Research Review

Vatsal Srivastava

In the past five decades, there has been a great deal of contribution in the field of Artificial Intelligence towards the problem of Planning and Search. The aim of this article is to review some of these developments and their importance and impact to the discipline.

Stanford Research Institute Problem Solver (STRIPS)

Stanford researchers Richard E. Fikes and Nils J. Nilsson build the Stanford Research Institute Problem Solver (STRIPS) [1]. STRIPS attempts to find a sequence of operators in a space of world models to transform a given initial world model into a model in which a given goal formula can be proven to be true. STRIPS represents a world model as an arbitrary collection of first-order predicate calculus formulas and is designed to work with models consisting of large numbers of formulas. It employs a resolution theorem prover to answer questions of particular models and uses means-ends analysis to guide it to the desired goal-satisfying model.

The STRIPS system was built to solve a different class of problem from game-playing agents developed before that time, and the researchers were careful to define the type of problem and a syntax to describe it.

The STRIPS machine carried out this task with an adopted General Problem Solver (GPS) strategy. The strategy extracted differences between the present and goal state and applied operators that would reduce the differences. The STRIPS research was built on a PDP-10 mainframe computer which was common in universities at the time. This likely made the solver results easy to reproduce and measure against. Additionally, the clear syntax which the paper defined for states and actions established a language which the AI community could use to communicate problems in this domain. This likely led the "STRIPS problem domain" to be so named.

Fast Planning Through Planning Graph Analysis (Graphplan)

Graphplan [2] was created by Avrim Blum and Merrick Furst, with subsequent extensions and improvements made by many researchers at many different institutions around the world. **Graphplan** is a general-purpose planner for STRIPS-style domains, based on ideas used in graph algorithms. Given a problem statement, Graphplan explicitly constructs and annotates a compact structure called a Planning Graph, in which a plan is a kind of "flow" of truth-values through the graph. A Planning Graph encodes the problem such that useful constraints inherent in the problem are used to reduce the amount of search needed. These graphs have polynomial size and can be built in polynomial time. The output is similar to a parallel plan [3] in that the output consists of a series of steps each containing multiple sequence-independent actions. This graph has the property that useful information for constraining search can quickly be propagated through the graph as it is being built. Graphplan then exploits this information in the search for a plan.

Reviving Partial Order Planning (RePOP)

Presented by XuanLong Nguyen & Subbarao Kambhampati [5], this paper challenged the prevailing pessimism about the scalability of partial order planning (POP) algorithms [4] by presenting several novel heuristic control techniques that make them competitive with the state of the art plan synthesis algorithms. The key insight is that the techniques responsible for the efficiency of the currently successful planners–viz., distance based heuristics, reachability analysis and disjunctive constraint handling–can also be adapted to dramatically improve the efficiency of the POP algorithm. The ideas were implemented in a variant of UCPOP called REPOP1 . The empirical results show that in addition to dominating UCPOP, REPOP also convincingly outperforms Graphplan in several “parallel” domains. The plans generated by REPOP also tend to be better than those generated by Graphplan and state search planners in terms of execution flexibility.

References:

[1] Richard E. Fikes, Nils J. Nilsson. 1971. STRIPS: A New Approach to the Application of .Theorem Proving to Problem. Solving, IJCAI '71 Proceedings of the 2nd international joint conference on Artificial intelligence. Pages 608-620. http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf

[2] Avrim L Blum, Merrick L. Furst. 1997. Fast Planning Through Planning Graph Analysis. Artificial Intelligence. Pages 281-300. http://www.cs.cmu.edu/~avrim/Papers/graphplan.pdf

[3] C. Knoblock. Generating parallel execution plans with a partial order planner. AIPS 94, 1994. http://www.dtic.mil/dtic/tr/fulltext/u2/a285888.pdf

[4] Hoon Chang, J.A. Abraham. POP: an efficient Performance OPtimization algorithm based on integrated approach. http://ieeexplore.ieee.org/document/379724/

[5] X. Nguyen and S. Kambhampati. Reviving partial order planning. IJCAI'01 Vol. 1. 2001. http://rakaposhi.eas.asu.edu/ucpop-revive.pdf