Research review of developments in AI planning and search

Nicholas Chen

Stanford Research Institute Problem Solver (STRIPS)

STRIPS is the first major planning system which combines techniques from state-space search, theorem proving and control theory [1, 2]. It represents states as a collection of first-order predicate formulas, and finds a sequence of operators which transform the initial state to the goal state. Unlike predecessors which solve problems by applying all applicable operators to the initial state till the goal is met (which can result in a very large search tree), STRIPS obtains "differences" between the current state and the goal state, and identifies "relevant" operators which reduce the "differences". While the algorithm behind STRIPS is a milestone, the action representation of STRIPS was far more influential – Most planning systems following it have used a variant of the STRIPS language.

Graphplan

Graphplan, an algorithm guaranteed to find the shortest plan, works in STRIPS-like domains but constructs a planning graph instead of a state-space graph [3]. This planning graph encodes the problem in a way that allows the inherent constraints of the problem reduce the search space, and can be constructed in polynomial time. It searches for plans by using backward chaining (regressive search), and reduces the search space by recording information of failed subgoals so that similar subtrees in the search tree will not be visited again. At the time of Graphplan's inception, most of AI planning research was focused on partial-order planners. Graphplan was different from these in that it incrementally increased the plan length until it finds one, and prunes the search tree using reachability information. As such, Graphplan was orders of magnitude faster than other partial-order planners, influencing researchers to look at techniques beyond the standard AI toolbox [4].

Point-based value iteration for solving POMDPs

In the robotics community, path planning is usually posed as a partially observable Markov decision process (POMDP) problem. The POMDP approach to planning differs from the previous two 'classical planning' approaches in that it assumes the world is probabilistic (hence the Markov Decision Process) and not fully observable – properties important to a robot with imperfect sensors. In a POMDP, at each time step, the agent is originally in an environment s. It takes an action a which causes the environment to transit to other states with specified probabilities. Also, the agent makes an observation o about the current state it is in (which is also given by a probability distribution). Lastly it receives a reward R, and the process repeats. The actions are chosen such that at each time step, the expected future discounted reward is maximized. To do so, the POMDP are solved approximately (POMDPs are usually intractable), and one key algorithm is point-based value iteration (PBVI) [5]. Prior to PBVI, POMDP solvers can only handle tens of states, limiting the applicability of POMDPs to problems. PBVI restricts value function computations to a finite subset of the belief space, avoiding the exponential growth of the value function. This key breakthrough enabled modern POMDP solvers to be able to handle thousands of states.

References

1. Russell and Norvig (2009). Artificial Intelligence: A Modern Approach. Prentice Hall.

2. Fikes and Nilsson (1971). STRIPS: a new approach to the application of theorem proving to problem solving. In Proceedings of the 2nd international joint conference on Artificial intelligence (IJCAI'71). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 608-620.
3. Blum and Furst (1997). Fast planning through planning graph analysis. Artificial Intelligence, 90(1-2):281-300.
4. Rintanen and Hoffmann (2001). An overview of recent algorithms for AI planning. KI, 15(2):5-11.
5. Pineau, Gordon, & Thrun, (2003). Point-based value iteration: An anytime algorithm for POMDPs. In International joint conference on artificial intelligence (pp. 1025-1032).