Research Review

The planning problem in Artificial Intelligence is about the decision making performed by intelligent creatures like robots, humans, or computer programs when trying to achieve some goal. Classical planning refers generically to planning for restricted state-transition systems which can be divided into two particular types of search spaces: state spaces and plan spaces[1]. There are some defination languages such as STRIPS(which is the first one to describe planning problem) and PDDL. In addition, Planning-Graph, Propositional Satisfiability and Constraint Satisfaction are the techniques used for Neoclassical Planning. Heuristic techniques for node selection is also important in searching such as FastForward and HSP[2].

STRIPS is cited as providing a seminal framework for classical planning problem in which the world is regarded in a static state and transformable to another one. In retrospect, STRIPS was extremely limited in both the scope of planning issues it addressed and the complexity of problems it could solve. Even with those limitations, the STRIPS representation and reasoning framework was used as the basis for most automatic planning research for many years. The STRIPS framework had sufficient intuitive appeal to most researchers for them to believe that it was a viable foundation on which to develop techniques that would be effective in more realistic models[3]. STRIPS has more influence on definition rather than on his algorithm.

Before Graphplan came out, most planning researchers were working on PSP-like planners. But GRAPHPLAN- and SAT-based planning systems can quickly solve problems that are orders of magnitude harder than those tackled by the best previous planners. GRAPHPLAN algorithm is one of the most exciting developments in AI planning for two reasons: 1. GRAPHPLAN is a simple, elegant algorithm that yields an extremely speedy planner in many cases, orders of magnitude faster than previous systems. 2. the representations used by GRAPHPLAN form the basis of the most successful encodings of planning problems into propositional SAT[4]. Many subsequent planning systems have used ideas from it. Many of the successors are much faster than the original Graphplan such as IPP, STAN, GraphHTN, SGP, Blackbox, Medic, TGP, LPG[5].

Partial-order planning is an approach to automated planning that leaves decisions about the ordering of actions as open as possible. It contrasts with total-order planning, which produces an exact ordering of actions[6]. Partial-order reduction methods try to decrease the number of search steps by recognizing different forms of independence of actions/transitions. Partial-order planning is faster and thus more efficient than total-order planning but it requires a lot more computational power for each node. Partial-order planning steps down from the stage after It dominating about twenty years of research in AI planning. Despite its scale-up problems, partial order planning remains attractive over state space and CSP-based planning for offering a higher degree of execution flexibility[7].

refrences：

[1] Automated Planning theory and planning. Malik Ghallab 2004

[2] Artificial intelligence, A modern approach. Stuart J. Russell 2013

[3] STRIPS, a retrospective. Richard E. Fikes and Nils J. Nilsson 1993

[4] Recent Advances in AI Planning. Daniel S. Weld 1999
[5] AI Planning course at the University of Maryland. Dana Nau 2012
[6] https://en.wikipedia.org/wiki/Partial-order_planning
[7] Reviving Partial Order Planning. XuanLong Nguyen and Subbarao Kambhampati 2001