

Historical development of AI Planning and Search

Jinxiang Ruan

Planning and is a sub-field of Artificial Intelligence (AI). In this article, three major developments in the field of AI planning research would be covered. For each development, I will provide a short summary of the original paper.

STRIPS

STRIPS (Stanford Research Institute Problem Solver) is an automated planner developed by Richard Fikes and Nils Nilsson in 1971 at SRI International [1]. Formally, 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.

Mathematically, a STRIPS instance is a quadruple (P, O, I, G) , in which each component has the following meaning:

1. P is a set of conditions (i.e., propositional variables);
2. O is a set of operators (i.e., actions); each operator is itself a quadruple
 - 1). Positive conditions (conditions must be true for the action to be executable)
 - 2). Negative conditions (conditions must be false for the action to be executable)
 - 3). Effect Add (effects that made true by the action)
 - 4). Effect Remove (effects that made false by the action)
3. I is the initial state, given as the set of conditions that are initially true (all others are assumed false);
4. G is the specification of the goal state; this is given as a pair (positives, negatives), which specify which conditions are true and false, respectively.

A plan for such a planning instance is a sequence of operators that can be executed from the initial state and that leads to a goal state.

Graphplan

Graphplan is an algorithm for automated planning developed by Avrim Blum and Merrick Furst in 1995 [2]. The idea is that rather than greedily searching, we first create a Planning Graph object. The Planning Graph is useful because it inherently encodes useful constraints explicitly, thereby reducing the search overhead in the future.

In Graphplan's planning graph:

- the nodes are actions and atomic facts, arranged into alternate levels,
- and the edges are of two kinds:
 1. From an atomic fact to the actions for which it is a condition,
 2. From an action to the atomic facts it makes true or false.

The first level contains true atomic facts identifying the initial state.

Lists of incompatible facts that cannot be true at the same time and incompatible actions that cannot be executed together are also maintained.

The algorithm then iteratively extends the planning graph, proving that there are no solutions of length $l-1$ before looking for plans of length l by backward chaining: supposing the goals are true, Graphplan looks for the actions and previous states from which the goals can be reached, pruning as many of them as possible thanks to incompatibility information.

Heuristic Search Planner

Following the significant success of Graphplan, interest in the planning problem was revitalised and other new ideas were explored. HSP [3] contributed by McDermott and by Geffner and Bonet demonstrated a method by which a surprisingly informative heuristic function could be constructed automatically simply by analysing the domain.

The underlying principle is: The heuristic value of a particular choice of action is based on an estimate of how much work remains to be accomplished following the addition of that action to the plan. To estimate the outstanding work is a very simple: the number of actions required to achieve all the outstanding goals.

References

[1] STRIPS Paper

<http://ai.stanford.edu/~nilsson/OnlinePubs-Nils/PublishedPapers/strips.pdf>

[2] GraphPlan Paper

<https://www.cs.cmu.edu/~avrim/Papers/graphplan.pdf>

[3] HSP Paper

<https://bonetblai.github.io/reports/aips98-competition.pdf>