

Udacity AIND Project: Historical Developments in AI Planning and Search (Research Review)

During the past, several years, AI planning has made major steps forward in terms of the size and difficulty of problems that can be solved. In this review, we give an overview of the techniques that have been part of the recent developments.

Planning emerged as a specific sub-field with the original work of Fikes and Nilsson [Fikes/Nilsson 1971] on the STanford Research Institute Problem Solver (STRIPS). The STRIPS project introduced a simple syntax for defining action schemas, in terms of the preconditions, add effects and delete effects of the action. Taking inspiration from STRIPS, the Action Description Language [ADL, Pednault, 1986], relaxed the restrictions of STRIPS and made it possible to encode real world problems. A computer-parsable standardized syntax was introduced for representing planning problems called, the Problem Domain Description Language or PDDL [Ghallab et al., 1998] was introduced in the late 90s and is still in use today. PDDL was inspired by both STRIPS and ADL.

Graphplan [Blum/Furs 1995] excited a great deal of interest when it was introduced because it constitutes an approach to planning that was radically new at the time (in the early 1990s). The first step of Graphplan is constructing the planning graph. Then using the planning graph, we extract the plan instead of just using it for providing heuristics. Following the significant success of Graphplan, interest in the planning problem was revitalized and other new ideas were explored. A very influential direction was initiated by work of McDermott in the planning system UNPOP [McDermott 1996] and Geffner and Bonet in HSP [Bonet et al. 1997]. The idea behind this work is to use a classic heuristic guided search. The novel contribution made by McDermott and by Geffner and Bonet was to demonstrate a method by which a surprisingly informative heuristic function could be constructed automatically, simply by analyzing the domain.

The planner PbR [Ambite/Knoblock 1997] was one of the first planners to demonstrate that plans could be obtained by means of iterative repair performed on a flawed initial plan, using rewriting rules selected by efficient local search techniques. This idea is explored in LPG [Gerevini/Serina 2002] using locally extending plan graphs. LPG does not handle arbitrary use of numbers, but manages actions with duration and concurrency. LPG replaces the Graphplan search with a far more efficient and more powerful local search technique. This involves identifying an initial candidate “plan” (which might, in fact, not even be executable) and refining it by generating alternative possible repairs or modifications to the candidate.

Over the past decade several new techniques have emerged as promising for future development of classical planning towards application. These include heuristic forward search based on informative heuristics that can be automatically generated and that exploit the structure of the problem; local search techniques that can be applied to efficiently repair a rapidly generated, flawed plan; hybrid systems that exploit structure and systems that go beyond the simple notion of batch planning and can anticipate execution-time discoveries