Reasonable Landmark Orderings for Lifted Classical Planning

Anonymous submission

Abstract

Landmarks (LMs) are facts or actions contained in *every* solution to a planning problem. They are used to guide a search and are generated along with ordering relations defining in which order they need to be included. Most orderings come directly from the LM generation method, but so-called *reasonable orderings* (ROs) are generated in a post-processing. A RO $l_1 \xrightarrow{r} l_2$ describes that when achieving l_2 before l_1, l_2 needs to be achieved for a second time, and it is thus *reasonable* to achieve l_1 before l_2. Generation of RO comes with low computational costs, and in certain domains they increase performance of LM-based heuristics. However, existing RO generation methods need a *ground* model, and the recently introduced *lifted* LM techniques cannot use them. We present a RO generation method for *lifted* planning. We show that generation time is negligible even on hard-to-ground models, and that it helps to guide search empirically. Further, we investigate the theoretical connection to the present method.

1 Introduction

Landmarks (LMs) are elements – usually facts or actions – contained in *every* solution to a planning problem (Hoffmann, Porteous, and Sebastia 2004). They are generated along with ordering relations defining in which order they need to be included and have proven to be a valuable source of information when guiding a search. While some ordering relations are directly generated by the landmark generation method(s), the class of *reasonable orderings* (RO) (Koehler and Hoffmann 2000) is generated in a post-processing step. A reasonable ordering $l_1 \xrightarrow{r} l_2$ describes that – when the LM l_2 is achieved before l_1 – l_2 needs to be reached for a second time, and it is *reasonable* to reach l_1 before l_2. Reasonable orderings are not helpful in every domain, but in some, they increase the performance of LM-based heuristics. But their generation comes with low computational costs and thus they are used by state-of-the-art LM-based heuristics like the one of LAMA (Richter and Westphal 2010).

While the work on solvers of the last decades has mainly been based on *ground* models, recently, approaches have been introduced that avoid the (sometimes) expensive step of grounding (Ridder and Fox 2014; Corrêa et al. 2020, 2021, 2022; Lauer et al. 2021; Wichlacz, Höller, and Hoffmann 2021, 2022; Wichlacz et al. 2023) and solve planning problems directly on the lifted input model. One line of research on guiding a lifted search is based on landmarks (Wichlacz, Höller, and Hoffmann 2021, 2022). The while some steps of existing RO generators are also applicable on a lifted model, some need a *ground* representation and need to be replaced to work in lifted planning. Thus, LM-based heuristics in lifted planning cannot exploit reasonable ordering relations.

In this paper, we make the following contributions:

- We present a generation method for reasonable ordering in *lifted* planning.
- We investigate the theoretical connection to the present method.
- We show empirically that generation time is negligible even on large models, that the generated orderings can be helpful in lifted planning, and that our method is as good as existing ones in a direct comparison.

We next introduce the background in lifted planning, before we come to the generation methods, its theoretical properties, and the evaluation.

2 Background

A lifted planning problem is a tuple $\Pi = (\mathcal{P}, \mathcal{O}, \mathcal{A}, \mathcal{I}, \mathcal{G})$ where $\mathcal{P}$ is a set of (first-order) *predicates*, $\mathcal{O}$ is a set of *objects*, $\mathcal{A}$ is a set of *action schemas*, $\mathcal{I}$ is the *initial state*, and $\mathcal{G}$ is the *goal* definition. Predicates $P \in \mathcal{P}$ come with an *arity k*, i.e.P has k *parameters*, written $P(x_1, \dots, x_k)$, where each x_i is a variable. Let $\mathcal{X}$ be the set of all variables used in Π. Parameters can be *grounded* (instantiated) with objects $\mathcal{O}$. We write $P(u_1, \dots, u_k)$ to denote a partially grounded predicate along with its arguments $u_i \in \mathcal{X} \cup \mathcal{O}$. If $u_i \in \mathcal{O}$ for all i then $P(u_1, \dots, u_k)$ is a *ground atom* or *fact*, which we write as p. States are sets of ground atoms. We also assume $\mathcal{I}$ to be grounded. The goal also is a set of ground atoms.

An *action schema* A is a tuple $(X_A, pre(A), add(A), del(A))$ with a set of parameter variables X_A as well as *precondition*, *add list*, and *delete list*, all of which are sets of predicates parameterized with variables from X_A. The arity of A is $|X_A|$. We can instantiate action schemas by replacing each $x \in X_A$ by some $o \in \mathcal{O}$ to obtain *ground actions* a. The set of ground actions, or *actions* for short, is $\mathcal{A}^{\mathcal{O}}$.

Action a is applicable in state s if $pre(a) \subseteq s$. Applying a to s results in the state $(s \setminus del(a)) \cup add(a)$. A *plan* for Π is a sequence π of ground actions that is successively applicable in $\mathcal{I}$ and results in a state s' such that $\mathcal{G} \subseteq s'$.

A ground atom p is a *ground landmark* of Π if, for every plan π for Π, there exists a state s traversed by π where $p \in s$. A partially ground atom p is true in a state s, written $s \models p$, iff there is an instantiation p' of p with $p' \in s$.

For the following, consider the execution of a plan as a sequence $\pi = (s_0, a_1, s_1, a_2, \ldots, s_{n-1}, a_n, s_n)$ of actions a_i and states s_i resulting from executing the plan with $s_0 = \mathcal{I}$ and s_i the state resulting from the application of a_i in s_{i-1}.

Definition 1 (Lifted Landmarks) *A partially grounded atom $p = P(u_1, \ldots, u_k)$ is a* lifted landmark *of Π if and only if for every plan $\pi = (s_0, a_1, s_1, a_2, \ldots, s_{n-1}, a_n, s_n)$ for Π, there exists a state s_i where $s_i \models p$.*

Lifted LMs are a special case of disjunctive landmarks (a list of facts from which one must be in every solution), with a disjunction over possible values of unset parameters.

A landmark l_1 is added at time i iff $s_i \models l_1$ and $i = 0$ or $s_{i-1} \not\models l_1$. We write *first*(l_1, π) and *last*(l_1, π) for the first and last time that l_1 is added in a plan π. Let l_1 and l_2 be landmarks of Π, then:

- l_1 is *ordered naturally before* l_2 iff in the execution π of every plan for Π, *first*$(l_1, \pi) <$ *first*(l_2, π).
- l_1 is *ordered greedy-necessarily before* l_2 iff in the execution π of every plan for Π, $s_{\mathit{first}(l_2,\pi)-1} \models l_1$.
- l_1 is *ordered necessarily before* l_2 iff in the execution π of every plan for Π, whenever l_2 is added at time i, $s_{i-1} \models l_1$.

Note that necessary and greedy-necessary orderings are special cases of natural orderings.

There are three methods to generate LMs on the lifted model: One uses a back-chaining approach starting from state features in the goal (which are LMs). It intersects the preconditions of all actions adding these LMs, which results in a LM. This is continued until convergence (Wichlacz, Höller, and Hoffmann 2022). The second method uses FAM groups (Fišer 2020). FAM groups capture how values of a variable can change. Based on them, one can build a graph in which cuts between the current value of a variable and one that must be reached (e.g. a LM) must be crossed (Wichlacz, Höller, and Hoffmann 2022). The third method resembles the (ground) LM-Cut heuristic (Helmert and Domshlak 2009) in the lifted setting (Wichlacz et al. 2023).

Be aware that the LM generation is out of scope for this paper – we are aiming at generating reasonable orderings.

3 Generating Reasonable Orderings

Reasonable orderings have originally been introduced by Koehler and Hoffmann (2000) in the context of *goal* ordering, not *LM* ordering. The authors identified sufficient conditions to show that, when a goal fact l_2 is established before another l_1, l_2 is deleted while achieving l_1. This is e.g. the case when all actions adding l_1 also delete l_2 or have a (direct or indirect) precondition mutex with l_2. In such situations, the authors introduce an ordering $l_1 \xrightarrow{r} l_2$ meaning that it is *reasonable* to achieve l_1 before l_2.

Hoffmann, Porteous, and Sebastia (2004) adapted the concept from goals to LMs. Reasonable ordering relations have a slightly different meaning than other orderings used for LMs: while an ordering $l_1 \rightarrow l_2$ as introduced in Section 2 meant that it is not possible to achieve l_2 before l_1, now it might even be that, though $l_1 \xrightarrow{r} l_2$ holds, l_2 *is* actually achieved before l_1 in *every plan*. However, we know that in this case, l_2 needs to be deleted and *re*-achieved afterwards. This can be exploited in heuristics calculated based on the LMs (Richter and Westphal 2010; Büchner et al. 2023).

The basic definition is as follows (Hoffmann, Porteous, and Sebastia 2004, Lem. 1/Thm. 6):

Definition 2 (Reasonable Orderings on LMs) *Let l_1 and l_2 be LMs. There is a Reasonable Ordering $l_1 \xrightarrow{r} l_2$ if*

(i) l_2 is in the aftermath *of l_1 and*
(ii) l_1 interferes *with l_2.*

The "aftermath" relation says that, when there is a state in a solution where l_1 holds and l_2 has not been true yet, there must be a state in the remainder where l_1 and l_2 both hold simultaneously *or* l_2 holds at a later point during the execution of every plan (Hoffmann, Porteous, and Sebastia 2004, p. 222). While this property is hard to prove in general (2004, Thm. 4), the authors introduce an approximation (2004, Lem. 1) that we also use in the following:

Definition 3 (Aftermath Approximation) *Let l_1 and l_2 be LMs, l_2 is in the* aftermath *of l_1 if*

(i) l_2 is in the goal condition, or
(ii) there is a chain of natural orderings $l_1 = L_1 \rightarrow \ldots \rightarrow L_n$, $n > 1$, $L_{n-1} \neq l_2$ and $l_2 \overset{\text{gn}}{\rightarrow} L_n$.

Since this definition is based on LMs and their ordering relations, it is not only applicable in the grounded, but also in the lifted setting.

Next let us consider Condition (ii) of Def. 2 and define "interference". What is tested is whether two facts can appear together in a state. Hoffmann et al. call this property (in-)consistence, we will call two facts that cannot appear together to be *mutex*. The mutex relation is hard to compute exactly even in the ground setting. Therefore Hoffmann et al. used approximation methods, e.g. based on the mutex relation of the (non-relaxed) planning graph, which is still hard to compute – even on domains like those used in the International Planning Competition (IPC). Recent systems like Fast Downward (FD) compile a finite domain representation (FDR) in a preprocessing, which is (among other things) used in the RO computation. To detect more cases where two facts cannot appear together, interference further checks if the actions adding l_1 necessarily delete l_2.

Since most recent systems use a FDR, we also define interfere based on FDR models (the same condition is also used by FD):

Definition 4 (Interference) *A fact l_1* interferes *with l_2 if*

1. *l_1 and l_2 are mutex, or*
2. *the add lists of all actions adding l_1 also contain some fact x which is mutex with l_2.*

The original definition for a non-FDR model was introduced by Hoffmann, Porteous, and Sebastia (2004, Def. 6).

3.1 Lifted RO Inference

While the aftermath approximation can also be used in the lifted setting, we need to come up with a different method to approximate the mutex relation. Our method is based on *Lifted Fact-Alternating Mutex* (FAM) groups as introduced by Fišer (2020).

Definition 5 (FAM Groups) *A Lifted Fact-Alternating Mutex (FAM) group is a tuple $\nu = (\mathcal{V}^{fix}, \mathcal{V}^{cnt}, atoms)$ with $\mathcal{V}^{fix} \cap \mathcal{V}^{cnt} = \emptyset$. For the set of atoms $atoms = \{P_1(v_1^1, \ldots, v_{k_1}^1), \ldots, P_l(v_1^l, \ldots, v_{k_l}^l)\}$, all variables are from $\mathcal{V}^{fix} \cup \mathcal{V}^{cnt}$.*

For a given assignment of fixed variables, the full grounding of the counted variables forms a mutex group.

Consider e.g. a FAM group with $\mathcal{V}^{fix} = \{p\}$, $\mathcal{V}^{cnt} = \{t, l\}$, $atoms = \{in(p,t), at(p,l)\}$, where p is a package, t is a truck, and l a location. in states that a package is in a truck and at that it is at some location. Then the semantics is that for a given package, all assignments to truck and location are mutex to each other, or: a package can only be at a single location or in a single truck.

In the lifted setting, we need to ensure that the mutex relation holds for all groundings of an atom, i.e., two partially grounded atoms p and q are mutex iff all of its groundings are pairwise mutex. I.e., they need to be part of the same mutex group, but with different counted variables. When they are in different mutex groups (two different trucks in the example), they can occur together in a state.

Definition 6 (Lifted Mutex Relation) *Let $P(v_1, \ldots, v_n)$ and $Q(w_1, \ldots, w_m)$ be two partially grounded atoms. There is a* lifted mutex relation *between P and Q when the following conditions hold:*

1. *They are not unifiable, i.e., there is no assignment of the variables that make them equal.*
2. *There exists a lifted FAM group g including P and Q such that for every assignment of fixed variables, they are in the same ground mutex group.*

Theorem 1 (Soundness) *When the condition in Def. 6 holds, the atoms are mutex.*

Proof: From the definition of FAM groups, we know that two facts are mutex when they share the fixed variables and differ in the counted variables. Since we want to show that all groundings of the partial lifted atom are mutex, the existence of a single assignment of a fixed variable suffices to make them potentially non-mutex. However, condition 2 ensures that the atoms need to end up in the same mutex group. Given that the two facts are in the same mutex group, we know that they cannot occur in the same state (i.e., are mutex) except when they are unifiable, since then, we can unify them and obtain a single ground fact. ■

Having a lifted definition of the mutex relation at hand, we can now define lifted interference. We do this based on the condition by Koehler and Hoffmann (2000):

Definition 7 (Lifted Interference) *Two partially grounded facts l_1 and l_2 interfere if one of the following holds:*

1. *l_1 and l_2 are mutex.*
2. *When for all partial instantiations of action schemas that add l_1 one of the following criteria holds:*
   (a) *one of its preconditions is mutex with l_2, or*
   (b) *one of its add effects is mutex with l_2.*

Now we have a method directly applicable on the lifted model. It inherits the incompleteness of the ground approaches, e.g. from the aftermath approximation. But further it adds more sources of incompleteness through its mutex relation. Consider e.g. that two partially ground atoms might be unifiable in our test, but no state where such a unification is possible is reachable.

Next we want to investigate the theoretical connection between our method and the one used by FD. The aftermath approximation is the same for both methods, but they differ in the interference and mutex relations. For this theoretical analysis, assume that we deal with a fully grounded problem (otherwise, FD's method is not applicable).

Interference Relation. First we compare the interference relations defined in Definition 4 and Definition 7. Since the latter is done on a ground model, for the first case of both definitions, it holds that the method with the stronger mutex relation might result in more reasonable orderings that are detected. The second condition of FD checks all actions adding the first atom and whether one of its effects is mutex with the second. Our method further takes the preconditions into account as suggested by (Koehler and Hoffmann 2000). Again, whether we can find more orderings depends on the mutex relation.

Mutex Relation. The inference of mutex groups by Fišer used here is an extension of the one used by FD. In addition to the latter, it incorporates subtypes into its reasoning, which is not done by FD (Fišer 2020, 9839f). I.e., on a ground model, it will find all mutex groups found by FD and it might find more (Fišer 2020).

This leads us to the following proposition:

Proposition 1 *On a ground model, our method finds at least the reasonable ordering relations as FD's method, but may find more.*

Be aware that this is a theoretical comparison, we are not aiming at using our method on a pre-ground model, but want to integrate it into a lifted planning system. Further, we have no evidence that this case is included in commonly-used planning benchmarks.

4 Evaluation

In this section, we investigate two objectives:

1. Can lifted search benefit from reasonable ordering relations?
2. How does our method compare to present ones?

We integrated[1] our method into two systems: (1) the lifted Powerlifted (PWL) system (Corrêa et al. 2020), and (2) the grounded Fast Downward system (Helmert 2006). All experiments ran on Intel Xeon E5-2650 CPUs with 2.30 GHz with a time limit of 30 min and 4 GB memory limit.

[1]We will make our code available when the paper gets accepted.

Coverage	LMC, no RO	LMC with RO	LAMA, no RO	LAMA with RO
blocksworld (40)	0	**11**	20	**36**
childsnack (144)	22	22	87	87
ged (312)	312	312	311	311
logistics (40)	15	15	40	40
org.-synthesis (56)	46	46	47	47
pipesworld (50)	22	22	29	29
rovers (40)	7	7	37	37
visitall (180)	63	63	**136**	135
Sum (862)	487	**498**	707	**722**

Figure 1: Coverage on the HTG domains.

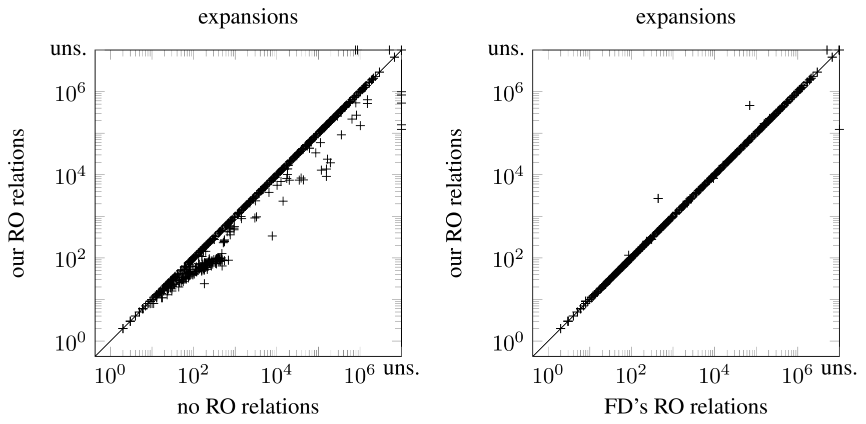

Figure 2: Expansions of the FD system in the ground setting.

Lifted Setting In this setting, we use the hard to ground (HTG) benchmarks[2] also applied in recent evaluations of lifted systems (e.g. Corrêa et al. (2021); Lauer et al. (2021); Höller and Behnke (2022)). We integrated our technique into the PWL-based planner introduced by Wichlacz, Höller, and Hoffmann (2022) that comes with an LM count heuristic (LMC) and also a LAMA-like (Richter and Westphal 2010) configuration, which we also use as baseline. We test two configurations: greedy best-first search (GBFS) with LMC, and the LAMA-like multi-fringe configuration that uses GBFS, in one fringe with the lifted h^{add} heuristic (Bonet and Geffner 2001; Corrêa et al. 2021), in a second fringe with the LMC heuristic. For both configurations we compare the results with and without ROs.

Our first result is that the runtime of our RO inference method is neglectable: the mean runtime is 0.0 seconds for every domain. The maximum runtime is 0.58 seconds for an organic synthesis problem. However, even in this domain, the mean runtime is 0.04 seconds across all instances.

The coverage results on the HTG benchmarks are given in Table 1. The first two columns give coverage results for the GBFS/LMC configuration, with and without RO relations. It can be seen that it improves the coverage in the *Blocksworld* domain, which is known to be a domain where such orderings are helpful. The following columns show the lifted LAMA system, again with an improvement in *Blocksworld*.

We do not find ROs in the other domains. This is due to two reasons. First, the benchmark set contains only few domains with ROs. Second, the present LM techniques are less successful in finding LMs than those from ground planning. On groundable instances, FD finds additional ROs on *Organic Synthesis* and *Logistics*. *Logistics* is an example where we cannot find ROs because a certain LM found on the grounded model is not returned by the lifted LM generation. However, our method will benefit from future developments in this direction. But while we were only able to improve coverage in one domain, we can see that our method works fast even on large models.

[2]https://github.com/abcorrea/htg-domains

Grounded Setting To test our method in a setting with more domains where RO can be found, we integrated our ordering relations into the Fast Downward (FD) planning system. We use the preprocessing and LM generation of FD. Then, we call our method with *the lifted model* and these LMs. We use the resulting ordering relations in the search of FD to have a direct comparison. We use the benchmark sets from the IPCs 1998 to 2018. We first evaluated satisfying configurations on the respective benchmarks. But using LAMA (coverage between 915 and 917 out of 1001 for all configurations) and GBFS with LMC (coverage between 766 and 768), there was not much to find in the results when comparing configurations without RO, with FD's RO, or ours. Instead, we combined an A^* search with an admissible LMC heuristic (Büchner et al. 2023), evaluated them on the benchmarks from the optimal track, and show results on the expansions instead of coverage. The results are shown in Figure 2, the left side compares a configuration using our ROs and a configuration not using ROs. It can be seen that there are instances where ROs decrease the number of expansions (mainly in the domains *Miconic*, *Logistics00*, and *Blocks*). The right side of Figure 2 gives a comparison of the configuration using our ROs and those generated by FD. The results are nearly identical, the two outlier instances are from the *Scanalyzer* domain, where we do not find ROs on the lifted model that are found on the ground model.

5 Conclusion

A reasonable ordering $l_1 \overset{r}{\to} l_2$ means that, when achieving LM l_2 before l_1, it needs to be achieved for a second time. This information is used to compute more accurate heuristic values. However, inferring ROs was so far only possible on the ground model. We presented an inference method for lifted planning. Empirically, we show on a benchmark set for lifted planning that (1) it is fast to compute even on large models and that (2) there are problems where it increases coverage. In a second evaluation directly comparing it with FD's (ground) method, we show that search guidance is comparable, despite the fact that our method ran on the lifted model, while FD's ran on the ground one.

References

Bonet, B.; and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence*, 129(1): 5–33.

Büchner, C.; Keller, T.; Eriksson, S.; and Helmert, M. 2023. Landmark Progression in Heuristic Search. In *Proceedings of the 33rd International Conference on Automated Planning and Scheduling (ICAPS)*, 70–79. AAAI Press.

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)*, 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)*, 94–102. AAAI Press.

Corrêa, A. B.; Pommerening, F.; Helmert, M.; and Francès, G. 2022. The FF Heuristic for Lifted Classical Planning. In *36th AAAI Conference on Artificial Intelligence (AAAI)*, 9716–9723. AAAI Press.

Fišer, D. 2020. Lifted Fact-Alternating Mutex Groups and Pruned Grounding of Classical Planning Problems. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, 9835–9842. AAAI Press.

Helmert, M. 2006. The Fast Downward Planning System. *Journal of Artificial Intelligence Research (JAIR)*, 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)*. AAAI Press.

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

Höller, D.; and Behnke, G. 2022. Encoding Lifted Classical Planning in Propositional Logic. In *Proceedings of the 32nd International Conference on Automated Planning and Scheduling (ICAPS)*, 134–144. AAAI 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 (JAIR)*, 12: 338–386.

Lauer, P.; Torralba, Á.; Fišer, 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)*, 4119–4126. IJCAI organization.

Richter, S.; and Westphal, M. 2010. The LAMA Planner: Guiding Cost-Based Anytime Planning with Landmarks. *Journal of Artificial Intelligence Research (JAIR)*, 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)*, 244–252. AAAI Press.

Wichlacz, J.; Höller, D.; Fišer, D.; and Hoffmann, J. 2023. A Landmark-Cut Heuristic for Lifted Optimal Planning. In *Proceedings of the 26th European Conference on Artificial Intelligence (ECAI)*. IOS Press.

Wichlacz, J.; Höller, D.; and Hoffmann, J. 2021. Landmark Heuristics for Lifted Planning – Extended Abstract. In *Proceedings of the 14th International Symposium on Combinatorial Search (SoCS)*, 242–244. AAAI Press.

Wichlacz, J.; Höller, D.; and Hoffmann, J. 2022. Landmark Heuristics for Lifted Classical Planning. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, 4665–4671. IJCAI organization.