History of AI planning and search

AI planning started from investigations into state-space search and from practical need of new applications like robotics and scheduling in other domains. The first major planning system was STRIPS (Stanford Research Institute Problem Solver, Files and Nilsson, 1971) for the Shakey robot project at Stanford Research Institute(SRI) in 1971. The control structure was modeled on GPS (General Problem Solver, Newell and Simon, 1961) a state-space search system that used means-end analysis. STRIPS used an earlier more restricted version of the classical planning language in use today - Action Description language or ADL (Pednault, 1986). The planners then used totally ordered action sequences where a sub-plan was computed for each subgoal and the sub-plans were then strung together. This is called linear planning but this technique was found incomplete later on (Sussman anomaly, 1975). A complete planner must use interleaving of actions from different subplans within a single sequence. Goal regression planning re-orders steps in a totally ordered plan so as to avoid conflict between subgoals (Waldinger, 1975).

Partial order planning dominated the next 20 years of planning research into the 90s. The first clear formal exposition was TWEAK(Chapman, 1987) a planner that allowed proofs of completeness and intractability of various planning problems, followed by SNLP and UCPOP. PDDL(Problem Domain Description language) was an attempt to standardize AI planning languages was introduced as a computer-parsable syntax for representing planning problems and has been in use from the late 90s.

With the GRAPHPLAN system (Avrim Blum, Merrick Furst (1995, 1997)), the partial order planners were exceeded in speed by several orders of magnitude. Graphplan takes as input a planning problem expressed in STRIPS and produces if possible a sequence of operations for reaching a goal state. The name is given due to the use of a novel planning graph to reduce the amount of search needed to find the solution otherwise from straightforward exploration of the state space graph.

With the introduction of the ignore-delete-list heuristic in UNPOP(McDermott, 1996), state space planning could again compete with partial order planning for large planning problems. The most successful state space planner is FF (Hoffman, 2001; Hoffman and Nebel, 2001; Hoffman 2005). Faster methods emerged than partial order planning with accurate heuristics derived from a planning graph and REPOP(Nguyen and Kambhampati, 2001) which scales even better than GRAPHPLAN to be competitive with the fastest state space planners.

The BLACKBOX planner which combines ideas from GRAPHPLAN and SATPLAN was developed by Kautz and Selman (1998). SATPLAN (Planning as Satisfiability) converts the planning problem instance into an instance of the Boolean satisfiability problem, which is then solved using a method for establishing satisfiability (example, WALKSAT).

Helmert(2001) analyzes several classes of planning problems, and shows that constraint-based approached are best for NP-hard problems while search based approaches do better in domains where solutions can be found without backtracking. Rather than instantiating propositionalized actions before the search begins, generating them dynamically only on as needed basis can help solve problems with large search space and objects. This cross-pollination between search such as STRIPS and constraint based approaches such as GRAPHPLAN with SATPLAN depending on the state space size and complexity is still the approach today as there is no one method that is found optimal in all state space types.

References:

Fikes, R.E. and Nilsson, N.J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *AIJ*, 2(3-4), 189-208

Newell, A. and Simon, H.A. (1961). GPS, a program that simulates human thought. In Billing, H. (Ed.), *Lernende Automaten*, pp. 109-124. R. Oldenbourg

Pednault, E.P.D. (1986). Formulating multiagent, dynamic-world problems in the classical planning framework. *In Reasoning about Actions and Plans: Proc. 1986 Workshop,* pp 47-82

Sussman, G. J. (1975). *A Computer Model of Skill Acquisition*. Elsevier/North-Holland.

Waldinger, R. (1975). Achieving several goals simultaneously. In Elcock, E. W. and Michie, D. (Eds.), *Machine Intelligence 8*, pp. 94-138. Ellis Horwood.

Chapman, D. (1987). Planning for conjunctive goals. *AIJ*, 32(3), 333-377

Blum, A.L. and Furst. M. (1995). Fast planning through planning graph analysis. In ICJAI-95, pp 1636-1642
Blum, A.L. and Furst. M. (1997). Fast planning through planning graph analysis. In AIJ-90(1-2), 281-300

McDermott, D. (1996). A heuristic estimator for means-end analysis in planning. In *ICAPS-96*, pp 142-149

Hoffman, J. (2001). FF: the fast-forward planning system. *AIMag*, 22(3), 57-62
Hoffman, J. and Nebel, B. (2001). The FF planning system: Fast plan generation through heuristic search*, JAIR, 14*, 253-302
Hoffman, J. (2005). Where "ignoring delete lists" works: Local search topology in planning benchmarks. *JAIR, 24*, 685-758

Nguyen. X. and Kambhampati, S. (2001). Reviving partial-order planning. In *IJCAI-01,* pp 459-466

Kautz, H. and Selman, B (1998). BLACKBOX: A new approach to the application of theorem proving to problem solving. Working Notes of the AIPS-98 Workshop on Planning as Combinatorial Search

Helmert, M. (2001). On the complexity of planning in transportation domains. In *ECP-01*