Research Review

One of the earliest attempts at AI Planning was the General Problem Solver (GPS)[1]. The General Problem Solver used a “find-and-reduce-differences” heuristic approach to determine search strategies for various problems. GPS used first-order logic to describe planning problems and environments, and search goals. The core technique in GPS is to analyzes the difference between the current state of the search in the search space and the distance from the current state to the goal state, in order to determine which node to expand in order to progress toward the goal state. It was one of the first heurtistic methodologies to search through a state space to compute a sequence of steps from an initial state to a goal state. STRIPS/GPS modeled a planning problem using a four-tuple:- Atoms, Operations, Initial States, and Goal States. The definition of the planning problem drove the heuristic to move from an initial state to a final state. GPS was also used first to model the planning algorithms used in the implementation of STRIPS, which was used in the algorithm of planning Shakey the Robot[2,3].

Shakey the Robot was one of the first attempts to create a robot enabled by artificial planning algorithms to navigate in a simulated environment. The planning framework for the Robot was based on STRIPS, a problem solver that searched through a state space to one in which a given goal is achieved. The Robot’s environment was modeled using First-order Logic and searching through the state space using theorem-proving systems. The Robot’s goals are expressed as statements in first-order logic, very much like airplane planning problem in this project. The system operated by modeling the environment using first-order logic statements and then proving that a set of goals correspond to theorems that can be proven from the formal statements describing the environment. For example, a statement *goto(x,y)* that moves the robot from state x to state y would be encoded by the logical statement “AtRobot(X,State) => AtRobot(Y, goto(X,Y,State)” for all X,Y,State. After defining all such actions the Robot was capable of performing using such a logical framework (that is derived from GPS), moving the robot from one part of the environment to the another could be specified using the current state of the robot, and the final goal state of the robot, in terms of x,y co-ordinates, and then using the theorem-prover module to create a series of actions that would lead from the the initial state of the robot to the final state of the robot.

One of the drawbacks of the STRIPS problem solver was that it required explicit descriptions of the state of objects, such as barriers and obstacles that could not be affected by the operators used to describe the robot’s movement. Thus, the planning algorithm had to consider additional logical states that had nothing to do with the actual steps required to satisfy a goal state. This made the modeling of the robot’s environment very cumbersome and also required the theorem prover to consider a large number of logical statements that essentially amounted to “obstacle X stayed in the same spot” while considering each step of the plan to the end state. Nonetheless, many of these techniques used in GPS and STRIPS seem to have motivated a lot of today’s planning techniques. Clearly, these techniques assumed a non-stochastic and deterministic world, unlike today’s AI planning techniques, but they ahead of their time even considering they operated in a simpler model. However, non-optimal sequential planning was still a challenge that relied on linear planning and Partial-order planning, which met with limited success in solving real-world planning problems until the resurgence of using heuristic methods to solve planning problems, from which finding optimal planning steps were proven to be computationally intractable.

One of the first heuristic planning schemes that proved the power and effeciency of non-optimal heuristic planner was Bonet and Geffner’s Heuristic State Planner (HSP). HSP modeled the world using a six-tuple to describe a planning problem as a sequence of states, including an initial state and a goal state, a set of actions, a set of transition functions associated with actions performed in any state, and finally, a cost associated with performing the action. Thus, the planning problem was effectively reduced to an constraint satisfaction problem, where reaching the goal state was associated with performing the set of actions with the smallest-possible cost to reach a goal state. The notion of an “admissible heuristic” is outlined in this paper, and is exactly the same one taught in the planning chapters of the AIMA book. Bonet and Gaffner used their admissible heuristic technique and problem description framework to compute sub-optimal solutions for a range of AI problems, and demonstrated that the HSP was a competitive and scalable technique that could be used to determine sub-optimal solutions for complex planning problems and games and puzzles.

[1] “Human problem solving”, Newell and Simon, 1970, Prentice-Hall.
[2]Wilber, B. M. A Shakey Primer, Technical Report . Stanford Research Institute, 333 Ravenswood Ave, Menlo Park, CA 94025, November 1972. [PDF, Details]
[3]Fikes, Richard E. and Nilsson, Nils J. **STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving**, Technical Note 43R. AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, May 1971
[4] Boney, B. and Geffner H., Planning as a Heuristic Search, Journal of Artificial Intelligence, Springer-Verlag, Feb 2000