Planning Research Review

The first planning system, **Stanford Research Institute Problem Solver** (**STRIPS)**, modelled based on the **General Problem Solver (GPS)**, a **state-space search** system used means-ends analysis. However, its influence was mainly on the representation language of planning problems by utilising **Action Description Language (ADL).** Later, the **Problem Domain Description Language (PDDL)** which is computer-parsable, standardized syntax for representing planning problems.

Early versions of planners (1970s) utilised the **linear planning** approach or **totally ordered action sequences** where a subplan for each subgoal is combined together in some order. Linear planning was found to be incomplete and unable to solve simple problems such as the Sussman anomaly.

Interleaving approach where actions from different subgoals were interleaved within a single sequence was found to be a complete. The interleaving approach led to the **goal-regression planning** where actions in a totally ordered plan are reordered to avoid conflicts between subgoals.

Partial-order planning where ideas include conflict detection and interference protection from different goals become dominant for 20 years. The first formal exposition (**TWEAK**) was simple and allowed proofs of completeness and tractability of planning problems, later giving rise to widely distributed implementations (**SNLP** and **UCPOP**).

A **state-space planning** approach became practical for solving large planning problems following (**UNPOP**) which introduced the ignore-delete-list heuristic. Its derivatives **Heuristic Search Planner** (**HSP**), **FF**, **FastDownward** and **LAMA** have been successful and won many planning competitions.

Graph-planning system such as **GRAPHPLAN**, **IPP**, **STAN**, and **SGP** were orders of magnitude faster than partial-orders planners. However, accurate heuristic from planning graphs led to partial-orders planners such as **REPOP** planner which is competitive with the best state-space planners while scaling better than **GRAPHPLAN** in parallelizable domains.

The analysis of several classes of planning problems demonstrated that constraint-based approaches such as **GRAPHPLAN** and **SATPLAN** are suitable for NP-hard domains while search-based approaches perform better in domains where feasible solutions are found without backtracking. Generating propositionalized actions dynamically instead of instantiating them before the start of the search may solve the issues of **GRAPHPLAN** and **SATPLAN** when operating in domains with many objects.

Reference:

Artificial Intelligence: A Modern Approach (3rd edition) by SJ Russell, P Norvig