Landmark Progression in Heuristic Search

Clemens Büchner[1], **Thomas Keller**[1,2], **Salomé Eriksson**[1], **Malte Helmert**[1]

[1]University of Basel, Switzerland
[2]University of Zürich, Switzerland
{clemens.buechner,tho.keller,salome.eriksson,malte.helmert}@unibas.ch

Abstract

The computation of high-quality landmarks and orderings for heuristic state-space search is often prohibitively expensive to be performed in every generated state. Computing information only for the initial state and progressing it from every state to its successors is a successful alternative, exploited for example in classical planning by the LAMA planner. We propose a general framework for using landmarks in any kind of best-first search. Its core component, the progression function, uses orderings and search history to determine which landmarks must still be achieved. We show that the progression function that is used in LAMA infers invalid information in the presence of reasonable orderings. We define a sound progression function that allows to exploit reasonable orderings in cost-optimal planning and show empirically that our new progression function is beneficial both in satisficing and optimal planning.

Introduction

Heuristic search is a widely applied approach to finding goal paths in large transition systems. *Landmarks* denote properties that hold in every solution. Furthermore, *landmark orderings* are a guaranteed pattern between time points at which two landmarks are satisfied. Landmarks and landmark orderings are widely used sources for the computation of heuristics (e.g., Hoffmann, Porteous, and Sebastia 2004; Richter, Helmert, and Westphal 2008; Helmert and Domshlak 2009; Bonet and Helmert 2010; Büchner, Keller, and Helmert 2021).

The generation of landmarks and orderings for state-space search problems is often too expensive to be performed for every generated state. An alternative is to compute landmarks and orderings only for the initial state and to *progress* information from a state to its generated successors according to the following principle: A property that holds on all goal paths for a state s also holds for all goal paths for a successor s' of s unless it is satisfied by the transition from s to s'. *Classical planning* is a research area where this kind of reasoning has been applied successfully. The LAMA planner (Richter and Westphal 2010) computes landmarks and landmark orderings only for the initial state and uses information from *greedy-necessary* and *reasonable* orderings to infer landmarks that are required again. Since LAMA uses suboptimal search algorithms and an inadmissible heuristic, it does not come with any guarantee with respect to solution quality.

The closely related LM-A* algorithm (Karpas and Domshlak 2009) guarantees optimality by performing a multi-path dependent heuristic search. It is guided by a heuristic that computes a cost partitioning over landmarks and it progresses information from greedy-necessary orderings. We show that LM-A* ignores reasonable orderings for good reason: there are cases where LAMA incorrectly infers a landmark as required again. While this does not affect the satisficing LAMA planner, it makes admissible landmark heuristics inadmissible and in turn leads to a loss of the optimality guarantee of LM-A*. Buffet and Hoffmann (2010) observe that LAMA progression is too pessimistic in the presence of chains of reasonable orderings; a single operator may achieve all involved landmarks at once, but this is not captured by the progression used in LAMA. Buffet and Hoffmann propose an improved progression function for LAMA, but we show that it is not sufficient to make LAMA progression sound.

To be able to *prove* soundness of progression functions, we formalize landmark progression via *landmark states* and introduce the best-first search framework LM-BFS. We propose new progression functions for reasonable, *natural*, *necessary* (Hoffmann, Porteous, and Sebastia 2004), and *weak* orderings (Büchner, Keller, and Helmert 2021) and formally prove that these progression functions as well as the progression function for greedy-necessary orderings (Richter and Westphal 2010) infer valid information.

Finally, we evaluate several progression functions in classical planning. First, we test the impact of using valid progression in the satisficing LAMA planner (Richter and Westphal 2010). We observe slight improvements in terms of plan quality without negative impact from using the valid but more expensive progression functions. With valid progression of reasonable orderings, we can for the first time also use these orderings for cost-optimal planning. We evaluate their impact on the admissible landmark heuristic (Karpas and Domshlak 2009) as well as the cyclic landmark heuristics (Büchner, Keller, and Helmert 2021) and show that it is beneficial to consider as many orderings as possible in the progression.

Background

While related work mainly considers *classical planning*, our contributions can be applied to any kind of search over a deterministic transition system.

Transition System A *transition system* or *state space* is a 6-tuple $\mathcal{T} = \langle S, A, T, s_I, S^*, \mathit{cost}\rangle$ where S is a set of *states*; A is a set of labels called *actions*; $T \subseteq S \times A \times S$ is a set of labeled transitions; $s_I \in S$ is the *initial state*; $S^* \subseteq S$ is the set of *goal states*; and $\mathit{cost} : A \to \mathbb{R}_0^+$ is the *cost function*. We further consider a set of *state features* $\mathcal{F}$, each of which is either true or false in any given state. The state spaces we consider are deterministic in the sense that there is at most one transition $\langle s, a, s'\rangle$ for each state-action-pair $\langle s, a\rangle \in S \times A$. We say an action a is *applicable in* s if there exists a transition $\langle s, a, s'\rangle$.

Assume in the following that $t_i = \langle s_{i-1}, a_i, s_i\rangle$. An *action sequence* $\langle a_1, \dots, a_n\rangle$ is applicable in state $s = s_0$ iff $t_i \in T$ for $1 \leq i \leq n$. The corresponding *trajectory* is denoted by $\pi = \langle t_1, \dots, t_n\rangle$ and has length $|\pi| = n$. Given two trajectories $\pi = \langle t_1, \dots, t_n\rangle$ and $\pi' = \langle t'_1, \dots, t'_m\rangle$ such that $s_n = s'_0$ we write $\pi \circ \pi'$ to denote their concatenation $\langle t_1, \dots, t_n, t'_1, \dots, t'_m\rangle$. For states s and s', $\mathcal{P}(s, s')$ is the set of all trajectories $\pi = \langle t_1, \dots, t_n\rangle$ with $s_0 = s$ and $s_n = s'$. An *s-plan* is a trajectory $\langle t_1, \dots, t_n\rangle$ such that $s_0 = s$ and $s_n \in S^*$, and we denote the set of all s-plans with $\mathcal{P}(s) = \bigcup_{s' \in S^*} \mathcal{P}(s, s')$. The *cost* of π is the sum of the action costs in the trajectory: $\mathit{cost}(\pi) = \sum_{i=1}^{n} \mathit{cost}(a_i)$. An s-plan is *optimal* if it has minimal cost among all s-plans.

Landmarks There are different notions of landmarks in the planning literature. On the one hand, there is work that describes landmarks as (possibly restricted) propositional formulas over the *atoms* of the planning task (e.g., Hoffmann, Porteous, and Sebastia 2004; Keyder, Richter, and Helmert 2010; Richter and Westphal 2010); and on the other hand, there is work that deals with landmarks that are actions or sets of actions representing *disjunctive action landmarks* (e.g., Helmert and Domshlak 2009; Keyder, Richter, and Helmert 2010; Büchner, Keller, and Helmert 2021). This work only considers the former case.

Let φ be a formula over $\mathcal{F}$. We say that φ *holds* in state s iff $s \models \varphi$, and we write $\pi \models \varphi$ to denote that φ holds in some state s_i of the trajectory $\pi = \langle t_1, \dots, t_n\rangle$. We say φ is a *landmark for a state* s iff $\pi \models \varphi$ for every s-plan $\pi \in \mathcal{P}(s)$. Moreover, we say φ is a *landmark* iff it is a landmark for s_I, and it is a *landmark between* s and another state s' iff $\pi \models \varphi$ for every $\pi \in \mathcal{P}(s, s')$.

A landmark L is *added at time* i by trajectory $\pi = \langle t_1, \dots, t_n\rangle$ iff $s_i \models L$ and (1) $i = 0$, or (2) $1 \leq i \leq n$ and $s_{i-1} \not\models L$. We say a landmark L is *first added by* π *at time* i, denoted $\mathit{first}(L, \pi) = i$, if it is added by π at time i and not added by π at a time $j < i$; and it is *last added by* π *at time* i, denoted $\mathit{last}(L, \pi) = i$, if it is added by π at time i and not added by π at a time $i < j \leq n$. If L is not added by π at any time $i \in \{0, \dots, n\}$, we say $\mathit{first}(L, \pi) = \mathit{last}(L, \pi) = \perp$. (For any $n \in \mathbb{N}_0$, it neither holds that $\perp < n$ nor $\perp > n$.)

Landmark Orderings Hoffmann, Porteous, and Sebastia (2004) introduce different kinds of temporal dependencies between two landmarks called *landmark orderings*, and Büchner, Keller, and Helmert (2021) add another two ordering types. Two of their definitions – called *natural ordering* by the former and *strong ordering* by the latter – are equivalent but defined differently. We use the terminology of Hoffmann, Porteous, and Sebastia but the definition of Büchner, Keller, and Helmert here and say that there is a *natural ordering* $A \to_{\mathrm{n}} B$ *for a state* s iff $\mathit{first}(A, \pi) < \mathit{first}(B, \pi)$ for all $\pi \in \mathcal{P}(s)$. Two special kinds of natural orderings impose stricter constraints on the ordering relation of landmarks. A *greedy-necessary ordering* $A \to_{\mathrm{gn}} B$ for state s denotes that A holds when B is achieved for the first time in every plan, i.e., $s_{\mathit{first}(B,\pi)-1} \models A$ for all $\pi \in \mathcal{P}(s)$. A *necessary ordering* $A \to_{\mathrm{nec}} B$ for s is even more restrictive and requires that A always holds when B is added, i.e., $s_{i-1} \models A$ for all $\pi \in \mathcal{P}(s)$ and all times $i > 0$ at which B is added in π (not only the first time).

Hoffmann, Porteous, and Sebastia further describe *reasonable orderings*. Intuitively, a reasonable ordering $A \to_{\mathrm{r}} B$ denotes that adding B before A requires that B is removed again as it is impossible to add A while B holds, but B must be added again after A. Hence, it would be *reasonable* to only add B after A. We obtain a formal definition by splitting the definition of Richter and Westphal (2010) into three separate cases. There is a reasonable ordering $A \to_{\mathrm{r}} B$ for state s iff every $\pi = \langle t_1, \dots, t_n\rangle \in \mathcal{P}(s)$ satisfies any one of the following conditions:

(i) $\mathit{first}(A, \pi) \leq \mathit{first}(B, \pi)$;

(ii) $s_{\mathit{first}(A,\pi)-1} \not\models B$ and there exists an i such that $\mathit{first}(A, \pi) \leq i \leq n$ and $s_i \models B$; or

(iii) $s_{\mathit{first}(A,\pi)-1} \models B$ and $s_{\mathit{first}(A,\pi)} \not\models B$ and there exists an i such that $\mathit{first}(A, \pi) < i \leq n$ and $s_i \models B$.

There has been some confusion regarding the soundness of reasonable orderings (e.g., Hoffmann, Porteous, and Sebastia 2004; Richter and Westphal 2010) that stems from the fact that a reasonable ordering $A \to_{\mathrm{r}} B$ does not imply that B cannot be reached before A. While this means that methods that require that landmarks are reached in some given order cannot exploit reasonable orderings, they still provide valuable information. Büchner, Keller, and Helmert (2021) exploit the observation that it can often not be avoided to reach B before A to improve landmark heuristics by considering cyclic dependencies between landmarks. They also introduce the similar yet simpler *weak orderings* $A \to_{\mathrm{w}} B$ for state s which require that $\mathit{first}(A, \pi) < \mathit{last}(B, \pi)$ for all s-plans $\pi \in \mathcal{P}(s)$. Weak orderings would be a generalization of reasonable orderings if it were not for the fact that reasonable orderings can be satisfied if A and B are reached simultaneously and weak orderings cannot.

Landmark Best-First Search

Progression based systems for state space search compute information at the beginning of the search and *progress* it from each state to its successors. We are aware of three systems that apply this idea in classical planning: LM-A* (Karpas and Domshlak 2009) computes landmarks for the initial state and tracks for each state encountered during search which landmarks still need to be achieved; the LAMA planner (Richter and Westphal 2010) uses a similar approach and additionally exploits *multi-queue* heuristic search, *preferred operators*, and *anytime search* to improve sub-optimal solutions over time; and LTL-A* (Simon and Röger 2015) reasons about linear temporal logic (LTL) formulas as a generalization of landmarks. These systems have in common that their heuristics evaluate a state in combination with information (i.e, landmarks or LTL formulas) dependent on the previously expanded path(s) to that state.

We propose *landmark best-first search* (LM-BFS), a generalization of LM-A*. It is a multi-path dependent best-first search algorithm with four generic components: It computes landmarks and orderings for the initial state s_I, progresses landmark information along expanded paths, merges landmark information obtained on different paths to a state, and uses a landmark heuristic for guidance. For the remainder of this paper we denote by $\mathcal{G}_I = \langle \mathcal{L}_I, \mathcal{O}_I \rangle$ a *landmark graph* for the initial state s_I. Its nodes are the landmarks $\mathcal{L}_I$ and its directed edges are the orderings $\mathcal{O}_I$ between the landmarks. Our framework is based on the notions of *past* and *future* landmarks of a state s:

$$\mathcal{L}^*_{past}(s) = \{L \in \mathcal{L}_I \mid \pi \models L \text{ for all } \pi \in \mathcal{P}(s_I, s)\}$$
$$\mathcal{L}^*_{fut}(s) = \{L \in \mathcal{L}_I \mid last(L, \pi) > 0 \text{ for all } \pi \in \mathcal{P}(s)\}$$

Note that a landmark can simultaneously be a past and future landmark for a state s.

Knowing which landmarks must be achieved in the future of s is essential for computing heuristic estimates, while knowing which landmarks have been achieved already is important to infer which orderings are still relevant. In practice, $\mathcal{L}^*_{past}$ and $\mathcal{L}^*_{fut}$ are usually unknown because it is infeasible to expand all paths between two states, and we instead approximate them. More specifically, we underestimate $\mathcal{L}^*_{fut}$ such that a heuristic will not wrongly assume a specific landmark still needs to be achieved, which could render it inadmissible. For $\mathcal{L}^*_{past}$ on the other hand an overapproximation is desirable because otherwise we might wrongly deem an ordering as still relevant. The information about past and future landmarks is stored in the so-called *landmark state*.

Definition 1. ***Landmark state***
A landmark state $\mathbb{L}$ *is* $\dashv$ *or a tuple* $\langle \mathcal{L}_{past}, \mathcal{L}_{fut} \rangle$ *such that* $\mathcal{L}_{past} \cup \mathcal{L}_{fut} = \mathcal{L}_I$. $\mathcal{L}_{past}$ *denotes the* past landmarks of $\mathbb{L}$, $\mathcal{L}_{fut}$ *denotes the* future landmarks of $\mathbb{L}$, *and* $\mathbb{L} = \dashv$ *denotes conflicting information. We say* $\mathbb{L}$ *is* valid in s *iff*

- $\mathbb{L} = \dashv$ *and* $\mathcal{P}(s) = \emptyset$, *or*
- $\mathbb{L} \neq \dashv$ *and* $\mathcal{L}_{past} \supseteq \mathcal{L}^*_{past}(s)$ *and* $\mathcal{L}_{fut} \subseteq \mathcal{L}^*_{fut}(s)$.

Validity describes the intended semantics of a landmark state: $\mathcal{L}_{past}$ contains at least all landmarks between s_I and s, and $\mathcal{L}_{fut}$ contains at most all landmarks added on all s-plans. The special landmark state $\dashv$ signals that conflicting information detects state s as a dead-end for which no s-plan can exist. We want to distinguish this case from $\langle \mathcal{L}_{past}, \mathcal{L}_{fut} \rangle$ explicitly because there is no natural way to express this with $\mathcal{L}_{past}$ and $\mathcal{L}_{fut}$; setting both to $\mathcal{L}_I$ might be valid in states with s-plans and leaving both empty is valid if $\mathcal{L}_I = \emptyset$.

Algorithm 1: The LM-BFS framework: a generic best-first search with reopening.

1 $\langle \mathcal{L}_I, \mathcal{O}_I \rangle :=$ compute_landmark_graph(s_I)
2 **foreach** $s \in S$ **do** $\mathbb{L}(s) := \langle \mathcal{L}_I, \emptyset \rangle$;
3 $\mathbb{L}(s_I) :=$ progress$(\langle \emptyset, \mathcal{L}_I \rangle, \langle \cdot, init, s_I \rangle)$
4 **if** $h(s_I, \mathbb{L}(s_I)) < \infty$ **then**
5   *open*.insert$(\langle s_I, 0, h(s_I, \mathbb{L}(s_I)) \rangle)$
6 **while not** *open*.empty() **do**
7     $\langle s, g, v \rangle =$ *open*.pop()
8   **if** $v < h(s, \mathbb{L}(s))$ **then**
9   *open*.insert$(\langle s, g, h(s, \mathbb{L}(s)) \rangle)$
10   **else if** $g < distances(s)$ **then**
11         $distances(s) := g$
12   **if** is_goal(s) **then**
13   **return** extract_plan(s)
14   **foreach** $\langle s, a, s' \rangle \in T$ **do**
15             $\mathbb{L} :=$ progress$(\mathbb{L}(s), \langle s, a, s' \rangle)$
16             $\mathbb{L}(s') :=$ merge$(\{\mathbb{L}(s'), \mathbb{L}\})$
17   **if** $\mathbb{L}(s') \neq \dashv$ **and** $h(s', \mathbb{L}(s')) < \infty$ **then**
18   *open*.insert$(\langle s', g + cost(a), h(s', \mathbb{L}(s')) \rangle)$
19 **return** unsolvable

Algorithm 1 outlines the LM-BFS framework. It computes a landmark graph for the initial state of a transition system (line 1). For each state, it maintains a landmark state initialized to consider all landmarks past and none future which is always valid (line 2).[1] LM-BFS then progresses the landmark state where all landmarks are considered future and none are past through an artificial *init*-transition to obtain a landmark state for the initial state s_I (line 3). Whenever a state s is retrieved from the open list, its heuristic value is re-evaluated (line 8) because $\mathbb{L}(s)$ might have changed since s was added to *open* due to additional paths to s. If the re-evaluation shows an increased heuristic value of s, s is inserted back into *open* and the next state is retrieved. Otherwise, s is expanded, $\mathbb{L}(s)$ is progressed to each successor state s' (line 15), and merged with previous information about s' (line 16).

A concrete implementation of our framework needs to specify the four central components. In this work we assume that the landmark generation method and the heuristic are both given, and point to the literature for a detailed discussion (e.g., Zhu and Givan 2003; Richter, Helmert, and Westphal 2008; Keyder, Richter, and Helmert 2010; Domshlak et al. 2011; Karpas and Domshlak 2009; Büchner, Keller, and Helmert 2021). Using existing heuristics might seem problematic at first because they do not operate on landmark states, but it is possible to reproduce all required information from it. Landmark progression was introduced by Richter, Helmert, and Westphal (2008) to use the landmark information of the initial state in all reachable successors. Karpas and Domshlak (2009) also use landmark progression in their LM-A* search algorithm where they further propose merging landmark information from multiple paths, which we adapt as follows.

[1] In our implementation, $\mathbb{L}(s)$ is not initialized until s is generated for the first time.

Definition 2. ***Merging function***
Let $S_{\mathbb{L}} = \{\mathbb{L}^1, \ldots, \mathbb{L}^n\}$ be a non-empty set of landmark states over the same landmark graph $\mathcal{G}_I$, where $\mathbb{L}^i = \bot$ or $\mathbb{L}^i = \left\langle \mathcal{L}^i_{past}, \mathcal{L}^i_{fut} \right\rangle$ for $1 \leq i \leq n$. The (landmark state) merging function merge *maps a set of landmark states to a landmark state as follows:*

$$\text{merge}(S_{\mathbb{L}}) = \begin{cases} \bot & \text{if } \bot \in S_{\mathbb{L}} \\ \left\langle \bigcap_{i=1}^n \mathcal{L}^i_{past}, \bigcup_{i=1}^n \mathcal{L}^i_{fut} \right\rangle & \text{otherwise.} \end{cases}$$

We observe that merging preserves validity, i.e., if all landmark states that are merged are valid in a state s, then so is the resulting landmark state.

Theorem 1. *Let $\mathbb{L}^1, \ldots, \mathbb{L}^n$ be landmark states that are valid in s. Then $\mathbb{L} = \text{merge}(\{\mathbb{L}^1, \ldots, \mathbb{L}^n\})$ is valid in s.*

Proof. If at least one $\mathbb{L}^i = \bot$, then $\mathbb{L} = \bot$, and validity of $\mathbb{L}$ follows directly from validity of $\mathbb{L}^i$. Otherwise we need to show $\bigcap_{i=1}^n \mathcal{L}^i_{past} \supseteq \mathcal{L}^*_{past}(s)$ and $\bigcup_{i=1}^n \mathcal{L}^i_{fut} \subseteq \mathcal{L}^*_{fut}(s)$. Since we know all $\mathbb{L}^i$ are valid in s we have $\mathcal{L}^i_{past} \supseteq \mathcal{L}^*_{past}(s)$ and $\mathcal{L}^i_{fut} \subseteq \mathcal{L}^*_{fut}(s)$ from which the claim follows directly. □

Progression

We now turn our attention to the progression of landmark states, the last remaining component of LM-BFS. In this section, we formally define different progressions from the literature in terms of a *progression function*.

Definition 3. ***Progression function***
A (landmark state) progression function prog *maps a landmark state $\mathbb{L}$ and a transition $t = \langle s, a, s' \rangle$ to $\bot$ if $\mathbb{L} = \bot$ and to $\mathbb{L}'$ otherwise. We say that* prog *is* valid *iff $\mathbb{L}$ is valid in s implies that $\text{prog}(\mathbb{L}, t)$ is valid in s'.*

The case $\mathbb{L} \neq \bot$ is defined differently in the upcoming definitions on different progression functions. For the common case, it is easy to see that mapping $\mathbb{L} = \bot$ and a transition $t = \langle s, a, s' \rangle$ to $\bot$ is valid: $\mathcal{P}(s) = \emptyset$ follows from validity of $\bot$ in s. Hence, assuming $\mathcal{P}(s') \neq \emptyset$ leads to a contradiction because otherwise we can choose such an s'-plan $\pi \in \mathcal{P}(s')$ and construct $\langle t \rangle \circ \pi$ to obtain an s-plan.

With this, proving validity of a progression function prog requires showing the following properties for $\mathbb{L}' = \text{prog}(\mathbb{L}, \langle s, a, s' \rangle)$: (1) if $\mathbb{L}' = \bot$ then $\mathcal{P}(s') = \emptyset$ and (2) if $\mathbb{L}' = \left\langle \mathcal{L}'_{past}, \mathcal{L}'_{fut} \right\rangle \neq \bot$, then (a) $\mathcal{L}'_{past} \supseteq \mathcal{L}^*_{past}(s')$ and (b) $\mathcal{L}'_{fut} \subseteq \mathcal{L}^*_{fut}(s')$. We will write (1), (2a), and (2b) to reference these cases in the following proofs.

We start by showing that the progression function used in LM-A* (Karpas and Domshlak 2009) is valid. To do this, we first introduce several simpler progression functions for which we show validity independently and then combine (i.e., merge) them to $\text{prog}_{\text{LM-A}^*}$.

The first such function is *basic progression* over $\mathbb{L}$ and $t = \langle s, a, s' \rangle$. It puts all landmarks that hold in s' into $\mathcal{L}_{past}$ and it removes all landmarks that are added by t (i.e., $s \not\models L$ and $s' \models L$) from $\mathcal{L}_{fut}$. The former is because landmarks that hold in s' are trivially landmarks between s_I and s'. For the latter, the intuition is that the landmark is added by t, so the requirement for it to be added in the future of s is satisfied and we therefore assume it does not have to be added in the future of s' anymore.

Definition 4. ***Basic progression***
The function prog_{basic} maps landmark state $\mathbb{L} = \langle \mathcal{L}_{past}, \mathcal{L}_{fut} \rangle$ and transition $t = \langle s, a, s' \rangle$ to $\mathbb{L}' = \langle \mathcal{L}_{past} \cup \mathcal{L}_{hold}, \mathcal{L}_{fut} \setminus \mathcal{L}_{add} \rangle$ where $\mathcal{L}_{hold} = \{L \in \mathcal{L}_I \mid s' \models L\}$ and $\mathcal{L}_{add} = \{L \in \mathcal{L}_I \mid s \not\models L \text{ and } s' \models L\}$.

Theorem 2. *The progression function prog_{basic} is valid.*

Proof. **(1)** $\mathbb{L} = \bot$ is the only such case.

(2a) $\mathcal{L}_{past} \cup \mathcal{L}_{hold} \supseteq \mathcal{L}^*_{past}(s')$: For any $L \in \mathcal{L}^*_{past}(s')$ we have that $\pi \circ \langle t \rangle \models L$ holds for all $\pi \circ \langle t \rangle \in \mathcal{P}(s_I, s')$ and hence $\pi \models L$ or $s' \models L$. If $\pi \models L$, we know that $L \in \mathcal{L}^*_{past}(s)$ and thus $L \in \mathcal{L}_{past}$ because $\mathbb{L}$ is valid in s. Otherwise we have $s' \models L$ and thus $L \in \mathcal{L}_{hold}$.

(2b) $\mathcal{L}_{fut} \setminus \mathcal{L}_{add} \subseteq \mathcal{L}^*_{fut}(s')$: For any $L \in \mathcal{L}_{fut} \setminus \mathcal{L}_{add}$ we have $L \in \mathcal{L}_{fut} \subseteq \mathcal{L}^*_{fut}(s)$ because $\mathbb{L}$ is valid in s. Hence, L is added at time $i > 0$ by all $\pi \in \mathcal{P}(s)$ and in particular by all $\pi = \langle t \rangle \circ \pi'$. Since $L \notin \mathcal{L}_{add}$, we know that L is not added by t and L must hence be added by all $\pi' \in \mathcal{P}(s')$. Therefore $L \in \mathcal{L}^*_{fut}(s')$ which concludes the proof. □

The next two progression functions focus solely on adding more landmarks to the set of future landmarks. They progress the set of past landmarks to $\mathcal{L}_I$ which is always valid (since $\mathcal{L}_I \supseteq \mathcal{L}^*_{past}(s)$ for any s) and acts as a neutral element for set intersection on landmark sets. We exploit the latter property later on, when merging the result of these progression functions with the result of a progression function that considers past landmarks (e.g., basic progression).

Richter, Helmert, and Westphal (2008) observe that a past landmark A is *required again* in state s if $s \not\models A$ and there is a greedy-necessary ordering $A \rightarrow_{\text{gn}} B$ such that B is not in the past.

Definition 5. ***Greedy-necessary ordering progression***
The function prog_{gn} maps landmark state $\mathbb{L} = \langle \mathcal{L}_{past}, \mathcal{L}_{fut} \rangle$ and transition $t = \langle s, a, s' \rangle$ to $\mathbb{L}' = \langle \mathcal{L}_I, \mathcal{L}_{\text{gn}} \rangle$ where $\mathcal{L}_{\text{gn}} = \{A \in \mathcal{L}_I \mid s' \not\models A \text{ and } \exists A \rightarrow_{\text{gn}} B \in \mathcal{O}_I : B \notin \mathcal{L}_{past}, s' \not\models B\}$.

Theorem 3. *The progression function prog_{gn} is valid.*

Proof. **(1)** $\mathbb{L} = \bot$ is the only such case.

(2a) $\mathcal{L}_I \supseteq \mathcal{L}^*_{past}(s')$ follows directly.

(2b) $\mathcal{L}_{\text{gn}} \subseteq \mathcal{L}^*_{fut}(s')$: For any $A \in \mathcal{L}_{\text{gn}}$ there is an ordering $A \rightarrow_{\text{gn}} B \in \mathcal{O}_I$ such that $B \notin \mathcal{L}_{past}$ and $s' \not\models B$. Thus, $\textit{first}(B, \pi) > 0$ for all $\pi \in \mathcal{P}(s')$. The ordering also implies $s_{\textit{first}(B,\pi)-1} \models A$, and with $\textit{first}(B, \pi) > 0$ we get $\pi \models A$. Since $A \in \mathcal{L}_{\text{gn}}$ implies $s' \not\models A$, it follows that π adds A at time $i > 0$, and thus $A \in \mathcal{L}^*_{fut}(s')$. □

Similarly, landmarks that hold in all goal states but not in the current state are also required again.

Definition 6. ***Goal progression***
The function prog_{goal} *maps landmark state* $\mathbb{L} = \langle \mathcal{L}_{past}, \mathcal{L}_{fut} \rangle$ *and transition* $t = \langle s, a, s' \rangle$ *to* $\mathbb{L}' = \langle \mathcal{L}_I, \mathcal{L}_{goal} \rangle$ *where* $\mathcal{L}_{goal} = \{L \in \mathcal{L}_I \mid s' \not\models L \text{ and } \forall s^* \in S^* : s^* \models L\}$.

Theorem 4. *The progression function* prog_{goal} *is valid.*

Proof sketch. prog_{goal} is valid because it is a special case of prog_{gn}. To see this, we transform the original transition system to an equivalent one with a single goal state g with transitions to g from all original goal states. With a special feature γ that holds only in g and no other state, we can add landmark orderings $L \rightarrow_{\text{gn}} \gamma$ for all landmarks L that hold in all $s^* \in S^*$ (i.e., $\mathcal{L}_{goal}$) because L must hold to reach g, the only state where γ holds. Then, $\mathcal{L}_{\text{gn}}$ (Definition 5) on the new landmark graph is a superset of $\mathcal{L}_{goal}$ on the old one. □

Now we have all components to define the progression function used in LM-A* (Karpas and Domshlak 2009).

Definition 7. **LM-A*** ***progression***
The function $\text{prog}_{\text{LM-A}^*}$ *maps landmark state* $\mathbb{L}$ *and transition* t *to the landmark state* $\mathbb{L}' = \text{merge}(\{\text{prog}_{basic}(\mathbb{L}, t), \text{prog}_{\text{gn}}(\mathbb{L}, t), \text{prog}_{goal}(\mathbb{L}, t)\})$.

Corollary 1. *The progression function* $\text{prog}_{\text{LM-A}^*}$ *is valid.*

Proof. Follows directly from Theorems 1, 2, 3, and 4. □

The LAMA planner (Richter and Westphal 2010) additionally considers reasonable orderings. Richter and Westphal observe that a reasonable ordering $A \rightarrow_{\text{r}} B$ denotes that B should remain in the set of future landmarks if it is reached before A. Therefore, they say a landmark can become *accepted* only if all its predecessors in the landmark graph are accepted already. We express their notion of accepted landmarks as $\mathcal{L}_{past}$ of a landmark state in our framework.

Definition 8. **LAMA** ***progression***
The function $\text{prog}_{\text{LAMA}}$ *maps landmark state* $\mathbb{L} = \langle \mathcal{L}_{past}, \mathcal{L}_{fut} \rangle$ *and transition* $t = \langle s, a, s' \rangle$ *to* $\text{merge}(\{\mathbb{L}_{acc}, \text{prog}_{\text{gn}}(\mathbb{L}, t), \text{prog}_{goal}(\mathbb{L}, t)\})$ *where* $\mathbb{L}_{acc} = \langle \mathcal{L}_{past} \cup \mathcal{L}_{acc}, \mathcal{L}_{fut} \setminus \mathcal{L}_{acc} \rangle$ *and in turn* $\mathcal{L}_{acc} = \{B \in \mathcal{L}_I \mid s' \models B \text{ and } \forall (A \rightarrow_\tau B) \in \mathcal{O}_I : A \in \mathcal{L}_{past}\}$.

The LAMA progression achieves that the landmark B remains in $\mathcal{L}_{fut}$ while there exists an ordering $A \rightarrow_\tau B$ for which A is not accepted. However, it is not valid because the meaning of accepted landmarks does not match the semantics of past landmarks.

Theorem 5. *The progression function* $\text{prog}_{\text{LAMA}}$ *is* not *valid.*

Proof. Figure 1 shows a counterexample where LAMA progression results in an invalid landmark state. The states in the transition system from Figure 1a indicate which state features α, β, and γ hold in each state. The only plan is $\pi = \langle b, c, a \rangle$. We assume the landmark graph depicted in Figure 1b to be given. It contains only correct information: α, β, and γ are landmarks because all of them hold in some state visited by π. Furthermore, $\alpha \rightarrow_{\text{n}} \gamma$ holds because α is true before γ, $\beta \rightarrow_{\text{gn}} \gamma$ holds because β is true one step before γ is true for the first time, and $\beta \rightarrow_{\text{r}} \alpha$ holds because α is removed when β is first added, but holds again afterwards (condition (ii) in the definition of reasonable orderings).

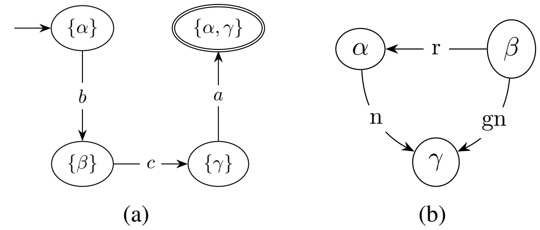

Figure 1: A transition system (a) and its landmark graph (b).

	$\{\alpha\}$	$\{\beta\}$	$\{\gamma\}$	$\{\alpha, \gamma\}$
$\mathcal{L}_{past}$	$\emptyset$	$\{\beta\}$	$\{\beta\}$	$\{\alpha, \beta\}$
$\mathcal{L}_{fut}$	$\{\alpha, \beta, \gamma\}$	$\{\alpha, \gamma\}$	$\{\alpha, \gamma\}$	$\{\gamma\}$

Table 1: Evolution of past and future landmarks according to the LAMA progression for the example from Figure 1.

Since $s_I \models \alpha$, we have $\alpha \in \mathcal{L}^*_{past}(s_I)$. However, since the LAMA progression will not accept α until β is accepted, we get $\text{prog}_{\text{LAMA}}(\langle \emptyset, \mathcal{L}_I \rangle, \langle \cdot, init, s_I \rangle) = \langle \emptyset, \mathcal{L}_I \rangle$ which contradicts the definition of validity as $\emptyset \not\supseteq \mathcal{L}^*_{past}(s_I)$. □

Given that the notion of accepted landmarks is different to the one of past landmarks, it is no surprise that $\text{prog}_{\text{LAMA}}$ derives invalid landmark states. One might also assume that it does not matter if $\mathcal{L}_{past}$ is not accurate, since we are mainly interested in $\mathcal{L}_{fut}$. However, if we continue the LAMA progression on the example from the proof (see Table 1), we see that invalid $\mathcal{L}_{past}$ information can also invalidate $\mathcal{L}_{fut}$ (for example, the goal state should have no future landmarks), at which point the heuristic is sure to be affected.

Note that we are not the first to observe that the LAMA progression is flawed. Richter and Westphal report themselves that the heuristic resulting from this progression is not goal-aware (Richter and Westphal 2010, p. 149). Moreover, Buffet and Hoffmann (2010) recognize the issue that two landmarks A and B in a reasonable ordering $A \rightarrow_{\text{r}} B$ may be achieved simultaneously. If there is a chain of reasonable orderings where all involved landmarks can be achieved at once, LAMA accepts at most the first one when it should accept the entire chain. Buffet and Hoffmann adapt $\mathcal{L}_{acc}$ in Definition 8 to achieve this, obtaining valid landmark states whenever B is not added strictly before A. However, this does not change the result of our counterexample, meaning the altered LAMA progression is still invalid.

Since LAMA is a satisficing planner, using sound information is not imperative. As long as the heuristic estimates are somewhat accurate, they should guide the search reasonably well. However, we can easily extend the example from Figure 1 to render the landmark component of LAMA arbitrarily bad. To do so, replace the transition $\langle \{\beta\}, c, \{\gamma\} \rangle$ by a trajectory $\langle \langle \{\beta\}, c_1, \{\gamma_1\} \rangle, \ldots, \langle \{\gamma_{n-1}\}, c_n, \{\gamma_n\} \rangle \rangle$ and replace γ in the goal state by γ_n. Then, there is a landmark for each γ_i and they are all ordered naturally after α. Using $\text{prog}_{\text{LAMA}}$ results in all γ_i still being in $\mathcal{L}_{fut}$ in the goal, leading to a heuristic value of n when using the heuristic of LAMA. This is undesirable even for satisficing planning.

New Progression Functions

The negative validity result for LAMA progression raises the question if it is possible to derive sound future landmark information from reasonable orderings. The following progression function answers this question in the affirmative. As before, progression functions defined in this section progress the past landmarks to $\mathcal{L}_I$ which is the neutral element when merging landmark states resulting from different progression techniques.

Definition 9. ***Reasonable ordering progression***
The function $\mathrm{prog}_{\mathrm{r}}$ *maps landmark state* $\mathbb{L} = \langle \mathcal{L}_{past}, \mathcal{L}_{fut} \rangle$ *and transition* $t = \langle s, a, s' \rangle$ *to*

$$\mathbb{L}' = \begin{cases} \text{Ⅎ} & \text{if } \exists A \rightarrow_{\mathrm{r}} B \in \mathcal{O}_I : \\ & \quad A \notin \mathcal{L}_{past}, s \models B, s' \models A, s' \models B \\ \langle \mathcal{L}_I, \mathcal{L}_{\mathrm{r}} \rangle & \text{otherwise} \end{cases}$$

where $\mathcal{L}_{\mathrm{r}} = \{ B \in \mathcal{L}_I \mid \exists A \rightarrow_{\mathrm{r}} B \in \mathcal{O} : A \notin \mathcal{L}_{past}$ *and* $(s' \not\models A$ *or* $s' \not\models B) \}$

To show validity of $\mathrm{prog}_{\mathrm{r}}$, we first observe that reasonable orderings $A \rightarrow_{\mathrm{r}} B$ imply $last(B, \pi) \geq first(A, \pi)$ for all $\pi \in \mathcal{P}(s_I)$.

Lemma 1. *Let A and B be landmarks for a state s and let $A \rightarrow_{\mathrm{r}} B$ be a reasonable ordering for s. Then $last(B, \pi) \geq first(A, \pi)$ for all $\pi \in \mathcal{P}(s)$.*

Proof. We show that all s-plans π add B at time $i \geq f = first(A, \pi)$ in all three cases of reasonable orderings.

(i) Follows directly since $f \leq first(B, \pi) \leq last(B, \pi)$.
(ii) Since $s_{f-1} \not\models B$ and $s_g \models B$ for $g \geq f > 0$, π adds B at time i where $f \leq i \leq g$ and thus $f \leq last(B, \pi)$.
(iii) Since $s_f \not\models B$ and $s_g \models B$ for $g > f > 0$, π adds B at time i where $f < i \leq g$ and thus $f \leq last(B, \pi)$.

□

Furthermore, we use the following lemma without proof.

Lemma 2. *Let s, s', and s'' be states of a transition system such that $\pi \in \mathcal{P}(s, s')$ and $\pi' \in \mathcal{P}(s', s'')$ exist. Furthermore, let L be a landmark between s' and s'' (i.e., $\pi' \models L$). If $last(L, \pi') > 0$ then $last(L, \pi \circ \pi') = |\pi| + last(L, \pi')$.*

With these results, we can show that reasonable ordering progression is valid.

Theorem 6. *The progression function* $\mathrm{prog}_{\mathrm{r}}$ *is valid.*

Proof. **(1)** Proof by contradiction: Assume there is an s'-plan $\pi' \in \mathcal{P}(s')$. Since $A \notin \mathcal{L}_{past}$, there is a trajectory $\pi \in \mathcal{P}(s_I, s)$ from s_I to s, and $\pi_I = \pi \circ \langle t \rangle \circ \pi'$ is an s_I-plan. From $A \notin \mathcal{L}_{past}$ and $s' \models A$ it follows that $first(A, \pi_I) = |\pi| + 1$. Then one of the following cases must hold due to the definition of reasonable orderings:

(i) $|\pi| + 1 \leq first(B, \pi_I)$,
(ii) $s \not\models B$ and there exists a g such that $|\pi|+1 \leq g \leq |\pi_I|$ and $s_g \models B$, or
(iii) $s \models B$ and $s' \not\models B$ and there exists a g such that $|\pi| + 1 < g \leq |\pi_I|$ and $s_g \models B$.

However, all three conditions lead to a contradiction: (i) because $s \models B$ implies $first(B, \pi_I) \leq |\pi|$, (ii) because $s \models B$, and (iii) because $s' \models B$. Thus, no s'-plan exists.

(2a) $\mathcal{L}_I \supseteq \mathcal{L}^*_{past}(s')$ follows directly.

(2b) $\mathcal{L}_{\mathrm{r}} \subseteq \mathcal{L}^*_{fut}$: For any $B \in \mathcal{L}_{\mathrm{r}}$, there is a $A \rightarrow_{\mathrm{r}} B \in \mathcal{O}_I$ with $A \notin \mathcal{L}_{past}$, and either $s' \not\models A$ or $s' \not\models B$ holds. From $A \notin \mathcal{L}_{past}$ we know that there is a $\pi \in \mathcal{P}(s_I, s)$ such that $\pi \not\models A$, and we define $\pi_I = \pi \circ \langle t \rangle \circ \pi'$ for all $\pi' \in \mathcal{P}(s')$. From Lemma 1 we know that that $last(B, \pi_I) \geq first(A, \pi_I)$. We now prove the claim both for $s' \not\models A$ and $s' \not\models B$:

$s' \not\models A$**:** Then $\pi \circ \langle t \rangle \not\models A$ and consequently $first(A, \pi_I) = |\pi| + 1 + first(A, \pi')$. With Lemma 1 we get $last(B, \pi_I) \geq |\pi| + 1 + first(A, \pi')$ and with Lemma 2 it follows that $last(B, \pi') \geq first(A, \pi')$. Since $s' \not\models A$ we get $0 < first(A, \pi') \leq last(B, \pi')$.

$s' \not\models B$**:** From $\pi \not\models A$ it follows that $first(A, \pi_I) = |\pi| + first(A, \langle t \rangle \circ \pi')$. Since $A \notin \mathcal{L}_{past}$ we get $s \not\models A$ and thus $first(A, \langle t \rangle \circ \pi') \geq 1$. Hence, $first(A, \pi_I) \geq |\pi| + 1$ and with Lemma 1 $last(B, \pi_I) \geq |\pi| + 1$. From Lemma 2 it follows that $last(B, \pi') \geq 0$. Finally, $last(B, \pi') > 0$ follows from $s' \not\models B$.

Since one of the two cases above must hold and both say $last(B, \pi') > 0$, B must be added in the future of s'.

□

The definition of reasonable ordering progression shows that we can derive more information from landmark orderings than just which landmarks are required again. In particular, we can find that certain trajectories cannot be extended to plans because they contradict the definition of a reasonable landmark ordering. Wang, Baier, and McIlraith (2009) exploit this observation for other ordering types by encoding this knowledge into the description of the planning task and its states directly. In LM-BFS, we deal with this information by pruning states for which we find Ⅎ to be valid. We follow Wang, Baier, and McIlraith and adapt greedy-necessary ordering progression to catch such cases as well.[2]

Definition 10. ***Dead-end aware Greedy-necessary ordering progression***
The function $\mathrm{prog}^*_{\mathrm{gn}}$ *maps landmark state* $\mathbb{L} = \langle \mathcal{L}_{past}, \mathcal{L}_{fut} \rangle$ *and transition* $t = \langle s, a, s' \rangle$ *to*

$$\mathbb{L}' = \begin{cases} \text{Ⅎ} & \text{if } \exists A \rightarrow_{\mathrm{gn}} B \in \mathcal{O}_I : \\ & \quad B \notin \mathcal{L}_{past}, s \not\models A, s' \models B \, . \\ \mathrm{prog}_{\mathrm{gn}}(\mathbb{L}, t) & \text{otherwise} \end{cases}$$

Theorem 7. *The progression function* $\mathrm{prog}^*_{\mathrm{gn}}$ *is valid.*

Proof. **(1)** Proof by contradiction: Assume there is an s'-plan $\pi' \in \mathcal{P}(s')$. Since $B \notin \mathcal{L}_{past}$, there is a trajectory $\pi \in \mathcal{P}(s_I, s)$ from s_I to s, and $\pi_I = \pi \circ \langle t \rangle \circ \pi'$ is an s_I-plan. Furthermore, from $B \notin \mathcal{L}_{past}$ and $s' \models B$ it

[2]Note that this inference is currently not beneficial in practice. Wang, Baier, and McIlraith (2009) point out that LAMA only finds orderings that hold along all trajectories. Finding orderings that hold only along all plans is harder and we are not aware of any landmark (ordering) generators that do so.

follows that $\mathit{first}(B, \pi_I) = |\pi| + 1$. From the definition of greedy-necessary orderings it follows that $s \models A$, but this contradicts the condition $s \not\models A$ for $\mathbb{L}' = \Finv$. Hence no such s'-plan exists.

(2a) and (2b) Follow from Theorem 3.

$\square$

We continue by proposing progression functions for the remaining landmark ordering types and show their validity.

Definition 11. ***Natural ordering progression***
The function $\mathrm{prog_n}$ *maps landmark state* $\mathbb{L} = \langle \mathcal{L}_{\mathit{past}}, \mathcal{L}_{\mathit{fut}} \rangle$ *and transition* $t = \langle s, a, s' \rangle$ *to*

$$\mathbb{L}' = \begin{cases} \Finv & \text{if } \exists A \to_{\mathrm{n}} B \in \mathcal{O}_I : A \notin \mathcal{L}_{\mathit{past}}, s' \models B \\ \langle \mathcal{L}_I, \emptyset \rangle & \text{otherwise} \end{cases}.$$

Theorem 8. *The progression function* $\mathrm{prog_n}$ *is valid.*

Proof. **(1)** Proof by contradiction: Assume there is an s'-plan $\pi' \in \mathcal{P}(s')$. Since $A \notin \mathcal{L}_{\mathit{past}}$, there is a trajectory $\pi \in \mathcal{P}(s_I, s)$ from s_I to s, and $\pi_I = \pi \circ \langle t \rangle \circ \pi'$ is an s_I-plan. Furthermore, $A \notin \mathcal{L}_{\mathit{past}}$ implies that $\pi \not\models A$ and hence $\mathit{first}(A, \pi_I) > |\pi|$. From the definition of natural orderings it follows that $\mathit{first}(B, \pi_I) > |\pi| + 1$ but this contradicts $s' \models B$ which implies $\mathit{first}(B, \pi_I) \leq |\pi| + 1$. Therefore, no such s'-plan exists.

(2a) $\mathbb{L}_I \supseteq \mathcal{L}^*_{\mathit{past}}(s')$ follows directly.

(2b) $\emptyset \supseteq \mathcal{L}^*_{\mathit{fut}}(s')$ follows directly.

$\square$

Necessary orderings are similar to greedy-necessary orderings. The difference is that A is required again whenever A does not hold and B must hold in the future.

Definition 12. ***Necessary ordering progression***
The function $\mathrm{prog_{nec}}$ *maps landmark state* $\mathbb{L} = \langle \mathcal{L}_{\mathit{past}}, \mathcal{L}_{\mathit{fut}} \rangle$ *and transition* $t = \langle s, a, s' \rangle$ *to*

$$\mathbb{L}' = \begin{cases} \Finv & \text{if } \exists A \to_{\mathrm{nec}} B \in \mathcal{O}_I : \\ & \quad s \not\models A, s \not\models B, s' \models B \\ \langle \mathcal{L}_I, \mathcal{L}_{\mathrm{nec}} \rangle & \text{otherwise} \end{cases}$$

where $\mathcal{L}_{\mathrm{nec}} = \{A \in \mathcal{L}_I \mid s' \not\models A \text{ and } \exists A \to_{\mathrm{nec}} B \in \mathcal{O}_I : B \in \mathcal{L}_{\mathit{fut}}, s' \not\models B\}$.

Theorem 9. *The progression function* $\mathrm{prog_{nec}}$ *is valid.*

Proof sketch. Analogous to the proof for $\mathrm{prog}^*_{\mathrm{gn}}$ in Theorems 3 and 7; replace all $B \notin \mathcal{L}_{\mathit{past}}$ with $B \in \mathcal{L}_{\mathit{fut}}$ and reason about all times where B is added, not just the first one. $\square$

The progression of weak orderings is similar to $\mathrm{prog_r}$. One difference is that a weak ordering $A \to_{\mathrm{w}} B$ does not allow the special case where A and B are added simultaneously. Furthermore, it is not required that B becomes false when A is first added, hence there is no case for pruning based on weak orderings.

Definition 13. ***Weak ordering progression***
The function $\mathrm{prog_w}$ *maps landmark state* $\mathbb{L} = \langle \mathcal{L}_{\mathit{past}}, \mathcal{L}_{\mathit{fut}} \rangle$ *and transition* $t = \langle s, a, s' \rangle$ *to* $\mathbb{L}' = \langle \mathcal{L}_I, \mathcal{L}_{\mathrm{w}} \rangle$ *where* $\mathcal{L}_{\mathrm{w}} = \{B \in \mathcal{L}_I \mid \exists A \to_{\mathrm{w}} B \in \mathcal{O}_I : A \notin \mathcal{L}_{\mathit{past}}\}$.

Theorem 10. *The progression function* $\mathrm{prog_w}$ *is valid.*

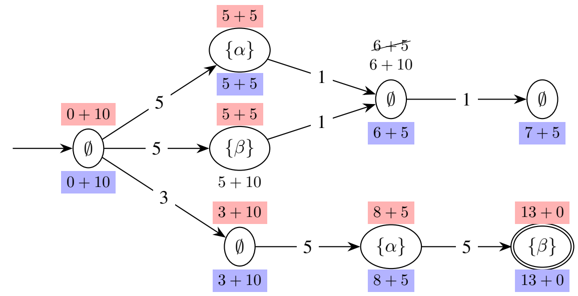

Figure 2: Example transition system where the dominant progression expands more states. The $g + h$ values of an A* search are annotated, above for $\mathrm{prog}_{\mathit{basic}}$ and below for $\mathrm{merge}(\{\mathrm{prog}_{\mathit{basic}}, \mathrm{prog_r}\})$, and colored if expanded.

Proof sketch. Cases (1) and (2a) are identical to the corresponding cases in the proof of Theorem 3. For (2b) we can argue similarly to reasonable orderings: If A is not past in s, then $\pi \not\models A$ for some trajectory π from s_I to s and in particular $s \not\models A$. Since the ordering holds for all s_I-plans, we know that $\mathit{last}(B, \pi_I) > \mathit{first}(A, \pi_I)$ for all $\pi_I \in \mathcal{P}(s_I)$. With the same argument as for reasonable orderings, we can show that from $\mathit{last}(B, \pi_I) > \mathit{first}(A, \pi_I)$ it follows that $\mathit{last}(B, \pi') > \mathit{first}(A, \pi')$ for all s'-plans $\pi' \in \mathcal{P}(s')$. Consequently, B is future in s'. $\square$

We have seen that LM-A* combines multiple valid progression functions to merge their results and obtain a more informed and still valid progression function. Given a set of valid progression functions, using all of them intuitively makes sense: the set of future landmarks can only grow and makes heuristics more informed. However, there are some limitations to this claim. On the one hand, not all heuristics benefit from having more landmarks (e.g., Seipp, Keller, and Helmert 2020). On the other hand, differences in heuristic values change the expansion order of a search algorithm which may have negative effects on its performance. To see this, consider the example in Figure 2, which depicts two landmark-based A* searches (Hart, Nilsson, and Raphael 1968) using different progression functions. The numbers above a state indicate the $g + h$ values when using $\mathrm{prog}_{\mathit{basic}}$, and the numbers below a state indicate the same values when using $\mathrm{prog}' = \mathrm{merge}(\{\mathrm{prog}_{\mathit{basic}}, \mathrm{prog_r}\})$. The considered landmark graph is $\langle \{\alpha, \beta\}, \{\alpha \to_{\mathrm{r}} \beta\} \rangle$ and the (admissible) heuristic estimates a cost of 5 per future landmarks.

Let s be the non-goal state where β holds. When progressing from the initial state to s, $\mathrm{prog}_{\mathit{basic}}$ will result in landmark state $\langle \{\beta\}, \{\alpha\} \rangle$, while prog' obtains $\langle \{\beta\}, \{\alpha, \beta\} \rangle$. The latter results in a higher heuristic value which leads to s being expanded later. But as long as s is not expanded, its successor does not obtain the information that it can be reached without achieving α, yielding a lower heuristic value and resulting in the search expanding this area first.

Experimental Evaluation

For a practical analysis of our contributions, we consider classical planning, where landmarks are an established tool. The Fast Downward planner (Helmert 2006) supports the use of landmarks along the lines of LM-BFS. We implement some of the discussed progression functions in version 20.06 of Fast Downward. Experiments are conducted on Intel Xeon E5-2660 (satisficing) and Intel Xeon Silver 4114 (optimal) processors running on 2.2 GHz with a time limit of 30 minutes and a memory limit of 3.5 GiB. All code, benchmarks, and experiment data are published online (Büchner et al. 2023).

We leave some progression functions out of the analysis for good reason: prog_{nec} because we are not aware of landmark generators for necessary orderings; prog_{w} because Büchner, Keller, and Helmert (2021) derive weak orderings as a subset from reasonable orderings; and prog_{n} because the used landmark generators do not find orderings for which pruning ever triggers. For the same reason, we also do not implement the pruning cases in prog_{gn} and prog_{r}. Furthermore, we do not evaluate the simple progression functions in isolation, but only their derivatives $\text{prog}_{\text{LM-A}^*}$, $\text{prog}_{\text{LAMA}}$, and a new progression function we call *admissible reasonable orderings* and denote it by $\text{prog}_{ARO} := \text{merge}(\{\text{prog}_{basic}, \text{prog}_{\text{gn}}, \text{prog}_{goal}, \text{prog}_{\text{r}}\})$.

Note that implementing $\text{prog}_{\text{LM-A}^*}$ and $\text{prog}_{\text{LAMA}}$ allows to represent landmark states without storing $\mathcal{L}_{fut}$ explicitly. For one, all landmarks that are not past (respectively accepted) are implicitly future. Furthermore, the information gained by prog_{gn} and prog_{goal} and used by these two progressions can be computed purely based on $\mathcal{L}_{past}$ and the current state. Consequently, we can compute them on demand when evaluating the landmark heuristic for a given state. However, progression functions that include prog_{r} or prog_{w} (e.g., prog_{ARO}) cannot make use of this optimization. Consider for example a reasonable ordering $A \rightarrow_{\text{r}} B$. If $B \in \mathcal{L}_{past}$ and $A \notin \mathcal{L}_{past}$, then $B \in \mathcal{L}_{fut}$ by the definition of prog_{r}. Now assume we reach A but do not store $\mathcal{L}_{fut}$ explicitly, then after the next step there is no way to deduce whether B was added before or after adding A and hence it is impossible to reconstruct $\mathcal{L}_{fut}$ accurately. Given that we store landmark states as pairs of bitsets of length $|\mathcal{L}_I|$, our implementation of prog_{ARO} requires twice as much memory per landmark compared to $\text{prog}_{\text{LM-A}^*}$ and $\text{prog}_{\text{LAMA}}$.

Satisficing Planning

We embed prog_{ARO} in LAMA and compare the influence of a valid progression function over the invalid LAMA progression. Our benchmark suite consists of 2772 planning tasks from the satisficing track of the International Planning Competitions (IPCs) 1998–2018. We note that the order in which LAMA expands states can significantly influence planner performance. To filter out random noise, we run all configurations 5 different random seeds to vary the successor orderings. The numbers we report in the following denote their average value.

Since LAMA originally considers obedient-reasonable orderings, we use this as our baseline and denote it by $\text{prog}^{o}_{\text{LAMA}}$. We also consider LAMA without obedient-reasonable orderings, denoted as $\text{prog}_{\text{LAMA}}$, to have a closer comparison with prog_{ARO} which also does not consider them. Furthermore, we denote by prog^{c}_{ARO} a configuration that keeps cycles in the landmark graph of the initial state.

	$\text{prog}^{o}_{\text{LAMA}}$	$\text{prog}_{\text{LAMA}}$	prog_{ARO}	prog^{c}_{ARO}
$\text{prog}^{o}_{\text{LAMA}}$	–	7.2	160.0	203.6
$\text{prog}_{\text{LAMA}}$	**8.6**	–	161.2	204.0
prog_{ARO}	**204.4**	**203.8**	–	**94.8**
prog^{c}_{ARO}	**236.2**	**230.4**	85.0	–

Table 2: Per-task comparison of the final plan cost of LAMA. Each cell denotes how many tasks (on average) have a lower cost in the row method compared to the column. The winner of each pairwise comparison is highlighted in boldface.

Traditionally, LAMA systematically removes orderings until the landmark graph is acyclic. This is to avoid chains of landmarks that are permanently unaccepted which results in poor heuristic values, but as we have shown this is rather a consequence of the invalid progression and acyclic graphs do not resolve that problem completely.

The coverage is virtually unaffected by the choice of progression function. All four configurations solve between 2350.4 and 2352.2 problems. This may seem as if cases like the one presented in Figure 1 do not occur in practice, but we think this is rather a consequence of LAMA combining landmarks with guidance from the FF heuristic (Hoffmann and Nebel 2001); whenever $\text{prog}_{\text{LAMA}}$ suffers from the flaws we aim to fix with prog_{ARO}, the FF heuristic might be able to compensate for that.

Nevertheless, we can observe an improvement of prog_{ARO} over $\text{prog}_{\text{LAMA}}$ in terms of plan quality, i.e., the cost of the best plan found before the planner terminates. Table 2 compares pair-wise which strategy finds a cheaper plan in how many tasks. Interestingly, $\text{prog}_{\text{LAMA}}$ without obedient-reasonable orderings performs almost the same as $\text{prog}^{o}_{\text{LAMA}}$; the plan cost differs in only 15.8 tasks on average with a slight advantage for $\text{prog}_{\text{LAMA}}$. Since an analysis of obedient-reasonable orderings is out of the scope of this paper, we do not investigate this further.

The difference is more significant when we compare to prog_{ARO} and prog^{c}_{ARO}. While neither technique strictly dominates the others, our new progressions tend to find cheaper plans than $\text{prog}_{\text{LAMA}}$ and $\text{prog}^{o}_{\text{LAMA}}$. More specifically, prog_{ARO} on average finds more cheaper plans than all other configurations. We have expected prog^{c}_{ARO} to find even cheaper plans as considering more orderings potentially leads to more informed heuristics overall. However, this is not the case. One possible explanation is that these additional orderings increase the evaluation time for the heuristic. Since LAMA restarts the weighted A* search (Pohl 1970) with lower weight whenever it finds a plan, higher heuristic evaluation times might lead to fewer restarts resulting in more expensive plans overall.

In summary, these results indicate that the ARO progression can replace the LAMA progression with no major drawbacks. It even yields a slight advantage in terms of plan cost. Moreover, contrary to the LAMA progression, ARO is not limited by restrictions such as acyclic landmark graphs. This means it can benefit even more if better landmark generators for s_I become available in the future.

Cost-Optimal Planning

One of our main contributions is making non-natural orderings applicable for path-dependent *optimal* planning. This is an important step given the recent findings that considering cyclic dependencies in landmark heuristics is beneficial, as cycles in the landmark graph of solvable tasks contain at least one non-natural ordering (Büchner, Keller, and Helmert 2021). We compare $\text{prog}_{\text{LM-A}^*}$ to prog^c_{ARO} to evaluate the empirical impact of using the admissible landmark heuristic h^{MHS} (Karpas and Domshlak 2009) in the (equivalent) dual formalisation via operator counting constraints, and the strong cyclic landmark heuristic with either the Johnson cycle detection h^c_J or the oracle method h^c_O (Büchner, Keller, and Helmert 2021). We use the 1827 benchmark tasks from the optimal tracks of the IPCs 1998–2018. Landmarks and orderings were generated with the following methods: LM^{RHW} (Richter, Helmert, and Westphal 2008), LM^{h^1} and LM^{h^2} (Keyder, Richter, and Helmert 2010), and LM^{BJOLP} (Domshlak et al. 2011).

The task coverage increases between 2 to 7 from $\text{prog}_{\text{LM-A}^*}$ to prog^c_{ARO} for all tested combinations of landmark generators and heuristics. Interestingly, the cyclic landmark heuristics yield lower coverage than h^{MHS} which is a contrast to Büchner, Keller, and Helmert (2021) who recompute the landmarks from scratch in every state. Our observation aligns with the findings of Büchner (2020) that many problem instances have acyclic initial state landmark graphs; if the initial state is acyclic, then so are all successors in the path-dependent setting. In all these cases, the cyclic landmark heuristics have the overhead of the cycle detection although none exist. We anticipate that this result changes if new landmark generators specializing on finding cyclic orderings become available in the future.

Figure 3 shows the number of expanded states before the last f-layer using h^{MHS}. (The plots for h^c_J and h^c_O look almost identical.) We can see that prog^c_{ARO} expands fewer states in many problems and sometimes even results in the perfect heuristic (bottom line). Among all tasks solved with both progression functions, there are exactly two tasks for which prog^c_{ARO} expands more states than $\text{prog}_{\text{LM-A}^*}$. While this proves that sometimes better informed progression leads to more expansions in practice, it happens seldom enough to defend prog^c_{ARO} as the reasonable choice.

The lower amount of expansions reduces search time, which is also the reason for the slightly higher coverage of prog^c_{ARO}. The number of tasks for which we run out of memory is identical for both approaches, even though we store twice as much landmark information per state with prog^c_{ARO} compared to $\text{prog}_{\text{LM-A}^*}$, since we store for each landmark not only whether it is past, but also whether it is future.

Conclusion

We generalize the idea of landmarks from classical planning to deterministic transition systems. We propose the LM-BFS framework, a generic best-first search using landmark-based heuristics. Since recomputing landmarks in every state is usually too expensive, LM-BFS includes the notion of landmark progression which progresses landmark states from one state to its successors. We study progression functions from the literature and show that the progression in LAMA (Richter and Westphal 2010) is flawed. Furthermore, we propose valid alternatives that were never considered before.

Our experiments show that the newly introduced progression functions improve planner performance. However, some of them are not (yet) useful in practice because no landmark generator produces the required information; we are not aware of work that exploits necessary or weak orderings, and orderings that only hold on all plans but not all trajectories are hard to find. Coming up with new landmark generators is an interesting line of future work. Furthermore, allowing LM-BFS to add new landmarks and orderings during search would be an interesting extension, because sometimes those from the initial state are not good enough.

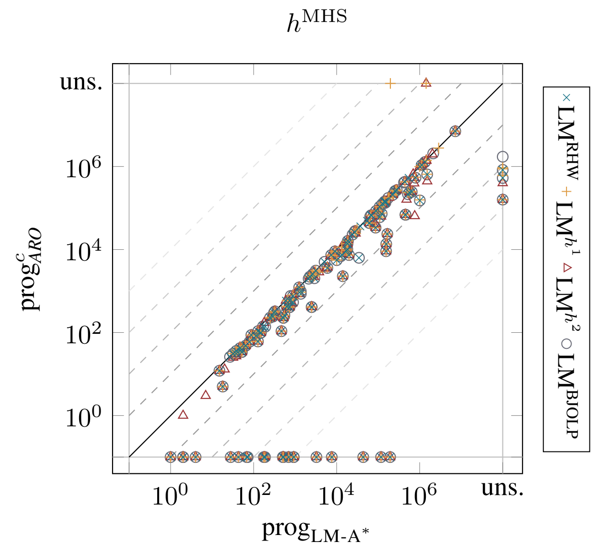

Figure 3: Number of expanded states before the last f-layer. Only instances where the numbers differ are displayed.

Acknowledgments

We have received funding for this work from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 817639). Moreover, this research was partially supported by TAILOR, a project funded by the EU Horizon 2020 research and innovation programme under grant agreement no. 952215.

References

Bonet, B.; and Helmert, M. 2010. Strengthening Landmark Heuristics via Hitting Sets. In Coelho, H.; Studer, R.; and Wooldridge, M., eds., *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, 329–334. IOS Press.

Büchner, C. 2020. *Generalization of Cycle-Covering Heuristics*. Master's thesis, University of Basel.

Büchner, C.; Keller, T.; Eriksson, S.; and Helmert, M. 2023. Code, Benchmarks and Experiment Data for the ICAPS

2023 Paper "Landmark Progression in Heuristic Search". https://doi.org/10.5281/zenodo.7733173.

Büchner, C.; Keller, T.; and Helmert, M. 2021. Exploiting Cyclic Dependencies in Landmark Heuristics. In Goldman, R. P.; Biundo, S.; and Katz, M., eds., *Proceedings of the Thirty-First International Conference on Automated Planning and Scheduling (ICAPS 2021)*, 65–73. AAAI Press.

Buffet, O.; and Hoffmann, J. 2010. All that Glitters is not Gold: Using Landmarks for Reward Shaping in FPG. In *ICAPS 2010 Workshop on Planning and Scheduling Under Uncertainty*.

Domshlak, C.; Helmert, M.; Karpas, E.; Keyder, E.; Richter, S.; Röger, G.; Seipp, J.; and Westphal, M. 2011. BJOLP: The Big Joint Optimal Landmarks Planner. In *IPC 2011 Planner Abstracts*, 91–95.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2): 100–107.

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 Gerevini, A.; Howe, A.; Cesta, A.; and Refanidis, I., eds., *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS 2009)*, 162–169. AAAI Press.

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 Boutilier, C., ed., *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, 1728–1733. AAAI Press.

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

Pohl, I. 1970. Heuristic search viewed as path finding in a graph. *Artificial Intelligence*, 1: 193–204.

Richter, S.; Helmert, M.; and Westphal, M. 2008. Landmarks Revisited. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008)*, 975–982. AAAI Press.

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

Seipp, J.; Keller, T.; and Helmert, M. 2020. Saturated Cost Partitioning for Optimal Classical Planning. *Journal of Artificial Intelligence Research*, 67: 129–167.

Simon, S.; and Röger, G. 2015. Finding and Exploiting LTL Trajectory Constraints in Heuristic Search. In Lelis, L.; and Stern, R., eds., *Proceedings of the Eighth Annual Symposium on Combinatorial Search (SoCS 2015)*, 113–121. AAAI Press.

Wang, L.; Baier, J.; and McIlraith, S. 2009. Viewing Landmarks as Temporally Extended Goals. In *ICAPS 2009 Workshop on Heuristics for Domain-Independent Planning (HDIP)*, 49–56.

Zhu, L.; and Givan, R. 2003. Landmark Extraction via Planning Graph Propagation. In *ICAPS 2003 Doctoral Consortium*, 156–160.