The PANDA Progression System for HTN Planning in the 2023 IPC

Daniel Höller

Saarland University, Saarland Informatics Campus,
Saarbrücken, Germany
hoeller@cs.uni-saarland.de

Abstract

The PANDA Progression System is an HTN planning system that can handle both totally ordered and partially ordered HTN models. It performs a progression search, i.e., it only processes tasks without predecessor in the task network. PANDA uses a graph search and guides search by using heuristics. The configurations for the IPC use the families of Relaxed Composition (RC) heuristics and Delete- and Ordering-Relaxation (DOR) heuristics. RC heuristics relax the HTN model to a classical model and apply heuristics from classical planning to compute heuristic values. This way, also admissible heuristics for optimal planning can be created. The family of DOR heuristics originally capture delete- and ordering-free HTN planning as IP. This basic encoding can be extended by other IP constraints, e.g. encoding landmarks.

Introduction

The PANDA progression (PANDApro) system is a planner from the PANDA framework (Höller et al. 2021), which can handle both totally ordered and partially ordered models.

Search-based systems in HTN planning can be divided into plan space-based systems and progression-based systems (see Bercher, Alford, and Höller, 2019). The latter only process tasks without predecessor in the task ordering of the current task network. PANDApro uses the systematic progression search introduced by Höller et al. (2020).

It uses the common preprocessing stack of the PANDA framework: HDDL (Höller et al. 2020) as standard input language, followed by the grounding procedure introduced by Behnke et al. (2020).

During search, PANDApro maintains a black-list of already visited nodes and processes every node only a single time, i.e., it uses a graph search. While this is (from a computational perspective) no problem in totally ordered HTN planning, it gets a task as hard as graph isomorphism in partially ordered HTN planning. To do it efficiently, PANDApro uses the techniques introduced by Höller and Behnke (2021), which apply several techniques for hashing search nodes, and exploit certain special cases present in many models of the commonly used benchmark sets.

PANDApro guides its search by using heuristics estimating the goal distance (or the remaining costs in case of optimal planning). For the IPC, it uses two families of heuristics, which are described in the following.

RC Heuristics

The family of relaxed composition (RC) heuristics (Höller et al. 2018, 2019, 2020) uses classical heuristics to estimate the goal distance during HTN search. To do so, it relaxes the HTN model to a classical model which is only used for heuristic calculation. It is created in a way that the set of solutions increases compared to the HTN model. HTN planning starts with the initial task(s) and decomposes them until only actions are left. This process can be seen as the building process of a tree. The classical RC model maintains which tasks are part of that tree, but in a bottom-up manner, *compositing* tasks. When an action from the original HTN is applied in this model, it is marked as part of the tree. Methods are represented in the RC model by special actions. These are applicable when all subtasks of the method are part of the tree. When they are applied, the decomposed task is marked as part of the tree. The goal of the overall problem is to mark the tasks in the current task network as being part of the tree.

This encoding solves several problems when translating HTN models to classical models. First, we always have a state-based goal (which is not the case in HTN models): adding the current tasks to the tree. Second, the model is also informed about applicability of actions, since actions can only be added when they are applicable. Like in other HTN heuristics, the encoding allows for task insertion (adding further actions apart from the decomposition hierarchy) to make actions applicable that are needed elsewhere. However, what is interesting about our encoding when compared to other heuristics (see e.g. Bercher et al., 2017), is that the costs of these added actions are incorporated into the heuristic value. In our implementation, we further restrict task insertion to those actions still reachable via decomposing the current task network. Third, our heuristic is – to some extend – informed about the decomposition process, because the tree must be created up to the current tasks.

Practically, the model can be updated instead of recomputed. The only things that need to be changed are the initial state and the goal condition of the RC model. The model is linear in the size of the HTN model, and can be combined with any classical heuristic. However, the update of the goal is not possible (efficiently) in every classical heuristic.

In the IPC, we combine it with the Add (Bonet and Geffner 2001), the FF (Hoffmann and Nebel 2001), and the LM-Cut (Helmert and Domshlak 2009) heuristic. We have shown that the combination of the RC model with an admissible heuristic from classical planning results in an admissible HTN heuristic, so we use the latter (RC with LM-Cut) for optimal planning.

DOR Heuristics

In HTN planning, finding a delete-relaxed solution as done by many classical heuristics is still NP-hard (Alford et al. 2014). To make heuristic computation feasible, a common additional relaxation made by HTN heuristics is *task insertion*. As already discussed for RC heuristics, this means that the planner (or heuristic) is allowed to add actions apart from the hierarchy.

In our work on Delete- and Ordering-Free HTN planning (Höller, Bercher, and Behnke 2020), we introduce the class of HTN models that do not include delete-effects nor ordering constraints between tasks in the methods and in the initial task network. We show that the resulting problem is still NP-hard to solve. Then we show how to (exactly) encode this problem into an integer linear program (IP), combining constraints describing the decomposition process and constraints describing a relaxed planning graph (Imai and Fukunaga 2015).

The heuristic used here builds on this line of work. We combine (a further relaxed version of) the constraints describing HTN decomposition with constraints encoding HTN landmarks, and operator counting constraints (Pommerening et al. 2014). We use two configurations, one using LM-Cut landmarks generated on the RC model, and one using the AND/OR HTN landmarks introduced by Höller and Bercher (2021).

Instead of solving the IP, we further use the relaxation to a linear model to make computation polynomial.

References

Alford, R.; Shivashankar, V.; Kuter, U.; and Nau, D. S. 2014. On the Feasibility of Planning Graph Style Heuristics for HTN Planning. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS*. AAAI press.

Behnke, G.; Höller, D.; Schmid, A.; Bercher, P.; and Biundo, S. 2020. On Succinct Groundings of HTN Planning Problems. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, 9775–9784. AAAI Press.

Bercher, P.; Alford, R.; and Höller, D. 2019. A Survey on Hierarchical Planning – One Abstract Idea, Many Concrete Realizations. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, 6267–6275. IJCAI organization.

Bercher, P.; Behnke, G.; Höller, D.; and Biundo, S. 2017. An Admissible HTN Planning Heuristic. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 480–488. IJCAI organization.

Bonet, B.; and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence*, 129(1-2): 5–33.

Helmert, M.; and Domshlak, C. 2009. Landmarks, Critical Paths and Abstractions: What's the Difference Anyway? In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*.

Hoffmann, J.; and Nebel, B. 2001. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research*, 14: 253–302.

Höller, D.; and Behnke, G. 2021. Loop Detection in the PANDA Planning System. In *Proceedings of the 31st International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI Press.

Höller, D.; Behnke, G.; Bercher, P.; and Biundo, S. 2021. The PANDA Framework for Hierarchical Planning. *Künstliche Intelligenz*, 30(1): 11–20.

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. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, 9883–9891. AAAI Press.

Höller, D.; and Bercher, P. 2021. Landmark Generation in HTN Planning. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, 11826–11834. AAAI Press.

Höller, D.; Bercher, P.; and Behnke, G. 2020. Delete- and Ordering-Relaxation Heuristics for HTN Planning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, 4076–4083. IJCAI organization.

Höller, D.; Bercher, P.; Behnke, G.; and Biundo, S. 2018. A Generic Method to Guide HTN Progression Search with Classical Heuristics. In *Proceedings of the 28th International Conference on Automated Planning and Scheduling (ICAPS)*, 114–122. AAAI Press.

Höller, D.; Bercher, P.; Behnke, G.; and Biundo, S. 2019. On Guiding Search in HTN Planning with Classical Planning Heuristics. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 6171–6175. IJCAI organization.

Höller, D.; Bercher, P.; Behnke, G.; and Biundo, S. 2020. HTN Planning as Heuristic Progression Search. *Journal of Artificial Intelligence Research (JAIR)*, 67: 835–880.

Imai, T.; and Fukunaga, A. 2015. On a Practical, Integer-Linear Programming Model for Delete-Free Tasks and its Use as a Heuristic for Cost-Optimal Planning. *Journal of Artificial Intelligence Research*, 54: 631–677.

Pommerening, F.; Röger, G.; Helmert, M.; and Bonet, B. 2014. LP-Based Heuristics for Cost-Optimal Planning. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*. AAAI press.