Historical Developments in Artificial Intelligence Planning and Search

Hierarchical Task Networks

Hierarchical Task Networks (HTN) based planners are similar to Standford Research Institute Problem Solver (STRIPS) based planners using Problem Domain Definition Language (PDDL), as seen in Classical Planners. However, Classical Planners generate a combination for every action, object and state defined in a problem domain – as seen in the code implementation of the Planning project. HTN aims to only define the tasks and subtasks that fit to the given problem definition and produce a plan of tasks to execute. HTN planning defines tasks to complete instead of ending in a goal state, tasks are defined that may be decomposed with methods resulting in subtasks, known as Information Hiding [2], and the constraints for each task are enforced, not allowing certain tasks to decompose/execute if the constraints are not met. Information hiding reduces the complexity of the planning algorithm, implying that the execution time will be lower than that of Classical Planning. HTN decomposes tasks until primitive tasks are available, which can be executed by planning operators. Therefore, logically, HTN would perform less computations, as the search space is smaller than that of Classical Planners. HTN still produces an optimal result, but quicker than Classical Planners, specifically STRIPS using PDDL as used in the Planning project. [2,3,4]

Simple Hierarchical Ordered Planners

Classical planners using A* search with heuristics are resource intensive, as seen in the Planning project analysis. Simple Hierarchical Ordered Planners (SHOP) aim to reduce the amount of resources used. SHOP implements HTN, however, produces a plan in the order that the tasks will be executed. This allows Depth First Search to be used efficiently [1], implying less resources used and an optimal plan is produced. SHOP also makes use of axioms to apply to the planning operators to make the plan more certain of the domain specified. This reduces the complexity as the plan is in a linear order (totally-ordered plan), and reduces the amount of resources used to result in the optimal plan. Hence, SHOP is a further improvement to HTN and Classical Planning. [1,5]

Encoding Planning Problems as Satisfiability Problems

Encoding Planning Problems as Satisfiability Problems (SAT) translates a planning problem into propositional logic formulae, satisfiability problems. SAT guesses the plan length by using iterative deepening and testing the satisfiability problem. If the problem is satisfied, an optimal plan can be produced with the length associated with that particular iteration. Once a length is found, all of the planning problem domain is encoded into satisfiability problems. There are various ways to improve the efficiency of the satisfiability problems. From the list of encoded actions, there will only be one that produces the desired result. Finding that single action that satisfies the problem requires using a satisfiability algorithm. The satisfiability algorithm can be a Systematic Search such as DP (Davis Putnam Logemann Loveland) or local search algorithms such as GSAT, Walksat, greedy search, hill climbing, etc. It is noted that SAT uses too many resources and takes long to execute when used in isolation. However, using SAT in conjunction with Planning Graphs results in a feasible and efficient solution to the problem. [6,7]

Bibliography

1. AIGameDev.com, Alex J. Champandard. 2008. *SHOP: Simple Hierarchical Ordered Planner*. [ONLINE] Available at: https://aigamedev.com/open/article/shop-htn/. [Accessed 10 April 2017].
2. Wikipedia. 2016. *Hierarchical task network*. [ONLINE] Available at: https://en.wikipedia.org/wiki/Hierarchical_task_network. [Accessed 10 April 2017].
3. Michigan Tech. 2010. *Hierarchical Task Network (HTN) Planning*. [ONLINE] Available at: http://www.cs.mtu.edu/~nilufer/classes/cs5811/2010-fall/lecture-slides-3e/cs5811-ch11b-htn. [Accessed 10 April 2017].
4. Dana S. Nau, University of Maryland. 2012. *Hierarchical Task Network Planning*. [ONLINE] Available at: https://www.cs.umd.edu/~nau/planning/slides/chapter11.pdf. [Accessed 10 April 2017].
5. University of Maryland. 2006. *SHOP (Simple Hierarchical Ordered Planner)*. [ONLINE] Available at: https://www.cs.umd.edu/projects/shop/description.html. [Accessed 10 April 2017].
6. The University of Edinburgh. 2004. *SAT-Based Planning*. [ONLINE] Available at: http://www.inf.ed.ac.uk/teaching/courses/plan/slides/SAT-Planning-Slides.pdf. [Accessed 10 April 2017].
7. Yiqiao Wang, University of Toronto. 2004. *Compilation of Planning to SAT*. [ONLINE] Available at: http://www.cs.toronto.edu/~sheila/2542/w06/readings/yiqiao.pdf. [Accessed 10 April 2017].