Landmark Heuristics for Lifted Planning – Extended Abstract

Julia Wichlacz, Daniel Höller, Jörg Hoffmann
Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
{wichlacz, hoeller, hoffmann}@cs.uni-saarland.de

Abstract

Planning problems are usually modeled using lifted representations, they specify predicates and action schemas using variables over a finite universe of objects. However, current planning systems like Fast Downward need a grounded (propositional) input model. The process of grounding might result in an exponential blowup of the model size. This limits the application of grounded planning systems in practical applications. Recent work introduced an efficient planning system for lifted heuristic search, but the work on lifted *heuristics* is still limited. In this extended abstract, we introduce a novel lifted heuristic based on landmarks, which we extract from the lifted problem representation. Preliminary results on a benchmark set specialized to lifted planning show that there are domains where our approach finds enough landmarks to guide the search more effective than the heuristics available.

Introduction

The standard in planning is heuristic search (Hoffmann and Nebel 2001; Helmert and Domshlak 2009; Richter and Westphal 2010; Seipp 2019). But it relies on transforming the *lifted* input descriptions, which describe actions and predicates in a parameterized form using variables ranging over a finite universe of objects, into a grounded task representation. A drawback of this approach is that the size of the grounded task representation may – in the worst case – grow exponentially with regard to action and predicate arity. But in many real world problems both can be quite large and the problems therefore quickly become infeasible to solve using grounded representations (see e. g. (Hoffmann et al. 2006; Koller and Hoffmann 2010; Koller and Petrick 2011; Haslum 2011; Matloob and Soutchanski 2016)). *Lifted* planning does not rely on a grounded task representation and instead works directly on the lifted planning models. While this is not a new concept (Penberthy and Weld 1992; Russell and Norvig 1995; Younes and Simmons 2003), lifted planning was only lately combined with heuristic search planning.

An effective lifted forward search planning mechanism, the *Power Lifted Planner (PWL)* has recently been developed by Corrêa et al. (2020). It avoids the use of a completely grounded representation and instead only grounds actions on demand during search by using techniques as used in database queries. Since the computation of heuristics is usually based on a grounded representation, the question remains how they can be transferred into the lifted setting. As a first heuristic, PWL uses goal counting. Lifted heuristics from the literature are based on delete relaxation, using symmetries to reduce the blow-up (Ridder and Fox 2014), exploiting a link to database technology (Corrêa et al. 2021), or splitting predicates into smaller predicates with a fixed arity (Lauer et al. 2021).

We present a novel approach based on landmarks. Landmarks are facts that need to be fulfilled at some point in the plan (Hoffmann, Porteous, and Sebastia 2004). They are computed before search and can be used to estimate the distance to the goal. Landmarks form the basis for a wide range of heuristics and have been exploited in many different ways (Karpas and Domshlak 2009; Helmert and Domshlak 2009; Richter and Westphal 2010). They are a good fit for lifted planning, since the computational overhead of generating the landmarks only occurs once before the search. We adapted a known method of computing landmarks (Hoffmann, Porteous, and Sebastia 2004) to the lifted setting by grounding actions and predicates only partially. The resulting heuristic dominates goal-counting, while being cheaper to compute than delete relaxation, thereby forming a solid middle ground between the two. Our preliminary experiments show that this middle ground can be useful.

Lifted Landmarks

In this work we only consider fact landmarks (Hoffmann, Porteous, and Sebastia 2004). In a ground setting, a fact p is a landmark for a given planning problem Π if and only if for every solution π for Π, π traverses a state s with $p \in s$.

Deciding whether a fact is a landmark is a computationally hard task. In ground classical planning, it is **PSPACE**-complete. Therefore techniques used in practice use sufficient criteria to (under-)approximate the set of landmarks in polynomial time (usually under delete-relaxation).

Heuristics often use ordering relations between landmarks to improve heuristic values. These are defined as follows. Let p and q be two landmarks.

- p is ordered directly before q, written $p \rightarrow_D q$, if a state where p is made true is preceded by a state where q holds.

- p is ordered reasonable before q, written $p \rightarrow_R q$, if making q true before p makes it necessary to delete and re-achieve p later on.

Based on these definitions from grounded planning, we define lifted landmarks as follows.

Definition 1 (Lifted Landmarks) *A partially grounded predicate* $P(u_1, \ldots, u_k)$ *is a* lifted landmark *for a given planning problem* Π *if and only if there is a ground instance that is a landmark for* Π.

This forms a compact and easy to extract definition of a special case of disjunctive landmarks (Hoffmann, Porteous, and Sebastia 2004; Helmert and Domshlak 2009), which might also be useful in grounded planning.

Consider e. g. a simple transport domain where a package *pa* must be in some truck to be carried to its destination. Though we might not know which truck is used, we know that the predicate $in(pa, v_t)$ is a landmark, where v_t is a variable of type *truck*.

Landmark Extraction

We realized a lifted extraction of *necessary subgoals* (Hoffmann, Porteous, and Sebastia 2004). Landmarks are thereby extracted via a backchaining procedure, starting from the goal condition:

- Every fact contained in the goal condition is a landmark.
- Given a landmark p, let A_p be the set of all actions with p in the add effect. We then determine the intersection of the preconditions of all actions in A_p. Since p has to be added by one of the actions, an atom q contained in the intersection of their preconditions is also a landmark.

This process is continued until no new landmarks are found.

We use the same mechanism to generate *lifted* landmarks. We start with the goal condition from the problem definition, which induces a set of lifted landmarks $P(u_1, \ldots, u_k)$. Since the goal condition is usually defined fully grounded, these initial landmarks are fully instantiated predicates.

Given a lifted landmark $P(u_1, \ldots, u_k)$ and the set of lifted actions with P in the add effects, we partially ground the actions to match the objects in the landmark.

Next we consider the resulting partially grounded actions and intersect the predicates contained in their preconditions. Let $Q(v_{11}, \ldots, v_{m1}), \ldots, Q(v_{1n}, \ldots, v_{mn})$ be these preconditions. We introduce a landmark $Q(w_1, \ldots, w_n)$ with

$$w_i := \begin{cases} o & w_{ij} = o \text{ for all } j \\ x_i & \text{otherwise} \end{cases}$$

where the x_i are arbitrary variables. We continue the process as in the grounded algorithm.

Besides the actual landmarks, the extraction process also generates $\rightarrow_D$ ordering relations between them (in the grounded as well as in the lifted case). The combination of these ordering relations and standard notions of interference can be used to extract $\rightarrow_R$ ordering relations (Hoffmann, Porteous, and Sebastia 2004). We have implemented the method described by Koehler and Hoffmann (2000), which is cheaper to compute in the lifted setting.

Landmark Heuristics

Based on our lifted landmarks, we implemented a heuristic similar to the one used by the LAMA system (Richter and Westphal 2010) in grounded planning and integrated it into the PWL planning system (Corrêa et al. 2020).

The basic idea is to count unfulfilled landmarks. However, the heuristic values can be improved by taking the ordering relations into account. Due to the lifted setting, we have to replace the simple contains test, which usually checks if a landmark is contained in the given state, by a test that checks whether there is an instantiation of the landmark contained in the current state.

Discussion

We ran a preliminary evaluation on the benchmark sets used in recent papers on lifted planning (Corrêa et al. 2020; Lauer et al. 2021). We compared our system against blind breadth-first search as well as greedy best first search in combination with goal counting and the lifted delete relaxation heuristic h^{Ladd} introduced by Corrêa et al. (2021). All heuristics are implemented in PWL. We also ran the grounded system Fast Downward (FD) (Helmert 2006) with the goal counting and h^{FF} (Hoffmann and Nebel 2001) heuristics.

As expected, FD performs well when grounding is feasible, but this is only the case on a few instances. When we compare our configurations, it can be seen that our system benefits from using the ordering relations on top of the landmarks. Compared to the lifted implementation of goal counting, our heuristic comes with some overhead, but it yields significant gains in domains where the approach is able to extract enough landmarks (currently this is the case in the *blocks* and the *rovers* domains). h^{Ladd} performs best in combination with preferred operators. This configuration performs well across most domains. It shows its worst performance in the *blocks* domain, which is known to be difficult for delete relaxation heuristics, and in the *ged* domain.

Apart from the necessary subgoals method, we experimented with the landmark extraction introduced by Keyder, Richter, and Helmert (2010). It represents a delete-relaxed planning problem as an AND/OR graph and extracts landmarks from this graph. It is complete for a certain type of landmarks. We implemented the approach using on demand grounding and ran the process in a depth-first manner to avoid keeping the entire model in memory. However, so far, the approach is still not competitive on the used benchmarks.

To conclude, it can be seen from our early results that landmarks are a promising source of information for lifted heuristic search. However, more work has to be done in landmark extraction and – as shown by the performance of the respective h^{Ladd} configuration – the derivation of preferred operators to fully exploit their potential.

Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102.

References

Corrêa, A. B.; Pommerening, F.; Helmert, M.; and Francès, G. 2020. Lifted Successor Generation using Query Optimization Techniques. In *Proceedings of the 30th International Conference on Automated Planning and Scheduling (ICAPS'20)*, 80–89. AAAI Press.

Corrêa, A. B.; Pommerening, F.; Helmert, M.; and Francès, G. 2021. Delete-Relaxation Heuristics for Lifted Classical Planning. In *Proceedings of the 31st International Conference on Automated Planning and Scheduling (ICAPS'21)*. AAAI Press.

Haslum, P. 2011. Computing Genome Edit Distances using Domain-Independent Planning. In *Proceedings of the ICAPS Scheduling and Planning Applications woRKshop (SPARK'11)*.

Helmert, M. 2006. The Fast Downward Planning System. *Journal of Artificial Intelligence Research* 26: 191–246.

Helmert, M.; and Domshlak, C. 2009. Landmarks, Critical Paths and Abstractions: What's the Difference Anyway? In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS'09)*, 162–169. AAAI Press.

Hoffmann, J.; Edelkamp, S.; Thíebaux, S.; Englert, R.; Liporace, F.; and Trüg, S. 2006. Engineering Benchmarks for Planning: the Domains Used in the Deterministic Part of IPC-4. *Journal of Artificial Intelligence Research* 26: 453–541.

Hoffmann, J.; and Nebel, B. 2001. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research* 14: 253–302.

Hoffmann, J.; Porteous, J.; and Sebastia, L. 2004. Ordered Landmarks in Planning. *Journal of Artificial Intelligence Research* 22: 215–278.

Karpas, E.; and Domshlak, C. 2009. Cost-Optimal Planning with Landmarks. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*, 1728–1733. Morgan Kaufmann.

Keyder, E.; Richter, S.; and Helmert, M. 2010. Sound and Complete Landmarks for And/Or Graphs. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI'10)*, 335–340. IOS Press.

Koehler, J.; and Hoffmann, J. 2000. On Reasonable and Forced Goal Orderings and their Use in an Agenda-driven Planning Algorithm. *Journal of Artificial Intelligence Research* 12: 338–386.

Koller, A.; and Hoffmann, J. 2010. Waking Up a Sleeping Rabbit: On Natural-Language Sentence Generation with FF. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS'10)*, 238–241. AAAI Press.

Koller, A.; and Petrick, R. 2011. Experiences with Planning for Natural Language Generation. *Computational Intelligence* 27(1): 23–40.

Lauer, P.; Torralba, A.; Fiŝer, D.; Höller, D.; Wichlacz, J.; and Hoffmann, J. 2021. Polynomial-Time in PDDL Input Size: Making the Delete Relaxation Feasible for Lifted Planning. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21)*. IJCAI organization.

Matloob, R.; and Soutchanski, M. 2016. Exploring Organic Synthesis with State-of-the-Art Planning Techniques. In *Proceedings of the ICAPS Scheduling and Planning Applications woRKshop (SPARK'16)*, 52–61.

Penberthy, J. S.; and Weld, D. S. 1992. UCPOP: A Sound, Complete, Partial Order Planner for ADL. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, 103–114. Morgan Kaufmann.

Richter, S.; and Westphal, M. 2010. The LAMA Planner: Guiding Cost-Based Anytime Planning with Landmarks. *Journal of Artificial Intelligence Research* 39: 127–177.

Ridder, B.; and Fox, M. 2014. Heuristic Evaluation based on Lifted Relaxed Planning Graphs. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS'14)*, 244–252. AAAI Press.

Russell, S.; and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice-Hall. ISBN 0-13-103805-2.

Seipp, J. 2019. Pattern Selection for Optimal Classical Planning with Saturated Cost Partitioning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*, 5621–5627. IJCAI Organization.

Younes, H. L. S.; and Simmons, R. G. 2003. VHPOP: Versatile Heuristic Partial Order Planner. *Journal of Artificial Intelligence Research* 20: 405–430.