Operator-Counting Heuristics in HTN Planning

Daniela Kuinchtner, Felipe Meneguzzi
Pontifical Catholic University of Rio Grande do Sul (PUCRS)
daniela.kuinchtner@edu.pucrs.br, felipe.meneguzzi@pucrs.br

Abstract

Within the field of Artificial Intelligence (AI) planning, which deals with the automation of world-relevant problems, there are several approaches to solve planning tasks, two of the most popular are *classical planning* and *hierarchical planning*. Hierarchies are common structures used to understand and conceptualize the world. Hierarchical Task Network (HTN) planning is the branch that represents and handles hierarchies. Heuristic search can be the basis for the existing solvers in classical and hierarchical planning. However, techniques in classical planning, such as linear programming-based heuristics, are more sophisticated compared to hierarchical ones due to the classical planning research so far. In this paper, we propose to use classical operator-counting heuristics in hierarchical planning to improve the performance of HTN planners.

Introduction

Planning is the task of finding a course of action that achieves the goal(s) of an agent when we execute it. In its simplest form, the model underlying the task contains a description of the relevant parts of the environment the agent is planning for and a set of actions that describe how the agent can change it. Usually, a (finite) set of propositional variables describes the environment; the definition of an action contains a description of preconditions that need to be fulfilled to apply it and a set of effects (changes to the system it causes). The objective is to find a sequence of actions transforming the system into a state where certain state features hold. This process is often done using *heuristic search* (Höller et al. 2019).

Over the past two decades, heuristic search has established itself as the method of choice for optimal deterministic planning. This progress is due to the intense focus on developing domain-independent admissible heuristics, such as delete-relaxation, critical path, landmark, and operator-counting ones (Trevizan, Thiébaux, and Haslum 2018).

Hierarchical Task Network (HTN) planning is an AI planning technique that breaks with the tradition of classical planning (Nau, Ghallab, and Traverso 2004). The basic idea behind this technique includes an initial state description, a task network as an objective to be achieved, and domain knowledge consisting of networks of primitive and abstract tasks. A task network represents a hierarchy of tasks that can execute if the task is a primitive one, or if the task is an abstract one, it is decomposed into refined subtasks. The planning process starts by decomposing the initial task network and continues until it decomposes all abstract tasks; that is, we find a solution. The solution is a plan that equates to a set of primitive tasks applicable to the initial world state (Georgievski and Aiello 2015).

There are mainly two strategies to solve HTN planning problems: decomposition-based search in a plan-space, which searches a space of partial plans, and progression-based search in state-space. Progression search builds solutions in a forward manner, enabling a heuristic to base its calculation on a fully specified and continuously updated state. So far, plan-space search is the basis of most HTN planners that guide the search by using heuristic functions. Those systems represent the set of search nodes more effectively by maintaining a partial ordering between tasks, but they have only limited information about the current state during the search. Unlike plan-space search, progression search systems can calculate their heuristics incorporating the current state because the system tracks it during the search. Some existing progression-based systems do not rely on heuristics (e.g., SHOP2). However, current heuristic planners for standard HTN models (e.g., PANDA) recently introduced progression-based search (Höller et al. 2019).

PANDA (Höller et al. 2021) is a solver that uses progression search as a search-based algorithm. It is a framework that combines different techniques related to HTN planning. Some of the existing heuristics implemented on PANDA are: delete- and ordering-relaxation, landmark counting, additive, maximization, and fast-forward heuristics. Until now, there is no implementation of linear programming- or integer programming-based heuristics (e.g., operator counting), which are sophisticated and effective ones based on previous papers, such as (Meneguzzi, Pereira, and Pereira 2019), and (Pommerening et al. 2014).

In this work, we address the lack of the most promising heuristics in HTN planning (e.g., operator-counting) to improve the performance of existing HTN planners that uses progression search (e.g., PANDA). We aim to implement an LP/IP-based heuristic in the PANDA framework to find optimal and more efficient plans than other heuristics.

Technical Approach

To improve HTN solvers' performance that uses progression search, we consider using the information of more elaborated classical heuristics, such as LP/IP-based heuristics. PANDA (Höller et al. 2021) is a solver that uses progression search as a search-based algorithm and HDDL (Höller et al. 2020) as input language.

We aim to explore if the PANDA solver enables the use of more elaborated heuristics. To improve PANDA by enabling it to use operator-counting heuristics in HTN planning, first, we aim to guide research on how already implemented heuristics work in PANDA by testing with existing domains and reproducing the results of Holler's et al. paper (Höller et al. 2019). Second, we need to investigate how each heuristic makes the action choice, how much that choice matters, and how much the fringe order matters. Third, we aim to explain in algorithmic form how we can implement an operator-counting heuristic in HTNs. Then, if possible, we expect to implement a C++/Python operator-counting heuristic for HTNs.

To use PANDA, first, we need to parse the HDDL information to the planner. Second, as the framework only supports ground planning for the progression search, the model passes the grounding process before planning. Then, using progression search, the planner builds the plan in a forward manner. We aim to use the Relaxed Composition model (Höller et al. 2019), which is an approach to use classical heuristics in HTN by modeling the process of AND/OR trees into a classical planning problem. A classical heuristic applied to the RC model estimates the number of action and method applications necessary to transform the current search node into a plan. That approach results in informed heuristics for HTN planning. The model is used to create heuristics with interesting theoretical properties, hopefully, in our case, operator-counting heuristics.

Project Management

The macro tasks involving this project are (1) to study the PANDA solver and operator-counting heuristics; and (2) to implement or to suggest how to implement operator-counting heuristics in the PANDA framework. A schedule for accomplishing this work into micro-tasks weekly is:

- Week 1 (5/17): proposal presentation;
- Week 2 (5/24): understand the PANDA solver algorithm and how we use the base heuristics hAdd, hFF, and hLM-cut;
- Week 3 (5/31): choose a domain and run this domain with different heuristics;
- Week 4 (6/07): analyze how other heuristics can improve the PANDA solver, such as operator-counting ones. Study this heuristic;
- Week 5 (6/14): suggest algorithmic form ideas of how to implement the operator-counting heuristic. If possible, implement the operator-counting heuristic;
- Week 6 (6/21): write the final paper; and
- Week 7 (6/28): final paper submission and presentation.

The first macro task we aim to carry out by installing PANDA and reproducing Holler's et al. paper results by running different heuristics. The latter task consists of implementing a C++/Python operator-counting heuristic in the PANDA framework and comparing our results with the existing ones.

Conclusion

In this paper, we propose the use of more elaborated classical heuristics information, such as IP/LP-based heuristics, to improve the performance of HTN solvers that use progression search. Throughout this research, we aim to (1) carry out a study on how PANDA planner works, (2) understand how each existing heuristics in PANDA works, (3) study how operator-counting heuristics work, (4) implement a solution of this heuristic, and (5) show a time comparison of these heuristics. Thus, after developing the proposed solution, we aim to find optimal plans and improve the performance of the PANDA planner.

References

Georgievski, I.; and Aiello, M. 2015. HTN planning: Overview, comparison, and beyond. *Artificial Intelligence* 222: 124–156.

Höller, D.; Behnke, G.; Bercher, P.; and Biundo, S. 2021. The PANDA Framework for Hierarchical Planning. *KI - Künstliche Intelligenz* .

Höller, D.; Behnke, G.; Bercher, P.; Biundo, S.; Fiorino, H.; Pellier, D.; and Alford, R. 2020. HDDL: An Extension to PDDL for Expressing Hierarchical Planning Problems. *Proceedings of the AAAI Conference on Artificial Intelligence* 34(06): 9883–9891.

Höller, D.; Bercher, P.; Behnke, G.; and Biundo, S. 2019. On Guiding Search in HTN Planning with Classical Planning Heuristics. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 6171–6175. International Joint Conferences on Artificial Intelligence Organization.

Meneguzzi, F.; Pereira, A. G.; and Pereira, R. F. 2019. Robust Goal Recognition with Operator-Counting Heuristics. In *In ICAPS 2019 Workshop on Explainable AI Planning (XAIP@ICAPS)*.

Nau, D.; Ghallab, M.; and Traverso, P. 2004. *Automated Planning: Theory amp; Practice*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Pommerening, F.; Röger, G.; Helmert, M.; and Bonet, B. 2014. LP-Based Heuristics for Cost-Optimal Planning. In *Proceedings of the Twenty-Fourth International Conferenc on International Conference on Automated Planning and Scheduling*, ICAPS'14, 226–234. AAAI Press.

Trevizan, F.; Thiébaux, S.; and Haslum, P. 2018. Operator Counting Heuristics for Probabilistic Planning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 5384–5388. International Joint Conferences on Artificial Intelligence Organization.