Given that forward state-space search

is one of the oldest techniques available for planning, and many other search spaces and approaches have been developed, this state-of-the-art performance is somewhat surprising. One of the key reasons for the success is the development of powerful domain-independent heuristics that work well on many AI planning domains.

control knowledge that can improve or speed-up forward state-space search in a non-optimal, or satisﬁcing, planning setting

capture information about the delete effects ignored in a relaxed plan, which can be used to learn knowledge that partially compensates for the underestimation

two general forms of control knowledge

4.1 Heuristic Functions

The ﬁrst and most traditional forms of control knowledge we consider are heuristic functions. A heuristic function is simply a function of a state s, action set A, and goal g that estimates the cost of achieving the goal from s using actions in A. If a heuristic is accurate enough, then greedy application of the heuristic will ﬁnd the goal without search. However, when a heuristic is less accurate, it must be used in the context of a search procedure such as best-ﬁrst search, where the accuracy of the heuristic impacts the search efﬁciency. In our experiments, we will use best-ﬁrst search, which has often been demonstrated to be an effective, though sub-optimal, search strategy in forward state-space planning. Note that by best-ﬁrst search, here we mean a search that is guided by only the heuristic value, rather than the path-cost plus heuristic value. This search is also called greedy best-ﬁrst search. In this paper, when we use the term best-ﬁrst search, it means greedy best

Recent progress in the development of domain-independent heuristic functions for planning has led to a new generation of state-of-the-art planners based on forward state-space heuristic search However, in many domains these heuristics can still have low accuracy, for example, signiﬁcantly underestimating the distance to goal, resulting in poor guidance during search

4.2 Reactive Policies in Forward Search

The second general form of control knowledge that we consider in this study is of reactive policies. A reactive policy is a computationally efﬁcient function , possibly stochastic, that maps a planning problem to an action in A. Given an initial problem ), we can use a reactive policy to generate a trajectory of pairs of problems and actions Ideally, given an optimal or near-optimal policy for a planning domain, the trajectories represent high-quality solution plans. In this sense, reactive policies can be viewed as efﬁcient domain-speciﬁc planners that avoid unconstrained search.