**Lemming: A Tool for Guided Plan Selection using Landmarks**

Jungkoo Kang* · Tathagata Chakraborti* · Michael Katz · Shirin Sohrabi · Francesco Fuggitti

IBM Research

**Abstract**

Lemming is a visualization tool for the selection of plans for a given problem, allowing the user to efficiently whittle down the set of plans and select their plan(s) of choice. We propose three different user experiences for this process, all based on the principle that using landmarks as guidance can help cut down the set of choice points for the user. The live demonstration at the conference will allow the audience to interact with the tool on different domains and problems.

**GitHub**  https://github.com/IBM/lemming

**Introduction**

The use of AI often requires a human-in-the-loop component so that users are able to make informed decisions. One such decision is identifying and choosing the most “interesting” plan for a particular user. It is possible to elicit the user preferences (Das et al. 2019; Mantik, Li, and Porteous 2022) and/or specify these preferences in a language that a planner can reason about, such as PDDL3.0 (Gerevini and Long 2005) and then let the planner select an optimal plan. However, this solution is not practical, especially in cases where not all preferences and constraints are known (or can be modeled) up front. To this end, there is a long history of work on generating multiple plans for a planning problem, either in the form of top-k planning (Riabov, Sohrabi, and Udrea 2014; Katz et al. 2018), top-quality planning (Katz, Sohrabi, and Udrea 2020), or diverse planning (Nguyen et al. 2012; Katz and Sohrabi 2020; Katz, Sohrabi, and Udrea 2022). This comes with the premise that the plan that the user is “interested” in is among the generated plans.

Recently, there have been several applications that explore such approaches (i.e., generate multiple plans and then involve the user in the selection process). Some of these applications are in the area of patient monitoring (Sohrabi, Udrea, and Riabov 2014), enterprise risk management (Sohrabi et al. 2018), conversational systems (Chakraborti et al. 2022; Rizk et al. 2020; Sreedharan et al. 2020b), and web service composition (Brachman et al. 2022). However, the user interfaces for interacting with such systems has received little attention. For example, in (Chakraborti et al. 2021), all plans were shown to the user as separate sequences to select from – an approach that of course does not scale to more ambiguous problems i.e. a larger set of plans, while in these applications (Sohrabi et al. 2020, 2018; Feblowitz et al. 2021) a custom solution was implemented.

In this paper, we present Lemming, a tool for providing a domain-independent approach to the plan disambiguation problem. Our tool allows end-users to compare and select a plan from any automated planner that produces multiple plans. The process of selecting a plan by the user can be costly, inconsistent, or error-prone. To address this, Lemming uses landmarks to help the user focus on a particular component of the search space. We propose three different user experiences for this process, all based on the principle that using landmarks as guidance can help cut down the set of choice points for the user.

**Existing tools**  There are several tools that help with specification (e.g., planning.domains (Muise 2023)) and visualization of plans (Magnaguagno et al. 2020; Magnaguagno 2020a,b). These approaches aim to help domain experts create planning models rather than guiding an end-user in the selection of the plans. On the other hand, while the notion of imprecision and uncertainty (Zhang and Huang 1994) or allowing easier comparison of plans by using a query space and clustering (Ghosh et al. 2002), or allowing some form of automated plan selection (Aha, Molineaux, and Ponsen 2005) is explored in the literature, none of these make the connection to the visualization and/or human in the loop component of the selection process.

Landmarks have an enormous history of use in speeding up the combinatorial search process for planning (Porteous, Sebastia, and Hoffmann 2001; Keyder, Richter, and Helmer 2010; Hoffmann, Porteous, and Sebastia 2004; Richter, Helmer, and Westphal 2008), as well as in planning-adjacent tasks like plan recognition (Pereira, Oren, and Meneguzzi 2020). In the past, landmarks have also been used to summarize plans (Chen and Mooney 2011; Grover et al. 2020; Sreedharan et al. 2020b) to the end-user and debug plans (Sreedharan et al. 2020a) to the developer in complex real-world domains such as in the authoring of goal-oriented conversational agents (Muise et al. 2019), as well as for localization in path planning settings (Mataric 1992). To the best of our knowledge, this is the first attempt at using landmarks for plan disambiguation with end users.

*Equal contribution.
Lemming Overview

The user interaction with Lemming begins with a domain-problem pair and optionally with an already generated set of plans. If the plans are not already provided, we use the Forbid Iterative planner (Katz et al. 2018; Katz, Sohrabi, and Udrea 2020) that produces a set of plans with the desired characteristics (e.g. quality, cost-bound, etc.) given a domain-problem pair. The objective of Lemming is to let the user explore and select from these plans the ones they are interested in. We optimize for two objectives:

1. Size of visualization: The visualization of the entire set of plans can be impractical depending on its size. For the user to make informed choices, they must be able to interact with a tractable representation of the plans.

2. Number of choices: As the size of the plan set grows, so does the number of choice points if the user is left to select options in the visualization without any guidance. The novelty of Lemming is in the use of landmarks to minimize the number of choices the user has to make.

Based on these two considerations, we end up with three different ways to visualize a set of plans, as follows:

Disambiguation Graph The first item of interest is a disambiguation graph that greedily partitions the set of plans into a sequence of most disambiguating partitions. While this might not be most useful to the user as a visualization by itself, it is key to the other modes of visualization e.g. as a means of proactively surfacing the next choice points to the user. This is Step 4 in Algorithm 1 where our disambiguation criterion \( L \) is a set of disjunctive action landmarks.

BUILD Experience In a “build experience” the user can progressively build their plan a few steps at a time, starting from the goal (or initial) state and using maximal suffixes (or prefixes) to choices of only the plans that the user has selected at any moment. This is shown in Steps 4 and 6 in Algorithm 1. Of course, an incremental build experience means that the user does not see the full picture upfront. This can lead to a loss of situational awareness and the user may end up pruning plans they might have been interested in.

SELECT Experience Contrary to BUILD, here we start with the full picture – where we show all the plans of interest and what states they traverse – and allow the user to select one (or more, in “commit mode”) landmarks and whittle down to their plans of choice. Thus, this view shows the full space of interesting solutions for the user to select from.

Limitations We want to emphasize here that our aim is to cut down on the number of disambiguation choices the user has to make – landmark makes for a natural ally here since it surfaces the most necessary (and probably important) parts of the planning task. However, a landmark-based approach does come with some limitations: 1) the worst-case number of choices the user has to make is the same with or without landmarks; 2) the greedy disambiguation graph may end up missing the user’s preferred plan (especially in the build experience); and 3) a collection of plans disambiguated with landmarks is not expressive enough to capture arbitrarily complex user preferences not modeled in the domain.

Algorithm 1: Guided Plan Selection in Lemming

1: Find a set of plans \( P \) for the planning task
2: Find a set \( L \) of disambiguation criteria for plans
3: while \( |L| \geq 1 \) or user does not break do
4: \( \text{BUILD} \) Choose \( l \) closest to goal/initial state
5: \( \text{SELECT} \) Choose \( l \) that locally maximizes disambiguation in the worst case (\( \text{arg min}_{i \in L} \max_{a \in l} |P_i| \))
6: \( \text{BUILD} \) Show only the part of the digraph containing the goal (or initial) state and part of the plans containing the actions chosen by the user. Hide nodes and edges upstream (or downstream) from \( l \)
7: \( \text{BUILD} \) Show only the part of the digraph containing the goal (or initial) state and part of the plans containing the actions chosen by the user. Hide nodes and edges upstream (or downstream) from \( l \)
8: Ask the user to choose an \( a_i \) in \( l \) (or none, if \( P_0 \neq \emptyset \))
9: Set \( P = P_i \) that corresponds to the chosen \( a_i \)
10: end while
11: if \( |P| > 1 \) Return randomly \( p_i \in P \) else Return \( P \)

select one (or more, in “commit mode”) landmarks and whittle down to their plans of choice. Thus, this view shows the full space of interesting solutions for the user to select from.

Figure 1: Different modes of plan disambiguation in Lemming. The green highlight indicates the next choice point for the user.
References


Hoffmann, J.; Porteous, J.; and Sebastia, L. 2004. Ordered Landmarks in Planning. JAIR.


Nguyen, T. A.; Do, M. B.; Gerevini, A.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012. Generating Diverse Plans to Handle Unknown and Partially Known User Preferences. AIJ.

Pereira, R. F.; Oren, N.; and Meneguzzi, F. 2020. Landmark-Based Approaches for Goal Recognition as Planning. AIJ.


