JOSE L. MATUS
November 7, 2017

Artificial Intelligence Planning Historical Developments

Research review by Jose L Matus

Automated planning and scheduling is one of the major fields of AI. Planning focuses on strategies or action sequences executed by Intelligent agents. To accomplish these tasks, these systems need to have input data containing descriptions of initial states of the world, desired goals and actions. And the role of planning systems is to find sequences of actions which lead from initial state to given goal.

In this review I will describe major developments in the field of AI planning search highlighting the relationship between them and their influence in the artificial intelligence field. I will describe the general framework **STRIPS**, under which all planning problems can be formulated. I will make a brief reference to **Planning Domain Definition Language**. Finally I will discuss **Planning Graphs** which are used to construct a Planning Graph object which can be used to obtain a solution.

Stanford Research Institute Problem Solver (STRIPS)

Designed by Richard Files and Nils Nilsson, it is an automated planner which goal was to find a series of operators in a space of models to alter an initial state into a model in which a given goal can be proven to be true.

Originally STRIPS was a name for the planning component in software used in **"Shakey, the robot"** developed at the Stanford Research Institute (SRI), which was the first machine to be able to reason about its own actions.

STRIPS assume that there exists a set of applicable operators which transform the world model into some other world model. The task of the solver is to find a sequence of operators which transform the given initial problem into one that satisfies the goal conditions.

The problem space for STRIPS is defined by the initial world model, the set of available operators and their effects on world models, and the goal statement. Operators are the basic elements from which a solution is built. Each operator corresponds to an action routine whose execution causes the agent to take certain actions. The available operators are grouped into families called schemata. Each operator is defined by a description consisting of two main parts: the effects the operator has and conditions under which the operator is applicable.

A problem is said to be solved when STRIPS produces a world model that satisfies the goal statement.

Planning Domain Definition Language

The PDDL was the first modeling language to be use widely for solving planning problems and it has remained the standard for the International Planning Competition since 1998. The PDDL was primarily inspired by STRIPS and ADL (The Action Description Language), with is a simpler representation of STRIPS that allows to encode more realistic problems by relaxing some of the STRIPS restrictions.

The usage of a common language for representing and solving planning problems encourages greater reuse of research, allows to analyze different approaches in an easier way and thus aids faster progress in the artificial intelligence field.

Planning Graphs

Developed in 1997 by Avrium Blum and Merrick Furst at Carnegie Mellon as a new approach to planing in STRIPS-like domains. It involved constructing and analyzing a brand new object called a Planning Graph. They developed a routine called GraphPlan which obtains the solution to the planning problem using a Planning Graph construct.

The idea is to first create a Planning Graph object which encodes useful constraints explicitly, thereby reducing the search overhead in the future. Planning Graphs can be constructed in polynomial time and have polynomial size. On the other hand, the state space search is exponential and is much more work to build. Planning graphs are not only based on domain information, but also the goals and initial conditions of the problem and an explicit notion of time.

Planning Graphs offer a means of organizing and maintaining search information that is reminiscent of the efficient solutions to Dynamic Programming problems. The GraphPlan algorithm uses a planning graph to guide its search for a plan. The search that it performs combines aspects of both total-order and partial-order planners. The algorithm guarantees that the shortest plan will be found.

Edges in a planning graph represent relations between actions and propositions. If a valid plan does exist in the STRIPS formulation, then that plan must exist as a subgraph of the Planning Graph. Another essential feature of planning graphs involve specifying mutually exclusive (mutex) relationships. Two actions are mutex if no valid plan could possibly contain both, and two states are mutex if no valid plan could make both simultaneously true. Identifying mutual exclusion relationships can be of enormous help in reducing the search for a subgraph of a Planning Graph that might correspond to a valid plan. Graphplan notices and records mutual exclusion relationships by propagating them through the Planning Graph.

The GraphPlan algorithm operates on the planning graph as follows: Start with a planning graph that only encodes the initial conditions. In Stage i, GraphPlan take the the planning graph from state i-1 and extends it one time step and then searches the extended planning graph for a valid plan of length i. If it finds a solution, then it halts, otherwise it continues to the next stage. Any plan that the algorithm finds is a legal plan and it will always find a plan if one exists. The algorithm also has a termination guarantee that is not provided by most planners.

Conclusion

- The STRIPS formulation gave researchers a general framework under which more advanced languages could be built.
- PPDL allows us to have one common language.
- The Planning Graph construct was a novel planning algorithm that uses ideas from standard total-order and partial-order planners, but differs most significantly by taking the position that representing the planning problem in a graph structure can significantly improve efficiency, although, it's main limitation is that it applies only to STRIPS-like domains.

References

1. Richard E. Fikes, Nils J. Nilsson (Winter 1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving". http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf

2. Stuart J. Russell, Peter Norvig (2010), Artificial Intelligence: A Modern Approach (3rd Edition).

3. Avrim L. Blum, Merrick L. Furst (1997) Fast Planning Through Planning Graph Analysis https://www.cs.cmu.edu/~avrim/Papers/graphplan.pdf