Exploring the effectivity of AND/OR landmark extraction on modern planning domains

How does the AND/OR landmark extraction algorithm perform on recent competition domains?

Pauline Hengst

Supervisor(s): Issa Hanou , Sebastijan Dumančić

EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
February 13, 2024

Name of the student: Pauline Hengst
Final project course: CSE3000 Research Project
Thesis committee: Issa Hanou , Sebastijan Dumančić , Luís Miranda da Cruz

Abstract

This paper explores the application of landmark-based planning algorithms, specifically focusing on AND/OR landmark extraction methods. Drawing from classical planning principles and recent advancements, we investigate the effectiveness of landmark extraction in guiding the search for solutions to planning problems. Our research questions center on identifying effective domains for landmark extraction, assessing the utility of extracted landmarks, and comparing our implementation with existing literature. Utilizing the Symbolicplanners.jl framework, we implement AND/OR landmark extraction and evaluate its performance across various domains. Due to challenges in implementation, the landmarks we were able to extract had limited meaning. We propose future work to refine the AND/OR method and expand our analysis to include the Hm procedure.

1 Introduction

Planning is used in many different domains, including robotics, logistics, and project management[7]. As such planning algorithms are very common. The challenge lies in efficiently finding solutions to these planning problems, given their inherent complexity. This prompts continuous efforts for enhancement of these algorithms. Such a method for improving planning algorithms is utilizing *landmarks*, which are steps that are required to reach a solution to a planning problem [6]. There are two components to utilizing landmarks in planning: extracting (i.e generating) the landmarks from a problem, and subsequently using the landmarks in the planning algorithm. Both of these steps must be efficient for the landmarks to speed up a planning task. Usually, landmarks are incorporated into a heuristic (e.g. landmark count) or marked as a sub-goal.[7]

To use a planning system for a problem, it must first be encoded into a problem representation. Such an encoding enables a system to convert a desired problem into abstract actions and states. This generalizes the solving task to a search task. This search is in a state-action space, for the solution to a plan, representing a chain of actions that take us to a desired goal state. STRIPS [2] and PDDL[1] enable the definition of such problems. The DELETE-relaxation simplifies planning problems by removing the delete list from each action. From this relaxation, landmarks can be extracted.[10]. DELETE-relaxations have been shown to be an instance of AND/OR graphs [5], a type of graph that describes task decomposition in terms of alternative goals (OR nodes) or sub-tasks (AND nodes). The work of Keyder, Richter, and Helmert [4] builds on the work of Zhu and Givan [10] and Mirkis and Domshlak [5] to show that landmarks can thus also be derived from AND/OR graphs.

The performance of the HM landmark extraction has been observed in various studies, however many domains have not been extracted. An overview of the domains added per year from the IPC can be found in the background section.

The main goal of this paper is to build on the work of Keyder, Richter, and Helmert [4] by implementing landmark extraction for AND/OR graphs, and asses its validity. The secondary goal is to scope the effectiveness of the landmark extraction across newer domains. This leads to the following research question: ***"How does the Hm landmark extraction algorithm perform on recent competition domains?"***, with the following sub-questions:

1. Which domains are currently known to be effective for AND/OR landmarks?
2. How do you measure the effectivness of a heuristic and a Landmark?
3. Which heuristics and domains does the symbolic planner currently implement? How much of this intersects with known effective domains of landmarks?
4. Which heuristics are most widely used in practice? How do these heuristics compare to those effective for AND/OR landmarks?
5. How suitable is the Symbolicplanner.jl library in measuring for these questions?

Research questions 1-3 will be answered qualitatively, the remaining questions will be measured through the implementation of AND/OR landmark extraction. The Julia language[1] has been chosen to implement landmark extraction algorithms.

This paper is structured as follows: section 2 provides background and related work, offering an overview of terminology and recent related works. Section 3 elaborates on the implementation and experimental setup for evaluating AND/OR landmark extraction algorithms. Section 4 presents the results from experiments. Section 5 discusses ethical implications and reproducibility of the research. Section 6 analyzes the results and their implications. Finally, section 7 summarizes key contributions and suggests directions for further exploration.

2 Background

In this section, we provide a basic understanding of planning algorithms and landmark generation and information relevant to the research.

2.1 Classical planning

Classical planning (also known as deterministic or sequential planning) is a method of finding a sequence of actions to achieve a desired goal or set of goals within a given environment or system, as described in the introduction.

In planning, heuristics guide the search for solutions within the solution space. Typically, heuristics are functions that estimate or measure the distance from a given state to the goal state [7].

Different types of landmarks can be identified:

---

[1] Julia is a high-level language aimed for use by the (non-computer-science) scientific community. It uses JIT (just-in-time) compilation features to ensure it is high-performing enough for complex tasks. For more information on the language: https://julialang.org/

Conjunctive and disjunctive landmarks: Pair of landmarks where both landmarks or neither landmark must be true. **Action and fact landmarks:** Landmarks that are either actions that must be taken or states(facts) that must occur.

Two planning systems that incorporate landmarks are the LAMA planner and the Fastdownaward planning system. LAMA [8] was used in the classical track of the IPC 2018 as a baseline, built on top of FastDownward. FastDownward is a domain-independent classical planning system [3].

2.2 Problem encoding

The STRIPS (Stanford Research Institute Problem Solver) is a framework to describe planning tasks [2]. In this framework, a state is represented as an assignment of all possible state variables, while an action is defined as the removal and addition of one or more variables to a state, each action requiring specified preconditions to be possible.

Concretely, A STRIPS 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(a)$\in R_0^+$,
- $I \subseteq F$ is the initial state, and
- $G \subseteq F$ is the set of goals.

To facilitate the description of planning problems, the Planning Domain Definition Language (PDDL) was developed. PDDL provides a standardized syntax and semantics for encoding real-world planning problems, allowing planners to interpret and convert them to their internal representation (such as STRIPS). Each PDDL *domain* represents a problem type, defining object types, possible actions[1]. PDDL problems or *instances* instantiate these domains by declaring the objects and goals that exist in the problem to be solved.

2.3 AND/OR landmark extraction

This section describes the AND/OR landmark extraction procedure and Hm landmark extraction procedure as described in Keyder, Richter, and Helmert [4]. The procedures consist of three steps; building the and or graph, extracting the landmarks, and in the case of hm extraction, transforming the input problem.

An AND/OR graph $G = \langle V_I, V_{\text{and}}, V_{\text{or}}, E \rangle$ is a directed graph with vertices $V_G := V_I \cup V_{\text{and}} \cup V_{\text{or}}$ and edges E, where V_I, V_{and}, and V_{or} are disjoint sets of initial nodes, AND nodes, and OR nodes, respectively.[4]

How a STIPS-style problem can be converted to an AND/OR graph is shown in algorithm 1.

Landmarks for each node can then be extracted by doing a fixpoint calculation. Each node is initialized to have all nodes as a landmarks, and the landmarks are traversed in the order that are added to the graph. then the following rules are used to update the landmark sets per node:

$$\begin{array}{ll}
\text{for a } v \in V_G & \text{LM}(v) \\
\text{LM}(v) = \{v\} & \text{if } v \in V_I \\
\text{LM}(v) = \{v\} \cup \{u \in \text{pred}(v) \mid \text{LM}(u)\} & \text{if } v \in V_{\text{or}} \\
\text{LM}(v) = \{v\} \cup \{u \in \text{pred}(v) \mid \text{LM}(u)\} & \text{if } v \in V_{\text{and}}
\end{array}$$

Algorithm 1 AND/OR graph generation

```
Input: STRIPS problem $P$ with Action set, fact set, Initial states
Add all initial facts to $V_I$
Add all remaining facts to $V_{or}$
for each action $a$ in $Actions$ do
    For every $f$ in add($a$), add edge $(a, f)$
    For every $f$ in pre($a$), add edge $(f, a)$
end for
```

Once the landmark set remains stable after update rules are applied (the fixpoint is achieved), the set of landmarks for each node has been found. This results in the same landmark sets per node as the Zhu and Givan [10] algorithm. To obtain landmarks for goals, goal nodes must be marked in the graph generation step and can taken from the resulting landmark set.

Hm landmark generation is an expansion of the AND/OR procedure. It is derived on the HM heuristic and cost calculation methods. HM extraction expands AND/OR extraction by transforming the input problem to the AND/OR algorithm.

Each action a in the input delete relaxed graph becomes a set of actions such that the set of preconditions is disjoint from the add and delete effects of *a*. This set is computed by iterating over the preconditions (and preconditions of preconditions) of a until a set of size *m+1* is found. Consequently, Hm landmark extraction with the *m* parameter set to 0 equals the AND/OR procedure. The benefit of the Hm procedure is the ability to find conjunctive landmarks, but has a very high time cost.

3 Methodology

This section describes what was implemented in symbolic planners and its difference to the existing literature. Hm-landmark extraction has been implemented in the LAMA planner [8] on top of the Fastdownward planner[3]. These frameworks are able to extract ordered, conjunctive, fact landmarks [2], using the Hm procedure.

To find out which domains are known to be effective, known literature will be surveyed in the area. Of special interest is data from planning competitions, such as the International Planning Competition. Within these completions, data from the **deterministic** track will be taken into account, from the available deterministic planning tracks. The most recent edition with a deterministic track was determined to be the IPC of 2018 [3]. These competitions yield domains suitable for state-of-the-art planners, so they are the most interesting to examine. The effectiveness of the use of landmarks is gauged differently across the literature, with landmark count, final plan length, and count of states traversed factoring into different metrics [10][4]. As we are not generating plans in this paper, we will consider landmark generation effective onto a given domain if: of the traversed states many (but not all) are considered landmarks, once trivial landmarks (goal and initial states) are removed.

---

[2] https://www.fast-downward.org/Doc/LandmarkFactory

[3] https://ipc2018-classical.bitbucket.io/

The utility of individual landmarks can be deducted from their properties. By intuition, the more information a landmark gives us, the more information can be used for a heuristic, and the more it will improve planning performance. On the other hand, having fewer landmarks is better than having many because we can exclude more items from the search. We will survey the literature to find out which qualities landmarks deducted from AND/OR landmarks gives.

Similarly, a landmark extraction procedure can be deemed effective overall if it is shown to extract non-trivial landmarks for a problem. However, as landmark extraction should not weigh down the overall time of plan generation, time must be taken into account as a measure of the quality of an extraction procedure. As we cannot examine the relative time the landmark generation takes relative to plan generation (since no plans are generated in this research), time will be compared to that found in the literature and related to problem size.

Currently symbolicplanners is able to handle ”Support for PDDL domains with numeric fluents and custom datatypes”[4] so should work with all types of found domains.

Summarizing, to complete the research objective we will:

1. Find domains compatible with Symbolicplanners.jl where and/or landmark extraction is known to be effective
2. Survey literature and data to find - what types of landmarks are extracted, whether these types are known to be effective
3. Implement AND/OR landmark extraction and apply to the found domains in the Symbolicplanners framework
4. Examine the extracted landmarks and compare extraction time and landmark quality to literature

3.1 Modifications to theoretical method

This section motivates some modifications made to the AND/OR algorithm described in section 2.3.

Performant Julia requires type stability, which is when one strives to maintain consistency with variable type throughout the program's execution. Julia can deduct types but recompiles a function every time a new type is computed. As such it has been opted to use as many existing types as possible from the Symbolicplanners library.

Propositional state variables (single boolean variables as used in STRIPS) have been replaced for propositional terms as used by Zhi-Xuan [9], as not to replace the terms with new data structures. The consequence is that state variables are no longer atomical but can be compounded into composite terms representing more complex logical statements. For example, (on(A, Table) and (on(A, B))) can now be represented as one node in the AND/OR graph.

The representation of the delete relaxation had been modified similarly. In the existing framework, the delete effects of ground actions are encoded as a `not` term. Instead of modifying the relaxation graph from which the and or graph is built, `not` edges from AND nodes are discarded.

4 Experimental Setup and Results

This section details the exact experiment formulated from the methodology, and presents the obtained results. The AND/OR procedure was implemented in line with section **??** with some modifications, described in section 3.1, and one experiment was executed. These were run on an 12th Gen Intel® Core™ with 16GB of main memory. The code and experiments are publicly available[5].

Experiment 1 benchmarks the implementation by comparing its extracted landmarks to those of the implementation of the Zhu and Givan [10] algorithm, using domains from Keyder, Richter, and Helmert [4]. The total set of domains for this experiment is: `Blocksworld, Grid, Miconic`, taken from the IPC 2008 benchmark set.

Table 1 shows the results from this experiment.

5 Responsible Research

In ensuring responsible research practices, we prioritize transparency and accessibility of our work. To this end, we will provide links to our code both within our paper and through the TU Delft repository, enhancing its accessibility and ensuring its long-term availability. Additionally, we will store PDDL domains in the repository to safeguard against potential loss from other sources. Our code is documented and adheres to industry standards, enabling comprehension of its intentions and limitations. Through these measures, we uphold ethical standards and facilitate the reprehensibility of our methods, promoting integrity and accountability in our research.

6 Discussion

The results of the first experiment, described in 1, are lacking. This is because of problems with the implementation. As shown, the amount of landmarks generated is equal to the amount of goals in the problem. On further inspection, the landmarks generated for each were the goals themselves. Inspecting the results showed that this was the case for every fact node in the original graph, meaning that no meaningful fact landmarks were extracted per problem.

Reproducing the original work proved some difficulties. The Hm procedure has been implemented in the LAMA planner[8]. The existence of this implementation was discovered very late in the research process as the naming convention of these implementations was unclear to the author; the Hm extraction did not get this name in the work of Keyder, Richter, and Helmert [4]. As a consequence, it was decided to implement with only the description of the process in the paper as a reference.

The problems in the generation of the landmarks most likely lie in the order of the traversal of the landmarks. In the main work, it is stated that to obtain the same landmarks as the Zhu and Givan [10] implementation, the landmark sets must be updated in the same order as the nodes are added to the original relaxed planning graph. Extracting this ordering

---

[4] https://juliaplanners.github.io/SymbolicPlanners.jl/stable/

[5] https://github.com/PaulTervoort/SymbolicPlanners.jl-landmarks/tree/final-pauline

	Symbolicplanners.jl			LAMA		
	Landmarks found	States traversed	Amount of goal states	Landmarks found	states traversed	(s)
Blocksworld(20)	9	352	9	..	..	..
Grid(5)	17489	7	7	..	..	..
Miconic(12)	3	52	3	..	..	

Table 1: Comparing Implemented method (Symbolicplanners.jl) to existing work in the LAMA planner.

proved unsuccessful, a serious limitation of the produced implementation.

As the generation of Hm landmarks relies on the AND/OR implementation, these results on the method could not be obtained.

Another consequence of the implementation is that an experiment with newer domains could not be executed. With a working implementation, another experiment would have been run, to inspect the landmarks extracted on more complex domains, taken from the IPC 2018 deterministic-track set.

7 Conclusions and Future Work

In conclusion, our research primarily aimed to assess the effectiveness of AND/OR landmark algorithms. Despite encountering challenges in implementation, our investigation shed light on crucial limitations in landmark generation, particularly in extracting meaningful fact landmarks. The discrepancy between the number of generated landmarks and their relevance to problem-solving tasks underscores the need for refinement in extraction techniques. Additionally, our exploration uncovered discrepancies in existing implementations, highlighting the importance of clarity and accessibility in research endeavors.

Future work entails correcting the AND/OR method in the Symbolicplanners framework and expanding it to apply the Hm procedure. The current implementation does extract sets of landmarks larger than the node itself for actions, these are also yet to be inspected.

References

[1] Constructions Aeronautiques et al. "Pddl— the planning domain definition language". In: *Technical Report, Tech. Rep.* (1998).

[2] Richard Fikes and Nils J. Nilsson. "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving". In: *Artif. Intell.* 2 (1971), pp. 189–208. URL: https://api.semanticscholar.org/CorpusID:8623866.

[3] Malte Helmert. "The fast downward planning system". In: *Journal of Artificial Intelligence Research* 26 (2006), pp. 191–246.

[4] Emil Keyder, Silvia Richter, and Malte Helmert. "Sound and Complete Landmarks for And/Or Graphs". In: *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*. NLD: IOS Press, 2010, pp. 335–340. ISBN: 9781607506058.

[5] Vitaly Mirkis and Carmel Domshlak. "Cost-Sharing Approximations for h+." In: *ICAPS*. Citeseer. 2007, pp. 240–247.

[6] Julie Porteous, Laura Sebastia, and Jörg Hoffmann. "On the extraction, ordering, and usage of landmarks in planning". In: *Sixth European Conference on Planning*. 2014.

[7] Silvia Richter. "Landmark-Based Heuristics and Search Control for Automated Planning (Extended Abstract)". In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. IJCAI '13. Beijing, China: AAAI Press, 2013, pp. 3126–3130. ISBN: 9781577356332.

[8] Silvia Richter and Matthias Westphal. "The LAMA planner". In: *Journal of Artificial Intelligence Research* 39 (2010), pp. 127–177.

[9] Tan Zhi-Xuan. "PDDL. jl: An Extensible Interpreter and Compiler Interface for Fast and Flexible AI Planning". PhD thesis. Massachusetts Institute of Technology, 2022.

[10] Lin Zhu and Robert Givan. "Landmark extraction via planning graph propagation". In: *ICAPS Doctoral Consortium* (2003), pp. 156–160.