

AI Planning: A Prospectus on theory and applications

Subbarao Kambhampati, Arizona State University (rao@asu.edu)

In this brief article, I will attempt to summarize the status of research and applications of domain-independent planning in AI.¹ Developing automated methods for generating and reasoning about plans and schedules, whether in aid of autonomous or human agents, has been part and parcel of AI research from the very beginning. The need for planning arises naturally when an agent is interested in controlling the evolution of its environment. Algorithmically, a planning problem has as input a set of possible courses of actions, a predictive model for the underlying dynamics, and a performance measure for evaluating the courses of action. The output or solution is one or more courses of action that satisfy the specified requirements for performance. A planning problem thus involves deciding “what” actions to do, and “when” to do them. The “when” part of the problem has traditionally been called the “scheduling” problem. This separation is motivated as much by computational considerations as it is by the fact that algorithms for automating scheduling have been around longer (e.g. in the Operations Research community) than those for automating the complete planning problem.

The simplest case of the planning problem, where the environment is static and deterministic, and the planner has complete information about the current state of the world, has come to be known as the classical planning problem. Classical planners face two important issues: modeling actions and change, and organizing the search for plans (action sequences) capable of achieving the goals. Although the representation problem continues to receive significant attention from the logic programming and non-monotonic reasoning communities, most implemented planners settle on some variant of the STRIPS action model, where the state of the world is represented in terms of a set of state variables and their values, and actions are represented as deterministic state-transforming functions. The organization of search in classical planning turned out to be more controversial. The early planners modeled the search in terms of exploration of the space of world states, with the transitions between states being effected by the actions. Soon, the need, as well as the utility, of manipulating partial plans during search became clear, and this led to the design of algorithms that search in the space of partial plans. The search proceeds by refining the partial plans through the addition of actions and orderings (called partial order planning), or by replacing abstract actions by plan fragments that are capable of carrying out those actions (called hierarchical planning). It is now understood that even the state space approaches can be seen as another way of refining partial plans (by growing the prefix or suffix of the plan), and that the different ways of refining a partial plan can be fruitfully interleaved.

In the case of scheduling, the problem has received significant attention from the operations research community. The classical scheduling methods considered situations where the resource constraints are static and pre-specified in numeric form. In contrast, the AI methods to scheduling allow declarative specification of both symbolic and numeric resource

¹A more detailed overview can be found in the article “Planning and Scheduling”, co-authored with Tom Dean, and to appear in the CRC press handbook of Computer Science and Engineering.

constraints, and handle changes in the constraints during execution. Constraint satisfaction forms the canonical backbone of most AI scheduling methods, and several sophisticated heuristic search strategies have been developed for it.

While classical planning model has driven the majority of research in planning, more recently considerable attention has also been paid to planning in environments that are stochastic, dynamic and partially observable (and thus do not satisfy classical planning assumptions). To handle partially observable environments, information gathering is made part of the planning activity, and the classical planning techniques are extended to allow interleaving of planning and scheduling. Similarly, stochastic environments are modeled through Markov Decision Processes (MDPs), and planning in such environments involves of constructing policies for the corresponding MDPs.

The development of formal models has been accompanied with research into improving the efficiency of plan generation in those models. Since most interesting variants of domain-independent planning problem are known to be computationally hard (P-SPACE Complete or worse), efficiency can be achieved only through exploitation of problem distribution, and domain structure. This is currently done through the development of decomposition techniques to split the domain and the problem into tractable sub-components, inductive and speedup learning techniques to learn search control information and abstractions relevant to the expected problem distribution, and languages to express domain specific search control information for planners. Another research thrust involves extending and complementing the refinement planning models that are based exclusively on subgoaling. Two promising directions include constraint logic programming type models that allow both subgoaling and constraint satisfaction approaches to go hand-in-hand in controlling the planner (thereby handling action selection and scheduling aspects of the planning problem through computational models that are most appropriate for them), and some variants of network flow approaches that generate plans based on a form of disjunctive projection of future states of the agent.

In terms of applications, ever since the STRIPS planner helped SHAKEY the robot navigate SRI hallways in the early seventies, both classical and stochastic planning techniques have been routinely used as part of the design of autonomous agents. More recently, AI planning techniques are also being applied in “software domains” such as database query planning, or internet browsing. Perhaps surprisingly, AI planning techniques have had limited success in supporting industrial applications. Although industrial settings present a fertile arena for planning problems -- be they project planning, process planning, assembly planning or maintenance planning -- the extent of automation at present is limited to supporting scheduling, with the harder action selection problem delegated to the humans.

While this state of affairs may partly be due to the restrictive types of environments that were considered in the planning community until recently, there are also other technical and sociological reasons. Historically, planning problems have often been considered too important to be delegated to programs, and thus the model of humans doing action selection and programs doing action scheduling has become natural. To introduce automated planning technology into such settings, one not only needs to show that human action selection can be prone to errors, but also be prepared to cast the planner in the role of a *decision support* system.

This involves supporting full-fledged interactive planning, where the planner provides support to the humans involved in the decision making, and in turn gracefully accepts suggestions on how to redirect its own search or to revise its results. Further, where possible the planning techniques need to be integrated into the any existing semi-automated tools that are routinely used by the humans solving those problems.

Incremental and interactive planning is also important from a different perspective. In most applications, planning might involve reasoning about a variety of constraints beyond temporal constraints. For example, path planning, assembly planning and manufacturing planning problems often need spatial and geometric reasoning capabilities. It is typically infeasible to ignore and/or abstract away these specialized constraints, and is undesirable to try to encode them all within a homogenous representation. This thus necessitates interaction between the planner and a host of specialized reasoners. Understanding the modalities of interaction between a planner and the external reasoners -- both human and machine -- is thus critically important.

Supporting incremental and interactive planning involves more than a simple graphical interface design however. The planner needs to understand the role it plays in the overall planning process, needs to keep track of the constraints on the plan imposed by external reasoners, and should be able to capture, understand and communicate the various types of rationales associated with its plan. Its search process should be transparent and controllable by the humans. All of these bring up research issues that need attention. Some of these issues are beginning to be addressed by the recent research on mixed-initiative and hybrid planning models, but a lot remains to be done.

In summary, AI planning has acquired a coherent body of knowledge and the current research directions seem both promising and exciting. While the application potential of existing planning techniques has only been partially exploited until now, I believe that the development of interactive planning models will significantly improve the situation.