The PANDADealer System for Totally Ordered HTN Planning in the 2023 IPC

Conny Olz[1], Daniel Höller[2], Pascal Bercher[3]

[1] Ulm University, Ulm, Germany
[2] Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
[3] The Australian National University, Canberra, Australia
conny.olz@uni-ulm.de, hoeller@cs.uni-saarland.de, pascal.bercher@anu.edu.au

Abstract

The PANDADealer system is an HTN planning system for solving totally ordered HTN planning problems. It builds on the heuristic progression search of the PANDApro system, and extends it with a look-ahead technique to detect dead-ends and inevitable refinement choices. The technique is based on inferred preconditions and effects of tasks, or more precisely, their decomposition methods.

Introduction

The PANDADealer (Dead-End Analysis with Look-Aheads and Early Refinements) system is a progression search-based planner that has been enhanced with a look-ahead technique based on inferred preconditions and effects of decomposition methods (Olz and Bercher 2023). It is specifically designed to solve totally ordered HTN planning problems. The system is build upon the PANDApro system and uses its pure heuristic search-based configurations (Höller 2023b) and also those using a combined heuristic- and landmark-based search guidance (Höller 2023a).

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 the first task in the task ordering of the current task network. PANDADealer is builds on the systematic progression search introduced by Höller et al. (2020) and uses the graph search described by Höller and Behnke (2021), i.e., it maintains a black-list of already visited search nodes to process every node only a single time.

The system 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).

The search is guided by heuristics, which estimate the distance to the goal (or the remaining costs in the case of optimal planning). Some configurations further exploit landmarks for search guidance. The most crucial component of PANDADealer that sets PANDADealer apart from PANDApro is its look-ahead technique. This technique, which detects dead-ends and inevitable refinement choices, is described in the work by Olz and Bercher (2023). We provide a brief overview of it in the next section, followed by a discussion on the heuristics and landmarks used.

Look-Ahead Technique

The look-ahead technique (Olz and Bercher 2023) employed in PANDADealer is based on inferred preconditions and effects of decomposition methods (Olz, Biundo, and Bercher 2021). These preconditions and effects are derived from the primitive tasks within the refinements of a method. Preconditions specify the facts that must hold in the state before executing the refinements, while effects indicate the changes in the state (additions or deletions) that occur after execution. Calculating the exact sets of preconditions and effects is computationally expensive; therefore, we only calculate a relaxed version in a preprocessing step, which disregards the executability of the refinements.

During the actual search, we treat the task network for each search node as a sequence of primitive tasks, where the compound tasks are enriched with inferred preconditions and effects. Starting from the first task, we check the preconditions of its methods in relation to the current state. For the "applicable" methods, we add all possible positive effects and remove the guaranteed negative effects, resulting in a new state. The new state is then used to evaluate the preconditions of the methods associated with the second task, propagating their effects in a similar manner. This process continues until the end of the task network. If the preconditions of a primitive task are not satisfied or no method of a compound task is applicable in its respective state, the search node is pruned as it forms a dead-end. If this is not the case but if a compound task has only one applicable method, we immediately decompose that task to eliminate future branching points. Further be aware that this "early application" of methods might help getting better heuristic estimates, because heuristics might not be able to detect that there is only a single applicable method.

For a comprehensive and detailed explanation of the look-ahead technique we refer to the respective work by Olz and Bercher (2023).

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. This is done based on a relaxation of the HTN model to a classical model. This model is only used for heuristic calculation. It is created in a way that the set of solutions increases compared to the

config	search	2. fringe	heuristic
agile-1	GBFS	no	rc(add)
agile-2	GBFS	no	rc(ff)
agile-lama	GBFS	LM-Cut	rc(add)
sat-1	GBFS	no	rc(add)
sat-2	$A^\star$, weight: 2	no	rc(ff)
optimal	$A^\star$	no	rc(lmc)

Table 1: Overview over the participating configurations.

track	config	ranking	IPC score
agile	agile-1	1.	11.74
	agile-lama	2.	11.68
	agile-2	4.	11.29
satisficing	agile-lama	1.	15.29
	sat-1	3.	14.89
	sat-2	6.	11.34
optimal	optimal	1.	8.33

Table 2: Rankings of participating configurations.

HTN model (which guarantees certain theoretical properties). 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 RC model captures (a relaxation of) the building process of that tree in the state of the classical model, but in a bottom-up manner, *compositing* tasks.

The RC model is computed once in a preprocessing step and updated during search. It is linear in the size of the HTN model and can be combined with arbitrary classical planning heuristics. 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. Höller et al. (2018) 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.

Landmarks

We further combined our PANDADealer system with landmark-based techniques for search guidance from the PANDA λ system (Höller 2023a), which also participated in the IPC. PANDA λ comes with two different techniques for landmark generation, one based on AND/OR graphs (Höller and Bercher 2021) and one based on the RC model. While we also support both, we used the latter one in the IPC.

Similar to the LAMA system from classical planning (Richter and Westphal 2010), our "lama" configuration combines heuristic-based and landmark-based guidance in a multi-fringe search, where one fringe is sorted by a heuristic, and one by an LM-count heuristic computed on the landmarks. The system extracts nodes from the fringes in turn and each successor node is inserted into both fringes with the respective heuristic estimate. The landmarks are generated using the LM-Cut (Helmert and Domshlak 2009) heuristic on the RC model of the initial search node. The generated landmarks are stored and tracked during search.

Configurations and Results

PANDADealer won all total-order HTN tracks of the IPC 2023. In Table 1 we give an overview over the configurations. Table 2 shows their rankings and IPC score.

In the agile track, the objective was to find a plan as quickly as possible within a 30-minute timeframe. The score for a solved task is $min\{1, 1 - log(t)/log(1800)\}$, where t is the time it took to solve the task. The cost of the plan was not taken into account. Results show that all configurations of PANDApro, including those with and without landmarks, and with and without our look-ahead technique, performed notably better than other competing planners. However, PANDADealer had a marginal advantage with two configurations, specifically agile-1 and agile-lama. For a detailed analysis of when the look-ahead technique proves beneficial, we refer to the work of Olz and Bercher (2023).

The satisficing track was a new addition to the HTN planning track. Here, the goal was to find the most cost-effective plan within the same 30-minute limit. The score for a solved task is the ratio C^*/C, where C is the cost of the discovered plan and C^* is the cost of a reference plan. For the configurations sat-1 and sat-2, we made slight modifications to the search engine: instead of stopping after finding the first plan, we continued the search until the time limit is reached. Whenever a new plan was found, we updated the returned plan if the new one had lower costs. Interestingly, we won this track with our configuration agile-lama, which was not further optimized for the satisficing track as it returned only the first plan and stopped the search afterwards. The landmarks appear to improve plan quality, but a separate evaluation needs to be conducted to verify this assumption.

For the optimal track, we competed with only one configuration, which also outperformed the other planners.

References

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.

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)*, 162–169. AAAI Press.

Hoffmann, J.; and Nebel, B. 2001. The FF Planning System:

Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research (JAIR)*, 14: 253–302.

Höller, D. 2023a. The PANDA λ System for HTN Planning in the 2023 IPC. In *Proceedings of the 11th International Planning Competition: Planner Abstracts – Hierarchical Task Network (HTN) Planning Track (IPC)*.

Höller, D. 2023b. The PANDA Progression System for HTN Planning in the 2023 IPC. In *Proceedings of the 11th International Planning Competition: Planner Abstracts – Hierarchical Task Network (HTN) Planning Track (IPC)*.

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)*, 168–173. AAAI Press.

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.; 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.

Olz, C.; and Bercher, P. 2023. A Look-Ahead Technique for Search-Based HTN Planning: Reducing the Branching Factor by Identifying Inevitable Task Refinements. In *Proceedings of the 16th International Symposium on Combinatorial Search (SoCS)*, 65–73. AAAI Press.

Olz, C.; Biundo, S.; and Bercher, P. 2021. Revealing Hidden Preconditions and Effects of Compound HTN Planning Tasks – A Complexity Analysis. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 11903–11912. AAAI Press.

Richter, S.; and Westphal, M. 2010. The LAMA Planner: Guiding Cost-Based Anytime Planning with Landmarks. *Journal of Artificial Intelligence Research (JAIR)*, 39: 127–177.