History of Heuristic Search Planning

A planning problem is framed to answer whether or not we can reach a goal state from an initial state. The solution involves a sequence of actions that can be taken to transition from states of the world, gradually, to satisfy the goals. Therefore a planning problem is a search problem. However the world is often defined as an atomic state(i.e. atomic facts/objects) and thus would require humans to specify a domain-specific heuristic to the search problem. But with planning we can use a factored representation for states along with action schemas that represent procedures for changing atomic states. A factored representation makes it possible to define domain-independent heuristics.

The first approach to derive a domain-independent heuristic from a planning problem was McDermott's UnPOP program(1996) which used means-end analysis to build a graph that helps find feasible and relevant actions to add to the solution. This was accomplished by using a regression match graph that created "and" branches(conjunctions of goals/preconditions) and "or" branches(different ways of achieving subgoal) along with "effort" numbers that estimated how many actions it would take to reach the end goal. In essence, McDermott used means-end analysis to create a heuristic estimation to guide a state space search. Further, McDermott was the first to suggest the ignore-delete-list-heuristic which assumes all goals and preconditions contain only positive literals. The heuristic was a consequence of the UNPOP program which tended to only consider the current situation in planning and not future consequences of extending the plan.

In addition, Bonet introduced a heuristic search planner(HSP) that made state-space search practical for large planning problems. First, in 1997, Bonet modified the original STRIP's representation of action schemas by replacing the add and delete list with a new effects list. Bonet went from a relational representation to a functional representation because the STRIP's representation generates more operators for each world and misleads the heuristic by introducing "spurious preconditions." Bonet later, in 2001, improved HSP and proposed HSP2 which performed forward and backward state-space search while also making his heuristic computation more efficient using backward search.

However, Bonet's HSP2 would later get outperformed by Hoffman's Fast-Forward Search in 2001 AIPS Planning Competition. As Bonet states it, FF solves all problems, really fast by differing from HSP in the following three ways: (1) Hoffman's heuristic provides a better estimate without deletes, (2) a hill-climbing search algorithm with a lookahead mechanism and (3) a fast pruning criterion. FF's heuristic algorithm is similar to HSP's but it is derived by applying GRAPHPLAN to a relaxed planning task which allows FF not to rely on the subgoal independence assumption inherit in HSP. Subgoal independence assumes that the cost of solving a conjunction of subgoals can be estimated by sum of their independent cost. Consequently, this approach can be inadmissible since it ignores positive interactions between subgoals and thus allows redundant actions. Further, Hoffman leverages a planning graph to take advantage of these positive interactions.

The heuristic search planner extended the efficiency of planning systems and made them feasible for large planning systems. It also allowed the use of domain-independent heuristics to be extracted from the problem and used to guide the search for goal states.

Sources

D. McDermott, A heuristic estimator for means-ends analysis in planning, in: Proc. 3rd International Conference on AI Planning Systems, AAAI Press, Menlo Park, CA, 1996, pp. 142–149

B. Bonet, G. Loerincs, H. Geffner, A robust fast action selection mechanism for planning, in: Proc. AAAI-97, Providence, RI, AAAI Press, Menlo Park, CA, 1997, pp. 714–719.

Blai Bonet, H. Geffner, Planning as heuristic search, Artificial Intelligence 129, 2001

J. Hoffman, B. Nebel, The FF Planning System: Fast Plan Generation Through Heuristic Search, in: Journal of Artificial Intelligence Research 14 (2001) 253-302