Research Review on Historical Developments in AI Planning

I chose the following methods and algorithms to resume:

1. PDDL

The first important action description language for automated planning was STRIPS, developed in 1971 by Richard Fikes and Nils Nilsson in Stanford.[1] Other researchers later improved on its ideas and introduced new, more complete languages, such as when Edwin Pednault proposed the ADL in 1987 to handle some of its shortcomings, like the lack of conditional operations between literals.[2] But until 1998, researchers didn't have a language with a strong enough formalism and semantics to describe planning across many different fields.

The PDDL was created by Drew McDermott as an attempt to unify and standardize AI planning language, motivated by the need to have all competitors participating in the 1998/2000 International Planning Competition (IPC) to use the same semantics for fair comparison between the projects. From the abstract in the original paper: *"Our hope is to encourage empirical evaluation of planner performance, and development of standard sets of problems, all in comparable notations."* [3]

The language continues to be developed across subsequent versions of the IPC and the ICAPS conference. It is today in version 3.1, having added new features such as *derived predicates* and *object-fluents*. PDDL had a big impact on the sharing of research in the field, and is still an active area of research.[4]

2. HSP – Heuristics Search Based Plans

The Heuristics Search-Based Plans was developed by Blai Bonnet and Hector Geffner in 1998, based on the idea of heuristics search. The idea is that as planning problems can be mapped into search problems in a suitable space, which can then be solved using a heuristic function extracted *automatically* from the problem encoding.[5]

A second version was developed in 2001, developed as a more general platform for experimenting with different state spaces and different heuristics. HSP was the first to apply heuristic search in large-scale planning problems, having significant results in version 2 and influencing the development of heuristics search in AI planning across the 2000's.[6]

3. SAT Planning

SAT-plan systems, or "Planning as Satisfiability", were created to convert any planning problem instance into an instance of the Boolean satisfiability problem, which is then solved using any method for establishing satisfiability. Although SAT problems are NP-complete and there is no known algorithm that efficiently solves each SAT problem, and it is generally believed that no such algorithm exists, current heuristical SAT-algorithms are able to solve problem instances involving tens of thousands of variables and formulas consisting of millions of symbols, which is sufficient for many practical SAT problems. [7] [8]

Some of the most famous algorithms used for SAT-planning were described by David G. Mitchel in a 2005 paper "A SAT Solver Primer": Backtracking, DPLL (Davis-Putnam algorithm), CDCL, and BCP (unit propagation). [9]

References

[1] https://en.wikipedia.org/wiki/STRIPS

[2]Pednault. « Formulating multi-agent dynamic-world problems in the classical planning framework ». Morgan Kaufmann, San Mateo, CA, 1987.

[3]McDermott, Drew; Ghallab, Malik; Howe, Adele; Knoblock, Craig; Ram, Ashwin; Veloso, Manuela; Weld, Daniel; Wilkins, David (1998). "PDDL---The Planning Domain Definition Language". Technical Report CVC TR98003/DCS TR1165. New Haven, CT: Yale Center for Computational Vision and Control.

[4] http://www.icaps-conference.org/

[5] Blai Bonet and Hector Geffner, «HEURISTIC SEARCH PLANNER 2.0 », AI Magazine Volume 22 Number 3 (2001) (© AAAI)

[6] https://towardsdatascience.com/ai-planning-historical-developments-edcd9f24c991

[7] https://en.wikipedia.org/wiki/Boolean_satisfiability_problem

[8] Jussi Rintanen, “Heuristics for Planning with SAT”, NICTA and the Australian National University Canberra, Australia

[9] D. Mitchell, A SAT Solver Primer, EATCS Bulletin, 85, pp. 112-133, February 2005.