3. $\forall o \in V^J \cap V_{\text{or}} : \exists \langle v, o \rangle \in E : v \in V^J \wedge \langle v, o \rangle \in E^J$
4. J is acyclic.

Intuitively, J is a justification for V_{G} if J contains a "proof" that all nodes in V_{G} are "true" under the assumption that all nodes in V_{I} are true. The set V^J represents the nodes that are proven to be true by J, and the edges E^J represent the arguments for why they are true. The four conditions then state that (1) all nodes in V_{G} must be proven true, (2) AND nodes are proven true if all their predecessors are true, (3) OR nodes are proven true if they have some true predecessor, and (4) the proof must be well-founded.

The delete relaxation can be understood as specifying an AND/OR graph in which the facts in the initial state constitute the initial nodes, other facts constitute OR nodes, and actions constitute AND nodes [9]. Edges then correspond to the relations between the facts and actions described by the preconditions and add effects of each action, with a directed edge from an AND node a to an OR node f when $f \in \mathit{add}(a)$, and from f to a when $f \in \mathit{pre}(a)$. Relaxed plans are then justifications for the goal set. This graph differs from the RPG used to compute landmarks or heuristics in that it only contains a single copy of each fact and action. RPGs correspond to unrolled versions of these graphs in which a copy of a node appears in every level of the graph after the first level in which it appears.

Many problems related to the delete relaxation can be understood as computations on this graph. For example, the h^+ heuristic is the cost of the lowest-cost justification J for the goal set G, where the cost of J is defined as the sum of the costs of the actions corresponding to the AND nodes it contains, and the $h^{\max}$ heuristic [1] is the minimum, over all justifications J for G, of the cost of the most costly path $\langle f_1, a_1, f_2, \ldots, a_{n-1}, f_n\rangle$ in J, where $f_1 \in V_{\mathrm{I}}$, $f_i \in V_{\mathrm{or}}$ for $i \neq 1$, $f_n \in G$, and $a_i \in V_{\mathrm{and}}$, and where the cost of a path is defined as above.

Definition 3 (AND/OR landmarks)
Given an AND/OR graph $\mathcal{G} = \langle V_{\mathrm{I}}, V_{\mathrm{and}}, V_{\mathrm{or}}, E\rangle$, a node n is a landmark for $V_{\mathrm{G}} \subseteq V_{\mathrm{I}} \cup V_{\mathrm{and}} \cup V_{\mathrm{or}}$ if $n \in V^J$ for all justifications J for V_{G}.

Intuitively, the landmarks for a set V_{G} in an AND/OR graph can be computed by considering the intersection of the vertex sets of all justifications for V_{G}, yet as the number of possible justifications is exponential, this method is intractable. However, the landmarks for V_{G} can also be characterised by the following system of equations:

$$\begin{aligned}
\mathit{LM}(V_{\mathrm{G}}) &= \bigcup_{v \in V_{\mathrm{G}}} \mathit{LM}(v) \\
\mathit{LM}(v) &= \{v\} && \text{if } v \in V_{\mathrm{I}} \\
\mathit{LM}(v) &= \{v\} \cup \bigcap_{u \in \mathit{pred}(v)} \mathit{LM}(u) && \text{if } v \in V_{\mathrm{or}} \\
\mathit{LM}(v) &= \{v\} \cup \bigcup_{u \in \mathit{pred}(v)} \mathit{LM}(u) && \text{if } v \in V_{\mathrm{and}}
\end{aligned}$$

where $\mathit{pred}(v) = \{u \mid \langle u, v\rangle \in E\}$.

Theorem 1 *For any AND/OR graph $\mathcal{G}$, the system of equations $\mathit{LM}(\cdot)$ has a unique* maximal *solution, where maximal is defined with regard to set inclusion, and this solution satisfies*

$$u \in \mathit{LM}(v) \iff u \text{ is a landmark for } \{v\} \text{ in } \mathcal{G}.$$

Moreover, for any node set V_{G}, $\mathit{LM}(V_{\mathrm{G}})$ is the set of landmarks for V_{G} in $\mathcal{G}$.

Proof sketch: Let $\mathit{LM}_{\mathrm{c}}(v)$ denote the complete set of landmarks for v. A solution to the system of equations exists, as it is satisfied by setting $\mathit{LM}(v) = \mathit{LM}_{\mathrm{c}}(v)$ for all v. To show that LM_{c} is the unique maximal solution, we show that all solutions to $\mathit{LM}(\cdot)$ satisfy $u \in \mathit{LM}(v) \Rightarrow u \in \mathit{LM}_{\mathrm{c}}(v)$. Define a counterexample X as a tuple $\langle u, v, J\rangle$ such that J is a justification for $\{v\}$, $u \in \mathit{LM}(v)$, $u \notin J$. Assume a counterexample exists and choose one where $|X| := |V^J|$ is minimal. Whether $v \in V_{\mathrm{and}}$ or $v \in V_{\mathrm{or}}$, it is possible to construct from this counterexample X a counterexample X' such that $|X'| < |X|$, contradicting the minimality of $|X|$. Hence, no counterexample exists. This shows that u must be contained in all justifications for $\{v\}$, which implies $u \in \mathit{LM}_{\mathrm{c}}(v)$. ■

The unique maximal solution to the $\mathit{LM}(\cdot)$ equations can be found in polynomial time by algorithms such as value iteration or the Bellman-Ford procedure, in the same way that these algorithms can be adapted to compute the additive heuristic h^{add} [8]. One way to compute the solution is to perform a fixpoint computation in which the set of landmarks for each vertex except those in V_{I} is initialized to the set of all of the vertices of the graph $\mathcal{G}$ and then iteratively updated by interpreting the equations as update rules. If the updates are performed according to the order in which nodes are generated in the relaxed planning graph (i. e., all nodes in the first layer, then all nodes in the second layer, etc.), then we obtain exactly the RPG label propagation algorithm by Zhu & Givan [12], computing *action landmarks* as well as causal fact landmarks. If only fact landmarks are sought, the equation for AND nodes can be modified to not include $\{v\}$ in $\mathit{LM}(v)$.

Orderings. Orderings for AND/OR landmarks can be defined analogously to orderings for planning landmarks, and they can be easily inferred from the *LM* sets. In particular, if u and v are two landmarks, we obtain a natural order $u \prec_{\mathrm{n}} v$ whenever $u \in \mathit{LM}(v)$.

For AND/OR graphs that represent delete relaxations, greedy-necessary orderings can also be computed with a slight extension. Let the set of *first achievers* for an OR node (fact) be defined as $\mathit{FA}(f) := \{a \mid a \in \mathit{pred}(f) \wedge f \notin \mathit{LM}(a)\}$. We can then infer $f \prec_{\mathrm{gn}} f'$ whenever $f \in \mathit{pred}(a)$ for all $a \in \mathit{FA}(f')$. Intuitively, this rule states that f is ordered greedy-necessarily before f' if f is a precondition for all actions that can possibly achieve f' for the first time. These orderings can be discovered during the computation of the landmarks and do not require any additional post-processing step.

5 LANDMARKS FROM THE Π^m PROBLEM

One method for estimating the cost of the delete relaxation is the previously mentioned $h^{\max}$ heuristic, which recursively estimates the cost of a set of facts as the cost of the most expensive fact in the set [1]. The $h^{\max}$ heuristic turns out to be a member of a more general formulation, the parameterised h^m family of heuristics which recursively estimate the cost of a set of facts G as the cost of the most expensive subset of G with size at most m [3]. For $m > 1$, this heuristic takes into account delete information in the problem, as a fact cannot be achieved in the context of a set to which it belongs with an action that deletes some other fact in the set.

It was recently shown that the h^m cost of a problem Π can be computed as the h^1 cost of a problem Π^m that results from a transformation of Π [2]. The facts of the new problem Π^m represent sets of facts of size m or less in the original problem. Its actions are obtained by making explicit in the precondition and add effects of the original actions those facts which, while not required or added by an action, may occur in the state in which the action is applied and persist after the application of the action, allowing them to be achieved in conjunction with the effects of the action. This is done by creating for each action a in Π a *set* of actions in the new problem, each having as a precondition in addition to the precondition of a itself, a set of facts C of size at most $m - 1$ such that C is disjoint from $\mathit{add}(a)$

and $del(a)$. For a set C and action a, the action a_C is then given by:

$$\begin{aligned} pre(a_C) &= \{S \mid S \subseteq (pre(a) \cup C) \wedge |S| \leq m\} \\ add(a_C) &= \{S \mid S \subseteq (add(a) \cup C) \wedge |S| \leq m\} \\ del(a_C) &= \emptyset \end{aligned}$$

Π^m is a problem with no delete effects that nevertheless encodes in its facts and actions some of the information about delete effects specified in the original problem. Any procedure applicable to a delete relaxation problem Π^+ can also be applied to Π^m to obtain information that can be translated back into the facts and actions of the original problem and used in that setting. In particular, the solution to the set of equations given above when the input is the Π^m problem defines *conjunctive* landmarks of size m or less that take into account *delete information* in the original problem Π.

Just as the h^m family of heuristics approaches optimality as m goes to infinity [3], it can be shown that the set of landmarks computed by the above procedure for Π^m will approach the complete and sound set of causal landmarks for the original problem Π. Yet since the complexity of computing Π^m and its size grow exponentially in m, this is unlikely to be feasible for high values of m.

Example. Consider the blocksworld problem of Figure 1. Apart from trivial landmarks such as those facts belonging to the initial state or goal, the complete set of causal delete-relaxation landmarks and orderings is *clear B* $\prec_{\text{gn}}$ *holding B*, implying that *holding B* must be made true in some state by any valid plan, and that *clear B* must be true in the state that immediately precedes it. In contrast, when the landmarks computation is applied to the Π^2 compilation of the problem, one of the obtained chains of orderings is the following:

$$\begin{aligned} &(\textit{clear B} \wedge \textit{holding A}) \prec_{\text{gn}} (\textit{clear B} \wedge \textit{handempty}) \prec_{\text{gn}} \\ &(\textit{holding B} \wedge \textit{ontable A}) \prec_{\text{gn}} (\textit{on B C} \wedge \textit{ontable A}) \prec_{\text{gn}} \\ &\qquad (\textit{on B C} \wedge \textit{holding A}) \end{aligned}$$

where $a \wedge b$ is a conjunctive landmark that implies that a and b must be true simultaneously in some state. These landmarks and orderings are only a subset of those found by the procedure, yet provide an almost complete roadmap for solving the problem.

The additional landmarks found in this way are not only conjunctive: the consideration of delete effects may also result in the discovery of fact landmarks for Π that are not landmarks in the Π^+ problem. In this example, the facts *holding A* and *ontable A* are also implied to be landmarks, as they are part of a conjunctive landmark.

6 EXPERIMENTAL RESULTS

We implemented the Π^m transformation and the computation of landmarks as discussed in Section 4. Here, we try to answer three main questions: whether our approach finds landmarks not found by previous approaches, whether these landmarks contain interesting information, and finally, whether current planners can exploit this information. All experiments were run on 2.3 GHz AMD Opteron machines using a 2 GB memory limit and 30-minute timeout.

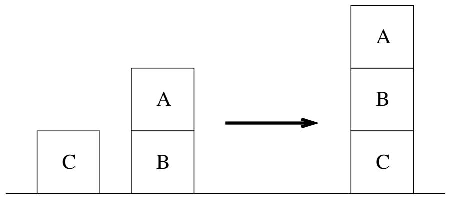

Figure 1. A blocksworld problem.

Table 1. Number of causal fact landmarks found by RHW and average ratio to this of the causal landmarks found by our approach. In the last column, conjunctive landmarks as well as facts are counted. Top part of table: STRIPS domains of IPC 1–5. Only solvable problems are listed for Mystery. Bottom part of table: domains of the optimal track of IPC 6. Numbers behind domain names show the number of tasks considered for that domain (the tasks where LM generation finished for all configurations).

Domain	# Causal LM Facts RHW	Ratio to RHW m = 1 (ZG)	m = 2 Facts	m = 2 Conj.
Airport (11)	1043	1.00	1.00	24.07
Blocks (35)	1444	1.00	1.05	8.36
Depot (21)	1379	1.07	1.13	13.11
Driverlog (19)	441	1.02	1.02	6.71
Freecell (54)	4110	1.26	1.27	15.33
Grid (4)	70	1.14	1.14	3.36
Gripper (20)	960	1.00	1.00	10.35
Logistics-1998 (23)	816	1.00	1.00	3.45
Logistics-2000 (28)	1319	1.00	1.00	4.02
Miconic (150)	7720	1.00	1.00	3.52
Mprime (26)	96	1.07	1.67	2.72
Mystery (16)	66	1.03	1.64	2.86
Openstacks (24)	2946	1.03	1.03	11.08
Pathways (30)	954	1.50	1.57	7.32
Pipesw. Not. (44)	754	1.22	1.29	4.25
Pipesw. Tank. (26)	524	1.15	1.24	5.42
PSR Small (50)	550	1.00	1.60	7.32
Rovers (32)	687	1.15	1.17	6.27
Satellite (23)	515	1.01	1.01	7.31
TPP (24)	751	1.13	1.32	5.94
Trucks (14)	467	1.23	1.25	8.92
Zenotravel (18)	309	1.05	1.05	5.25
Elevators (30)	629	1.12	1.12	3.66
Openstacks (30)	2925	1.03	1.03	11.37
PARC Printer (30)	2142	1.00	1.07	18.48
Peg Solitaire (30)	1457	1.00	1.02	19.33
Scanalyzer (26)	673	1.00	1.26	9.65
Sokoban (29)	605	2.73	5.25	43.02
Transport (30)	390	1.00	1.00	3.44
Woodworking (30)	1520	1.06	1.08	9.91

Number of Landmarks. Table 1 contrasts the number of causal landmarks found with the causal fact landmarks found by the RHW method [10] as used in the planner LAMA. With $m = 1$, our approach is equivalent to the procedure by Zhu & Givan [12], and in accordance with theory generates a superset of the causal fact landmarks that the RHW method finds, improving on the RHW method by 10–30% in several domains. With $m = 2$, we again generate a superset of the causal fact landmarks that $m = 1$ generates, improving on RHW by 10–60% in several domains. Particularly notable is the large number of conjunctive landmarks found with $m = 2$, surpassing the number of RHW facts by factors between 3 and 43. However, using $m = 2$ is computationally costly. Landmark generation with $m = 2$ timed out or ran out of memory in several cases in Airport and Freecell (as well as on large tasks in other domains that are far beyond the reach of current optimal planners).

Heuristic Accuracy of Landmark Information. In order to assess how the additional landmarks may influence heuristic accuracy, we use them in the LM-A* algorithm using the admissible landmark counting heuristic of Karpas & Domshlak [6], which we extend to handle conjunctive landmarks. Cost partitioning among landmarks is performed optimally. Table 2 shows the number of expanded states in

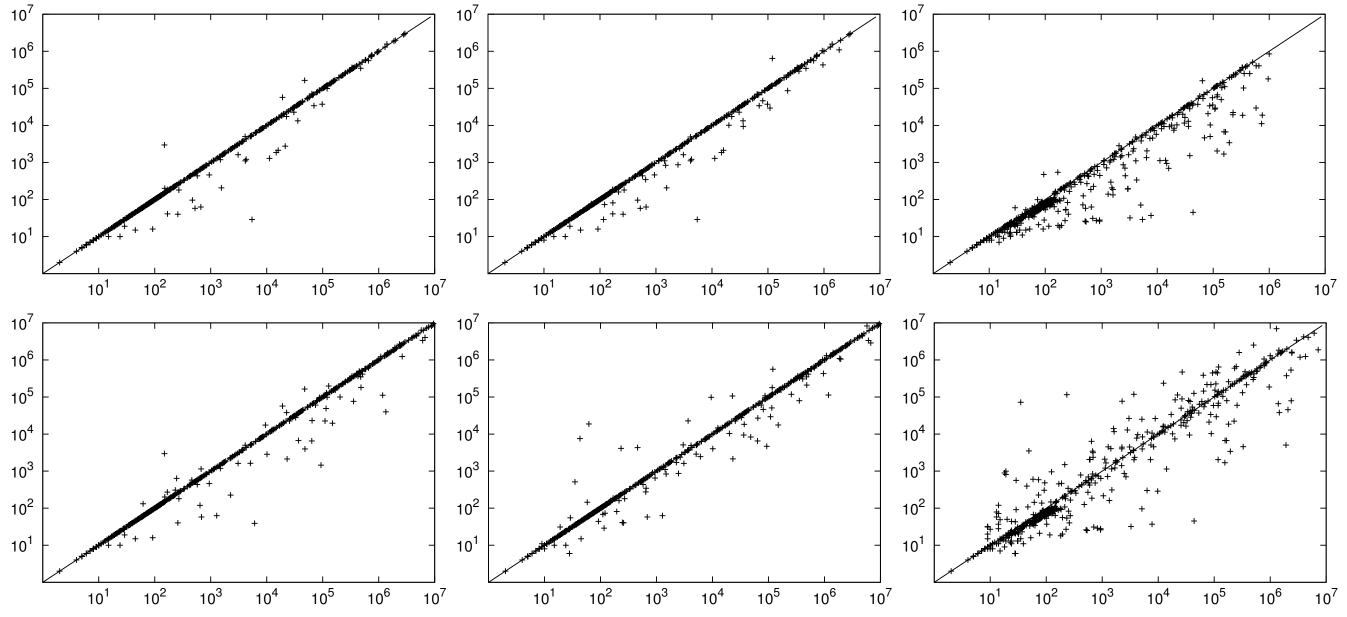

Figure 2. Expansions, compared to the RHW landmark generation (x-axes), of our approach using $m = 1$ (left), $m = 2$ when using only facts (middle), and $m = 2$ when using facts and conjunctive landmarks (right). Top row: optimal cost partitioning, bottom row: uniform cost partitioning.

Table 2. Expanded states when using the landmark generation of RHW and average improvement ratios of our approach using the optimal cost partitioning method. Numbers behind domain names show the number of tasks considered for that domain (the tasks solved by all configurations).

	RHW	**Improvement over RHW**		
	# Expansions	**m = 1**	**m = 2**	**m = 2**
Domain		**(ZG)**	**Facts**	**Conj.**
Airport (11)	384	1.00	1.00	1.12
Blocks (23)	2550007	1.00	1.00	7.84
Depot (4)	365373	1.07	1.48	3.75
Driverlog (8)	868496	1.00	1.00	1.02
Freecell (37)	189661	2.14	2.14	2.45
Grid (1)	270	1.50	1.50	1.64
Gripper (5)	458498	1.00	1.00	1.00
Logistics-1998 (3)	45663	1.00	1.00	1.48
Logistics-2000 (20)	862443	1.00	1.00	22.80
Miconic (141)	135213	1.00	1.00	1.34
Mprime (15)	313579	1.00	1.34	1.39
Mystery (12)	290133	1.00	1.00	1.00
Openstacks (7)	27392	1.00	1.00	1.00
Pathways (4)	152448	1.60	1.60	1.60
Pipesw. Not. (16)	1931233	1.05	1.05	1.46
Pipesw. Tank. (8)	29698	1.00	1.00	0.91
PSR Small (48)	697969	1.00	1.03	1.62
Rovers (5)	231520	1.06	1.06	1.06
Satellite (5)	1012920	1.01	1.01	1.08
TPP (5)	12355	1.00	1.00	1.00
Trucks (2)	108132	1.02	1.02	1.05
Zenotravel (8)	186334	1.00	1.00	1.02
Elevators (7)	483982	1.00	1.00	1.35
Openstacks (10)	649341	1.00	1.00	1.00
PARC Printer (12)	1118898	1.00	1.29	1.61
Peg Solitaire (23)	1734655	1.00	1.04	1.20
Scanalyzer (11)	23029	1.00	1.00	1.46
Sokoban (10)	1229907	1.02	1.05	0.90
Transport (9)	929285	1.00	1.00	1.00
Woodworking (10)	199666	1.41	1.41	2.35

those tasks solved by all configurations. We show results both for the case in which $m = 2$ is used only to compute additional facts, and for when the additional conjunctive landmarks are used during planning. As can be seen, the number of expansions is improved in some domains by 30–50% even when using only the additional facts found with $m = 2$. With conjunctive landmarks, improvements of factors above 2 occur in several domains, with Logistics-2000 showing an improvement beyond factor 22.

Figure 2 compares the expansion data from Table 2 with the number of expansions resulting from *uniform* cost partitioning. While our approach expands significantly fewer nodes than RHW when used in combination with optimal cost partitioning, with uniform partitioning this advantage is smaller for $m = 2$ when using only facts, and all but disappears for $m = 2$ when also using conjunctive landmarks.

Planning Performance. While optimal cost partitioning among landmarks leads to best heuristic accuracy, this method is unfortunately too costly to be competitive with the simpler uniform cost partitioning in terms of runtime and total number of problems solved. In Table 3, we report the total number of tasks solved with each of our experimental configurations when using the uniform partitioning method. Domains where landmark generation with $m = 2$ was computationally too costly (timing out in tasks that were solved by RHW) are shown in parentheses at the bottom of the table and not included in the total. Our approach with $m = 1$ solves more tasks than RHW, and $m = 2$ using only facts solves one more task than $m = 1$. Using conjunctive landmarks during planning, however, does not pay off.

The coverage results in this table are not as good as could be expected when considering the improvement in expanded states shown in Table 2. The scatter plots in Figure 2 indicate that this may in a large part be due to the uniform cost partitioning method.

Table 4 shows detailed results for selected domains, demonstrating how the benefit of additional heuristic accuracy does not always pay off compared to the extra computational effort needed for generating and managing the conjunctive landmarks. While in Logistics-2000,

Table 3. Solved problems when using the landmark generation of RHW and our approach using the uniform cost partitioning method. Numbers behind domain names show the total number of solvable tasks in that domain.

Domain	RHW	m = 1 (ZG)	m = 2 Facts	m = 2 Conj.
Blocks (35)	26	26	26	28
Depot (22)	7	7	7	7
Driverlog (20)	10	10	10	9
Grid (5)	2	2	2	2
Gripper (20)	7	7	7	7
Logistics-1998 (35)	3	3	3	3
Logistics-2000 (28)	20	20	20	22
Miconic (150)	142	142	142	142
Mystery (19)	15	15	15	15
Openstacks (30)	7	7	7	7
Pathways (30)	4	4	4	4
Pipesw. Not. (50)	19	19	19	18
Pipesw. Tank. (50)	12	13	13	11
PSR Small (50)	49	49	49	49
Rovers (40)	6	6	6	5
Satellite (36)	6	6	6	6
TPP (30)	6	6	6	6
Trucks (30)	2	2	2	2
Zenotravel (20)	8	8	8	8
Elevators (30)	13	13	13	14
Openstacks (30)	17	17	17	12
PARC Printer (30)	14	14	16	12
Peg Solitaire (30)	27	27	27	25
Scanalyzer (30)	9	9	9	6
Sokoban (30)	21	24	23	14
Transport (30)	11	11	11	11
Woodworking (30)	13	12	12	9
Total (951)	476	479	480	454
(Airport) (50)	(26)	(26)	(11)	(11)
(Freecell) (80)	(55)	(60)	(49)	(30)
(Mprime) (35)	(19)	(19)	(19)	(19)

Table 4. Detailed results for select domains, comparing $m = 2$ to RHW with respect to landmarks found, expanded states and runtime. Landmarks shown are causal facts for both approaches and conjunctive landmarks for $m = 2$ (second term in the sum).

	RHW			m = 2 using conj. LMs		
Inst.	**LM**	**Exp.**	**Time**	**LM**	**Exp.**	**Time**
Logistics-2000						
5-0	33	936	0.06	33 + 66	28	0.01
7-0	44	7751	0.58	44 + 112	37	0.03
10-0	56	194038	21.85	56 + 192	3421	3.69
11-0	61	156585	24.00	61 + 221	6706	7.65
12-0	56	117387	16.91	56 + 236	2041	2.82
Depot						
2	29	1488	0.08	34 + 193	310	0.18
4	54	2347873	220.95	60 + 111	531785	1237.57
7	42	167561	13.35	46 + 351	79755	78.00
10	47	1956533	197.43	55 + 82	375300	578.99
13	62	507369	77.64	62 + 625	336331	822.89
Driverlog						
3	10	1109	0.04	10 + 29	2105	0.10
5	17	247579	9.65	17 + 73	658799	73.10
7	17	26591	1.54	17 + 94	88915	19.66
10	14	504955	24.29	14 + 55	2506690	324.94
11	14	1298547	49.62	14 + 49	6969276	690.56

our approach using $m = 2$ performs better than RHW both with respect to expansions and time, in Depot, $m = 2$ performs better with respect to expansions, but worse with respect to time. Driverlog is an example where the conjunctive landmarks are not helpful at all and RHW performs better both with respect to expansions and time. We also found that while having more causal fact landmarks *usually* translates to better heuristic accuracy, this is not always the case when using the uniform cost partitioning scheme.

7 CONCLUSIONS AND FUTURE WORK

We have shown how to declaratively define the complete set of causal landmarks for AND/OR graphs. Combined with the Π^m compilation, this results in a parameterised method that permits the computation of conjunctive and fact landmarks that take into account delete information in planning problems. Our experimental results indicate that the use of these landmarks can significantly increase the accuracy of landmark-based admissible heuristics.

Future work includes the investigation of complete and approximate methods for decreasing the size of the Π^m problem by eliminating m-fluents that are irrelevant in the context of landmark generation. Another line of research is to develop cost-partitioning schemes that offer favourable tradeoffs between the speed of the uniform scheme and the heuristic quality of the optimal scheme.

ACKNOWLEDGEMENTS

We thank Héctor Geffner for inspiration and extensive discussion and Erez Karpas for his help with the LM-A* implementation. NICTA is funded by the Australian Government, as represented by the Department of Broadband, Communications and the Digital Economy, and the Australian Research Council, through the ICT Centre of Excellence program. This work was partly supported by the German Research Foundation (DFG) as part of SFB/TR 14 "Automatic Verification and Analysis of Complex Systems" (AVACS).

REFERENCES

[1] Blai Bonet and Héctor Geffner, 'Planning as heuristic search', *AIJ*, **129**(1), 5–33, (2001).
[2] Patrik Haslum, '$h^m(P) = h^1(P^m)$: Alternative characterisations of the generalisation from $h^{\max}$ to h^m', in *Proc. ICAPS 2009*, pp. 354–357, (2009).
[3] Patrik Haslum and Héctor Geffner, 'Admissible heuristics for optimal planning', in *Proc. AIPS 2000*, pp. 140–149, (2000).
[4] Malte Helmert and Carmel Domshlak, 'Landmarks, critical paths and abstractions: What's the difference anyway?', in *Proc. ICAPS 2009*, pp. 162–169, (2009).
[5] Jörg Hoffmann, Julie Porteous, and Laura Sebastia, 'Ordered landmarks in planning', *JAIR*, **22**, 215–278, (2004).
[6] Erez Karpas and Carmel Domshlak, 'Cost-optimal planning with landmarks', in *Proc. IJCAI 2009*, pp. 1728–1733, (2009).
[7] Emil Keyder and Héctor Geffner, 'Heuristics for planning with action costs revisited', in *Proc. ECAI 2008*, pp. 588–592, (2008).
[8] Yaxin Liu, Sven Koenig, and David Furcy, 'Speeding up the calculation of heuristics for heuristic search-based planning', in *Proc. AAAI 2002*, pp. 484–491, (2002).
[9] Vitaly Mirkis and Carmel Domshlak, 'Cost-sharing approximations for h^+', in *Proc. ICAPS 2007*, pp. 240–247, (2007).
[10] Silvia Richter, Malte Helmert, and Matthias Westphal, 'Landmarks revisited', in *Proc. AAAI 2008*, pp. 975–982, (2008).
[11] Silvia Richter and Matthias Westphal, 'The LAMA planner: Guiding cost-based anytime planning with landmarks', *JAIR*, (2010). To appear.
[12] Lin Zhu and Robert Givan, 'Landmark extraction via planning graph propagation', in *ICAPS 2003 Doctoral Consortium*, pp. 156–160, (2003).