Interactive Plan Selection using Linear Temporal Logic, Disjunctive Action Landmarks, and Natural Language Instruction

Tathagata Chakraborti, Jungkoo Kang, Francesco Fuggitti, Michael Katz, Shirin Sohrabi

IBM Research, USA

Abstract

We present `Lemming` – a visualization tool for the interactive 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 demonstrate four different user experiences for this process, three of them based on the principle of using disjunctive action landmarks as guidance to cut down the set of choice points for the user, and one on the use of linear temporal logic (LTL) to impart additional constraints into the plan set using natural language (NL) instruction.

Code https://github.com/IBM/lemming
https://github.com/IBM/nl2ltl
Video https://youtu.be/LnUJwA027O0

1 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 best 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) upfront. 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 (Srivastava et al. 2007; Nguyen et al. 2012; Vadlamudi and Kambhampati 2016; Katz and Sohrabi 2020; Katz, Sohrabi, and Udrea 2022).

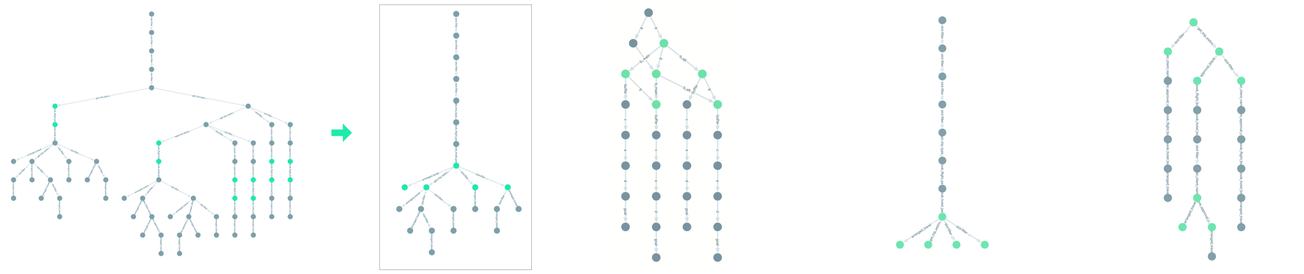
Recently, there have been several applications in which first multiple plans are generated and then the users are involved in the selection process. Some of these applications are in the area of patient monitoring (Sohrabi, Udrea, and Riabov 2014), enterprise risk management (Sohrabi et al. 2018), conversational 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 a larger sets of plans, while in these other applications (Sohrabi et al. 2020, 2018; Feblowitz et al. 2021) a custom user interface solution was implemented. In this paper, we present `Lemming`, a tool for providing a domain-independent approach to the plan disambiguation and selection problem by 1) using landmarks to help the user focus on a particular component of the search space; or 2) allowing the user to provide, in natural language, additional constraints to enforce on a set of plans.

Existing tools There are several tools that help with the specification (Muise 2023) and visualization of plans (Magnaguagno et al. 2020). While these tools visualize plans in various forms, their focus is on helping 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 the 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 (Hoffmann, Porteous, and Sebastia 2004) 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) for 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.

2 `Lemming` Overview

The user interaction with `Lemming` begins with a domain-problem pair and optionally with an already generated set of plans. Users can also generate plans using any planner that



(a) SELECT (b) NL2LTL (c) BUILD Forward (d) BUILD Backward

Figure 1: Different modes of plan disambiguation in Lemming. The green highlight indicates the next choice point for the user. Notice that LTL injection can lead to a new plan set (1b) while the other modes lead to successive reduction of the graph.

produces a set of solutions. We use (Katz et al. 2018) for this purpose, with the exception of (Speck, Mattmüller, and Nebel 2020) for the use case in Section 2.2 which requires support for domain axioms (Speck et al. 2019).[1]

2.1 Landmark Guided Plan Selection

Disjunctive action landmarks are actions that one or more of which must be present in a plan – for example, if A or B are disjunctive action landmarks then either A or B must be present in a plan that solves a given planning problem. Thus, they represent the choice points or points of ambiguity in how to solve a planning problem. For the disambiguation or selection problem, these naturally provide a hint on what information a user must navigate in order to make a selection of their preferred plan. We optimize for two objectives and end up with three ways to visualize a set of plans:

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.

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 either graphically or through language.

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. 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.

2.2 Natural Language Guided Plan Selection

The final view presents an integration of Lemming with a package NL2LTL (Fuggitti and Chakraborti 2023), presented at AAAI 2023, that helps translate natural language (NL) instruction to linear temporal logic (LTL) formulas. Natural language input is a key enabler of sequencing patterns in the industry (Chakraborti et al. 2022). While the package serves the general-purpose NL to LTL translation use case, in Lemming we demonstrate how it impacts the plan selection process in particular (and in contrast to the landmark-guided approach). In this mode, while still in the SELECT mode, the user imparts new rules to further constrain the set of plans instead of interacting graphically. These instructions are first translated into LTL formulas and then compiled into a new planning problem to produce a new set of plans. This has two implications compared to landmark-based selection: 1) the new plan set is not constrained to be a subset of the previous one, 2) we do need to call a planner after every new input. For the LTL to PDDL compilation, we use (Bonassi et al. 2023) but any existing approach (Bacchus and Kabanza 2000; Baier and McIlraith 2006b,a; Torres and Baier 2015) will suffice.

Limitations While landmarks make for a natural ally surfacing the most necessary (and potentially important) parts of the planning task: 1) the worst-case (although unlikely) 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 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.

[1]The Lemming GUI is built on top of the Reagraph library (Good Code 2023b) which, along with the Reaflow library (Good Code 2023a), provides a powerful set of features for building interfaces to planners and planning applications.

References

Aha, D. W.; Molineaux, M.; and Ponsen, M. 2005. Learning to Win: Case-Based Plan Selection in a Real-Time Strategy Game. In *ICCBR*.

Bacchus, F.; and Kabanza, F. 2000. Using temporal logics to express search control knowledge for planning. *AIJ*.

Baier, J. A.; and McIlraith, S. A. 2006a. Planning with First-Order Temporally Extended Goals using Heuristic Search. In *AAAI*.

Baier, J. A.; and McIlraith, S. A. 2006b. Planning with Temporally Extended Goals Using Heuristic Search. In *ICAPS*.

Bonassi, L.; De Giacomo, G.; Favorito, M.; Fuggitti, F.; Gerevini, A.; and Scala, E. 2023. Planning for Temporally Extended Goals in Pure-Past Linear Temporal Logic. In *ICAPS*.

Brachman, M.; et al. 2022. A Goal-driven Natural Language Interface for Creating Application Integration Workflows. In *AAAI Demo 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 Demo Track*.

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*.

Chen, D.; and Mooney, R. 2011. Learning to Interpret Natural Language Navigation Instructions from Observations. In *AAAI*.

Das, M.; Odom, P.; Islam, M. R.; Doppa, J. R.; Roth, D.; and Natarajan, S. 2019. Planning with Actively Eliciting Preferences. *Journal of Knowledge-Based Systems*.

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*.

Fuggitti, F.; and Chakraborti, T. 2023. NL2LTL – A Python Package for Converting Natural Language (NL) Instructions to Linear Temporal Logic (LTL) Formulas. In *AAAI Demo*.

Gerevini, A.; and Long, D. 2005. Plan Constraints and Preferences in PDDL3. Technical report, Department of Electronics for Automation, University of Brescia, Italy.

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

Good Code. 2023a. Node-based Visualizations for React. https://reaflow.dev. Accessed: 2023-12-12.

Good Code. 2023b. WebGL Network Graphs for React. https://reagraph.dev. Accessed: 2023-12-12.

Grover, S.; Sengupta, S.; Chakraborti, T.; Mishra, A. P.; and Kambhampati, S. 2020. RADAR: Automated Task Planning for Proactive Decision Support. *JHCI*.

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

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*.

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. *Knowledge Engineering Tools and Techniques for AI Planning*.

Mantik, S.; Li, M.; and Porteous, J. 2022. A Preference Elicitation Framework for Automated Planning. *Journal of Expert Systems with Applicaions*.

Mataric, M. 1992. Integration of Representation into Goal-Driven Behavior-Based Robots. *T-RO*.

Muise, C. 2023. Planning.Domains: A collection of tools for working with planning domains. https://editor.planning.domains. Accessed: 2023-12-12.

Muise, C.; Chakraborti, T.; Agarwal, S.; Bajgar, O.; Chaudhary, A.; Lastras-Montano, L. A.; Ondrej, J.; Vodolan, M.; and Wiecha, C. 2019. Planning for Goal-Oriented Dialogue Systems. *arXiv:1910.08137*.

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*.

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)*.

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 (IPA)*.

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++. *Knowledge Engineering Tools and Techniques for AI Planning*.

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

Speck, D.; Geißer, F.; Mattmüller, R.; and Torralba, Á. 2019. Symbolic Planning with Axioms. In *ICAPS*.

Speck, D.; Mattmüller, R.; and Nebel, B. 2020. Symbolic Top-k Planning. In *AAAI*.

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*.

Srivastava, B.; Nguyen, T. A.; Gerevini, A.; Kambhampati, S.; Do, M. B.; and Serina, I. 2007. Domain Independent Approaches for Finding Diverse Plans. In *IJCAI*.

Torres, J.; and Baier, J. A. 2015. Polynomial-Time Reformulations of LTL Temporally Extended Goals into Final-State Goals. In *IJCAI*.

Vadlamudi, S. G.; and Kambhampati, S. 2016. A Combinatorial Search Perspective on Diverse Solution Generation. In *AAAI*.

Zhang, H.-C.; and Huang, S. 1994. A Fuzzy Approach to Process Plan Selection. *IJPR*.