`Lemming`: A Tool for Guided Plan Selection

Michael Katz · Jungkoo Kang · Tathagata Chakraborti · Shirin Sohrabi

IBM Research AI

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.

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 which of course does not scale to more ambiguous problems i.e. 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 (Porteous, Sebastia, and Hoffmann 2001; Keyder, Richter, and Helmert 2010; Hoffmann, Porteous, and Sebastia 2004; Richter, Helmert, and Westphal 2008) 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. The tool will be open-sourced for the planning community at the time of its presentation.

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.

`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.

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: Pop disambiguation option $l = \{a_1, ..., a_k\} \in L$

Scheme-2 Choose l closest to goal/initial state

Scheme-3 Choose l that locally maximizes disambiguation in the worst case $(\arg\min_{\{l \in L\}} \max_{a_i \in l} |P_i|)$

5: Find plans $P_1, ..., P_k \in P$ according to l such that $\cup P_i \subseteq P$. Let $P_0 = P \setminus \cup P_i$.
6: **if** $\forall i, P_i \in \{P, \emptyset\}$, **continue**
7: Create and show a graphical representation (digraph) of P

Scheme-2 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

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 (we discuss this in a bit more detail in the attached video):

Scheme-1: Disambiguation Graph

The first 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 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.

Landmark-Based Disambiguation 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. In the past, landmarks have been used to summarize (Sreedharan et al. 2020b) and debug (Sreedharan et al. 2020a) complex real-world domains. On the other hand, 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, as mentioned above); and 3) a collection of plans disambiguated with landmarks is not expressive enough to capture arbitrarily complex user preferences not modeled in the domain.

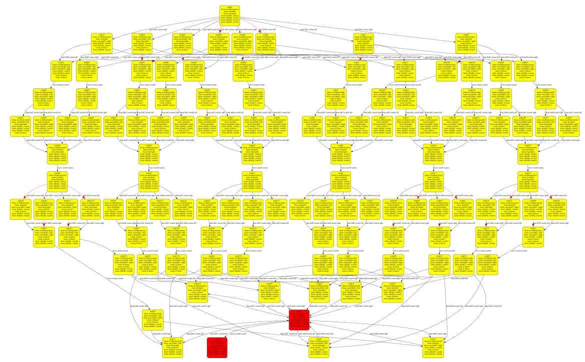

Figure 1: Scheme-3: State transition of a set of plans with goals and landmarks highlighted for selection. As we see in the attached video, these graphs can get big really quickly and without guidance (e.g. using landmarks) the user will need to disambiguate arbitrarily among many choices.

Scheme-2: Disambiguation Build Experience

The first proposed visualization scheme is a "build experience" where 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 landmarks 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.

Scheme-3: Disambiguation Selection Experience

Contrary to Scheme-2, 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 landmarks and whittle down to their plans of choice. A sample visualization and landmark selection step is shown in Figure 1. The supplementary video illustrates this scheme in detail. Currently, `Lemming` is a tool under construction, and at the time of submission, we support Scheme-3 only. This is illustrated in Figure 1 for a random problem-domain pair from the `gripper` IPC domain (McDermott 1998). However, we intend to demonstrate all three schemes at the conference.

Demonstration Logistics During the demonstration, the audience will be able to interact with the interface in two forms: 1) upload their own planning task, generate a set of plans, and use `Lemming` to select a desired plan; or 2) start with a set of plans in some domains of industry interest, and visualize and disambiguate them as before.

Link to Video https://ibm.biz/icaps-lemming

References

Aha, D. W.; Molineaux, M.; and Ponsen, M. 2005. Learning to Win: Case-Based Plan Selection in a Real-Time Strategy Game. In *Case-Based Reasoning Research and Development: International Conference on Case-Based Reasoning (ICCBR)*.

Brachman, M.; Bygrave, C.; Chakraborti, T.; Chaudhary, A.; Ding, Z.; Dugan, C.; Gros, D.; Gschwind, T.; Johnson, J.; Laredo, J.; Czasch, C. M.; Pan, Q.; Rai, P.; Ramalingam, R.; Scotton, P.; Surabathina, N.; and Talamadupula, K. 2022. A Goal-driven Natural Language Interface for Creating Application Integration Workflows. In *AAAI Demonstration Track*.

Chakraborti, T.; Agarwal, S.; Brimijoin, K.; Agarwal, P.; Rizk, Y.; Moran, D. S.; Boag, S.; and Khazaeni, Y. 2021. Planning for Automated Composition of Aggregated Assistants. In *ICAPS Demonstration*.

Chakraborti, T.; Rizk, Y.; Isahagian, V.; Aksar, B.; and Fuggitti, F. 2022. From Natural Language to Workflows: Towards Emergent Intelligence in Robotic Process Automation. In *BPM 2022 RPA Forum*.

Das, M.; Odom, P.; Islam, M. R.; Doppa, J. R.; Roth, D.; and Natarajan, S. 2019. Planning with actively eliciting preferences. *Knowl. Based Syst.*, 165: 219–227.

Feblowitz, M.; Hassanzadeh, O.; Katz, M.; Sohrabi, S.; Srinivas, K.; and Udrea, O. 2021. IBM Scenario Planning Advisor: A Neuro-Symbolic ERM Solution. In *AAAI*.

Gerevini, A. E.; and Long, D. 2005. Plan Constraints and Preferences in PDDL3. Technical Report R. T. 2005-08-47, University of Brescia, Department of Electronics for Automation.

Ghosh, A.; Parikh, J.; Sengar, V. S.; and Haritsa, J. R. 2002. Plan Selection Based on Query Clustering. In *VLDB*.

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

Katz, M.; and Sohrabi, S. 2020. Reshaping Diverse Planning. In *AAAI*.

Katz, M.; Sohrabi, S.; and Udrea, O. 2020. Top-Quality Planning: Finding Practically Useful Sets of Best Plans. In *aaai*.

Katz, M.; Sohrabi, S.; and Udrea, O. 2022. Bounding Quality in Diverse Planning. In *AAAI*.

Katz, M.; Sohrabi, S.; Udrea, O.; and Winterer, D. 2018. A Novel Iterative Approach to Top-K Planning. In *ICAPS*.

Keyder, E.; Richter, S.; and Helmert, M. 2010. Sound and Complete Landmarks for And/Or Graphs. In *ECAI*.

Magnaguagno, M. C. 2020a. Classical Plan Viewer. https://maumagnaguagno.github.io/Classical_Plan_Viewer.

Magnaguagno, M. C. 2020b. HTN Plan Viewer. https://maumagnaguagno.github.io/HTN_Plan_Viewer.

Magnaguagno, M. C.; Pereira, R. F.; Móre, M. D.; and Meneguzzi, F. 2020. Web Planner: A Tool to Develop, Visualize, and Test Classical Planning Domains. In *Knowledge Engineering Tools and Techniques for AI Planning*.

Mantik, S.; Li, M.; and Porteous, J. 2022. A preference elicitation framework for automated planning. *Expert Syst. Appl.*, 208: 118014.

McDermott, D. 1998. First International Planning Competition (IPC-1). https://ipc98.icaps-conference.org/.

Muise, C. 2023. Planning.Domains. http://planning.domains.

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*, 190: 1–31.

Porteous, J.; Sebastia, L.; and Hoffmann, J. 2001. On the Extraction, Ordering, and Usage of Landmarks in Planning. In *ECP*, 174–182.

Riabov, A.; Sohrabi, S.; and Udrea, O. 2014. New Algorithms for the Top-K Planning Problem. In *The ICAPS Scheduling and Planning Applications Workshop (SPARK)*.

Richter, S.; Helmert, M.; and Westphal, M. 2008. Landmarks Revisited. In *AAAI*.

Rizk, Y.; Bhandwalder, A.; Boag, S.; Chakraborti, T.; Isahagian, V.; Khazaeni, Y.; Pollock, F.; and Unuvar, M. 2020. A Unified Conversational Assistant Framework for Business Process Automation. *AAAI Workshop on Intelligent Process Automation*.

Sohrabi, S.; Riabov, A.; Katz, M.; and Udrea, O. 2018. An AI Planning Solution to Scenario Generation for Enterprise Risk Management. In *AAAI*.

Sohrabi, S.; Udrea, O.; Riabov, A.; and Hassanzadeh, O. 2020. Interactive Planning-Based Hypothesis Generation with LTS+ +. In *Knowledge Engineering Tools and Techniques for AI Planning*, 189–207.

Sohrabi, S.; Udrea, O.; and Riabov, A. V. 2014. Knowledge Engineering for Planning-Based Hypothesis Generation. *arXiv:1408.6520*.

Sreedharan, S.; Chakraborti, T.; Muise, C.; Khazaeni, Y.; and Kambhampati, S. 2020a. D3WA+ – A Case Study of XAIP in a Model Acquisition Task for Dialogue Planning. In *ICAPS*.

Sreedharan, S.; Chakraborti, T.; Rizk, Y.; and Khazaeni, Y. 2020b. Explainable Composition of Aggregated Assistants. In *ICAPS Workshop on Explainable AI Planning*.

Zhang, H.-C.; and Huang, S. 1994. A Fuzzy Approach to Process Plan Selection. *International Journal of Production Research*.