

Historical developments in planning

1- STRIPS

Stanford Research Institute Problem Solver (STRIPS) is a problem solver that operates on "world models". We assume that there exists a set of operations that can be applied to this world model to transform it into one that has the stated goal.

STRIPS was used in robotics for navigation and object manipulation. In this framework the state of the world needs to be encoded as facts and relations in first-order predicate calculus. Operators are effectively actions that the robot can take.

The contribution of this work relative to the state-of-art at the time was the separation of problem solving and search through world models. Instead of a naive exhaustive search a heuristic is employed that identifies "relevant" actions and only pursues those sub-problems. The theorem prover is determine if the current "world model" can lead to the desired goal. If so the problem is solved. Otherwise we look for an operator that maximizes relevance. This means-ends analysis guides the search process and allows for a smaller search space. However planning using STRIPS is at least PSPACE-hard and therefore not efficient.

2- Graphplan

Graphplan builds on STRIPS and introduces the paradigm of Planning Graph Analysis. Graphlab starts with building a compact graph representation in polynomial time and space. The algorithm then searches this graph in a manner that is similar to dynamic programming and therefore much more efficient than STRIPS. Interestingly Graphplan is not sensitive to the order of actions in the plan unlike traditional approaches.

Graphplan is an important contribution predominantly due to the novel representation of planning with graphs. As the authors point out many heuristics can be applied on top of Graphlab to accomplish more efficient searching. It also provides a means for converting planning into a max flow problem that has many fast algorithms. A downside of Graphlab is that it's limited to STRIPS-like domains meaning that objects cannot be added and consequences of actions cannot change over time.

3- HSP

Heuristic Search Planner (HSP) works on STRIPS-encoded problems. Employing heuristics allows us to find efficient solutions to an otherwise inefficient search. Finding the optimal cost even in a relaxed problem is still NP-hard. However HSP uses an iterative algorithm to find an approximation. Once this approximation is found we use A* or greedy (i.e. hill climbing) search.

In an empirical study the authors show that HSP is able to solve more problems compared to the alternatives but it's generally slower and does not provide any optimality guaranteed. However the solutions found were very closes to the shortest plans.

References

- STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, Fikes et al,
<http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf>
- Fast Planning Through Planning Graph Analysis, Blum et al,
<https://www.cs.cmu.edu/~avrim/Papers/graphplan.pdf>
- HSP: Heuristic Search Planner, Bonet et al,
<https://bonetblai.github.io/reports/aips98-competition.pdf>