New HTN Domains in the 2023 IPC

Gregor Behnke[1], **Jane Jean Kiam**[2], **Dominik Schreiber**[3]

[1]University of Amsterdam, Netherlands, g.behnke@uva.nl
[2]Universität der Bundeswehr, Munich, Germany, jane.kiam@unibw.de
[3]Karlsruhe Institute of Technology, Germany, dominik.schreiber@kit.edu

Abstract

We present four new hierarchical planning domains named Colouring, Lamps, SharpSAT, and Ultralight-Cockpit.

Colouring

Coloring is a Partially Ordered domain authored by G. Behnke. It encodes a version of the tiling problem (van Emde Boas 2019), which is frequently used for complexity reductions. Given a set of available tiles, each having a color at one of its edges, the task is to fill an $n \times n$ square with these tiles, s.t., touching edges have the same color. The outer edge has no required color. This problem is NP-complete for unary encoded n. The encoding uses the idea of proof encoding double-exponentially time-bounded Turing Machine (Alford, Bercher, and Aha 2015). The colouring is determined by a sequence of actions generated by a totally-ordered decomposition, which only checks the touching colours in the left and right directions. The HTN's partial order is used to simulate a memory of arbitrary size that keeps track of the up-facing colours of each line to check whether the up/down touching tiles are the same colour.

Lamps

The Totally Ordered domain *Lamps*, authored by G. Behnke, models a variant of the game "Lights Out," which is about an $n \times m$ field with lamps that can either be on or off. Switching a lamp forces all horizontally and vertically connected lamps of the same status (on or off) to also toggle. This reachability-based procedure can be easily modeled with an HTN, but is hard to express using classical planning.

SharpSAT

The Totally Ordered domain *SharpSAT*, authored by D. Schreiber, models the problem #SAT, or *(exact) model counting*. This problem is to count the number of different variable assignments ("models") which satisfy a given propositional formula (Gomes, Sabharwal, and Selman 2021). The complexity class of #SAT, named #P, is (handwavingly) somewhere between NP and PSPACE and therefore not considered as hard as HTN planning. Nevertheless, a hierarchical planning model for #SAT is appealing due to the problem's natural hierarchical structure, its rather simple formulation, and a number of interesting search properties (see below).

We express instances of #SAT as TOHTN planning instances. Our hierarchical model is based on the straight forward recursive CDP algorithm (Birnbaum and Lozinskii 1999), which is a modified DPLL search procedure (Davis, Logemann, and Loveland 1962). When finding a model at decision depth d, the procedure does not terminate (as DPLL) but instead adds $2^{|V|-d}$ to its global model count and backtracks. The only decisions a planner can make is choosing a literal to branch on (or, if unit clauses are present, which one to satisfy first). The domain has no dead-ends (DPLL backtracking is performed with explicit subtasks); however, the branching choices a planner makes can result in substantial differences with respect to the effective search space size. As such, informative heuristics and/ or restarts with different decision-making have the potential to make a big difference. The hierarchy's depth is limited to $\mathcal{O}(|V| + |C|)$ levels until all variables have been assigned.

Arbitrarily difficult benchmarks can be generated from DIMACS CNF instances, e.g., benchmarks from the International SAT Competitions[1] or randomly generated difficult 3-SAT instances. A found plan can be transformed to an actual model count in linear time using an associated decoder script. This linear-time procedure just looks for specific actions `A_OUTPUT_EXPONENTIAL_COUNT d` and adds $2^{|V|-d}$ to a model counter for each such action.

Ultralight-Cockpit

Ultralight-Cockpit is a Partially Ordered domain authored by J. J. Kiam and P. Jamakatel (Kiam and Jamakatel 2023), and is motivated by its application on a Pilot Assistance System (PAS) for single-pilot ultralight aircraft. It models various tasks to be performed by a private pilot, while focusing on tasks necessary for handling emergency situations. Modeling the tasks in HTN is natural, as instructions documented in pilot operating handbooks (e.g. SHARK (2017); Pooley (2003)), are sequences of abstract or primitive tasks, without reference to reachable states.

The HTN model of the Ultralight-Cockpit domain is mainly used for two purposes: for generating instructions in

[1]https://satcompetition.github.io/2022/downloads.html

form of a task plan to be displayed on the cockpit as guidance for the private pilot in distress (Jamakatel and Kiam 2024), as well as for automated pilot observation to recognize the pilot's goal task (Jamakatel et al. 2023) using the Plan and Goal Recognition (PGR) technique for HTN planning problems developed by Höller et al. (2018). With an automated goal recognition, the PAS can intervene or guide the pilot without asking continuously for the pilot's intent during an emergency. By doing so, the PAS avoids increasing the pilot's mental workload, as private pilots in general do not undergo intensive training or strict health screening to ensure their capabilities for coping with emergencies.

Unexpected precautionary landing may be necessary when anomalies arise. In this use case, to land an ultralight aircraft safely, the emergency landing site is still required to fulfill certain conditions such as being free of obstacles and within reach of the aircraft. Besides, the surface of the site as well as its slope gradient must be reasonable for landing. For the sake of benchmarking HTN-planners, instances of arbitrary numbers of landing sites with randomly set conditions (i.e. distance to aircraft, presence of obstacles, condition of the surface and the slope) can be generated.

References

Alford, R.; Bercher, P.; and Aha, D. 2015. Tight bounds for HTN planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 25, 7–15.

Birnbaum, E.; and Lozinskii, E. L. 1999. The good old Davis-Putnam procedure helps counting models. *Journal of Artificial Intelligence Research*, 10: 457–477.

Davis, M.; Logemann, G.; and Loveland, D. 1962. A machine program for theorem-proving. *Communications of the ACM*, 5(7): 394–397.

Gomes, C. P.; Sabharwal, A.; and Selman, B. 2021. Model counting. In *Handbook of satisfiability*, 993–1014. IOS press.

Höller, D.; Behnke, G.; Bercher, P.; and Biundo, S. 2018. Plan and Goal Recognition as HTN Planning. In *Proceedings of the 30th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, 466–473. Volos, Greece: IEEE Computer Society.

Jamakatel, P.; Bercher, P.; Schulte, A.; and Kiam, J. J. 2023. Towards Intelligent Companion Systems in General Aviation using Hierarchical Plan and Goal Recognition. In *Proceedings of the 11th International Conference on Human-Agent Interaction (HAI 2023)*. Association for Computing Machinery.

Jamakatel, P.; and Kiam, J. J. 2024. Generation and Communication of Strategic Plans at Different Levels of Abstraction for Intelligent Assistance Systems in General Aviation. In *34th Congress of the International Council of the Aeronautical Sciences (ICAS)*.

Kiam, J. J.; and Jamakatel, P. 2023. Can HTN Planning Make Flying Alone Safer? In *Proceedings of the 6th ICAPS Workshop on Hierarchical Planning (HPlan 2023)*, 44–48.

Pooley, D. 2003. *POOLEYS Private Pilots Manual: JAR Flying Training, Volume 1*. Cranfield, UK: POOLEYS.

SHARK. 2017. *Flight Manual: UL airplane*. SHARK.AERO CZ s.r.o.

van Emde Boas, P. 2019. The convenience of tilings. In *Complexity, Logic, and Recursion Theory*, 331–363. CRC Press.