

Chapter 1

Introduction

1.1 What is AI planning?

- modeling decision making needed by intelligent creatures acting in a complicated environment
- development of efficient algorithms for such decision making
- emphasis on general-purpose problem representation and general-purpose solution techniques; alternative would be to derive tailored algorithms for every problem separately

Impediments for the success of AI in producing genuinely intelligent beings are related to perceiving and representing knowledge concerning the world. The real world is very complicated in its all physical and geometric as well as social aspects, and representing all the knowledge required by an intelligent being may be too inflexible and complicated by the logical and symbolical means almost exclusively used in artificial intelligence and in planning. This has been criticized by many researchers [Brooks, 1991] and is a topic of continuing scientific debate as the problem is not well understood.

AI planning, and knowledge representation techniques in AI in general, are best applicable to restricted domains in which it is easy to identify what the atomic facts are and to exactly describe how the world behaves. These properties are best fulfilled by systems that are completely man-made, or systems in which planning needs to consider only at a very abstract level what is happening in the world.

Examples of completely man-made systems to which planning techniques have successfully been applied are given in the next section. This includes applications of planning in autonomous spacecraft.

A simple real-world application in which abstracting away the details of the real world would be transportation planning: how to get from Freiburg to London by public transportation, trains, airplanes and buses. If a robot were capable of finding its way between the couple of hundred of meters between the various forms of transportation and recognize the trains and buses to board it could easily travel all over the world. Planning what transportation to use is an easy problem in this case.

1.2 Where is AI planning used?

Truly intelligent robots or other artificial intelligent beings do not exist yet, and planning, like most other work on artificial intelligence, is still very much still something that takes place in research labs only.

Perhaps the most visible application of AI planning has been experimentation with autonomous spacecraft by the U. S. space agency NASA [Mussettola *et al.*, 1998].

At the level of applied AI research, AI planning is being used by many research projects that have produced autonomous but not very intelligent robots doing simple routine tasks in restricted environments, like delivering mail in an office or distributing medicine in a hospital. The uses of planning algorithms in this kind of setting, however, employ only very little from the potential of AI planning.

1.3 Types of planning problems

The word *planning* is very general, and denotes very many different things. Even in the AI and robotics context there are many types of planning, related to each other, but having different flavors.

The first problem in controlling autonomous robots, just their basic movement from one location to another, and the movement of the limbs, possibly with the ability to grip objects and move them, so called manipulators, is a very challenging problem. These problems are called *path planning* and *motion planning*, and they are not discussed in this lecture, as they require specialized representations of the geometric properties of the world, and cannot usually be efficiently represented in the general state-based model we are interested in. There is also the very well established research area of *scheduling* which is concerned with ordering and choosing a schedule for executing a number of predefined actions.

The more abstract planning that is the topic of this lecture is sometimes called task planning to distinguish it from the more geometric and physical forms of planning used in controlling the movements of robots and similar systems.

Even within task planning, there are many different types of planning problems, depending on the assumptions concerning the properties of actions and of the world that are made. Some of these are the following.

1. Determinism versus nondeterminism.

In the simplest form of planning the state of the world at any moment is unambiguously determined by the initial state of the world and the sequence of actions that have been taken. Hence the world is completely deterministic.

The assumption of a deterministic world holds when planning is to be applied in a sufficiently restricted setting. However, when the world is modeled in more detail and more realistically, the assumption does not hold any more: the plans have to take into account events that take place independently of the actions and also the possibility that the effects of an action are not the same every time the action is taken, even when the world appears to be the same.

Nondeterminism comes from at least two different sources.

First, the model of the world is usually very incomplete, and things that are possible as far as our beliefs are concerned can be viewed as a form of nondeterminism: we do not know whether somebody is going to phone or visit us, and then the visit or phone call can be modeled as a nondeterministic event that may or may not take place.

Second, many actions themselves are by their nature nondeterministic, either intentionally or unintentionally. Throwing two dice has 12 possible outcomes that usually cannot be predicted (which is why throwing dice is interesting!) Throwing some object to a garbage bin from a distance may or may not succeed.

Notice that there is still the possibility that the physical universe is completely deterministic, but as long as we do not know the exact causes of events, we might just as well consider them nondeterministic.

2. Observability.

For deterministic planning problems with one initial state there is no need to consider observations, because the goals can always be reached by one sequence of actions and the plan does not need to decide in the middle of plan execution between different courses of action.

When the actions or the environment can be nondeterministic, or when there are several initial states, the notion of plans as a sequence of actions is not sufficient.

There are two possibilities. Either planning is interleaved with plan execution: only one action is chosen at a time, it is executed, and based on the observations that are made the next action is chosen, and so on. Or a complete plan is generated, covering all possible events that can happen. This plan in the most general form has the structure of a program with branches (*if-then-else*) and loops.

In both cases, what can be observed has a strong impact on how exactly the actual state of the world can be determined: the more facts can be observed, the more precisely the current state of the world can be determined, and the better the most appropriate action can be chosen. If there is a lot of uncertainty concerning the current state of the world it may be impossible to choose an appropriate action.

If the current state can always be determined uniquely, we have *full observability*. If the current state cannot be determined uniquely we have *partial observability*, and planning algorithms are forced to consider sets of possible current states.

3. Time.

Most work on planning uses discrete (integer) time and actions of unit duration. This means that all changes caused by an action taken at time point t are visible at time point $t + 1$. So changes in the world take only one unit of time, and what happens between two time points is not analyzed further.

More complicated models of time and change are possible, but in this lecture we consider only discrete time. Most types of problems can be analyzed in terms of discrete time by making the unit duration sufficiently small. Rational and real time cause unnecessary conceptual difficulties. Effects of actions that are not immediate can easily be reduced to the basic case by encoding the delayed effects in the state description.

4. Control information and plan structure.

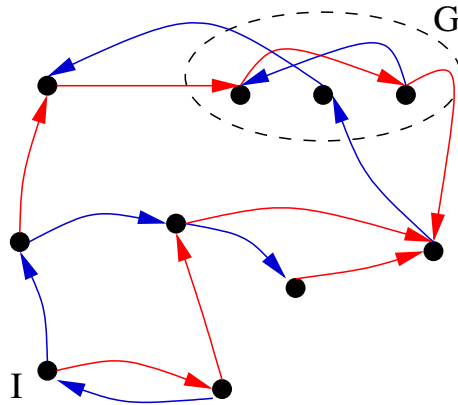


Figure 1.1: A deterministic planning problem

In the basic planning problem a plan is to be synthesized based on a generic description of how the actions affect the world.

There may be, however, further control information that may affect the planning process and the plans that are produced. In hierarchical planning, for example, information on the structure of the possible plans is given in the form of a hierarchical task network, and the plans that are produced must conform to this structure. This kind of structural information may substantially improve the efficiency of planning. Another way of restricting the structure of plans, for efficiency or other reasons, is the use of temporal logics [Bacchus and Kabanza, 2000].

5. Plan quality.

The purpose of a plan is often just to reach one of the predefined goal states, and plans are judged only with respect to the satisfaction of this property.

However, actions may have differing costs and durations, and plans could be assessed in terms of their time consumption or cost.

In nondeterministic planning, because different executions of a plan produce different sequences of actions, plans can be valued in terms of their expected costs, best-case costs, worst-case costs, and probability of eventually reaching the goals.

Plans with an infinite execution length can also be considered, and then plans may be valued according to their average cost per unit time, or according to their geometrically discounted costs.

1.4 Examples

Figure 1.1 illustrates what deterministic planning is. There is a set of states (the black dots), two actions (blue, red), an initial state I , and a set of goal states G . The task is to find a path from the initial state to one of the goal states. The planning problem is deterministic because in all states there is at most one red and at most one blue arrow going out of that state, which means that for all states the successor state is unambiguously determined by the action. In this example there are several possible plans for reaching G from I . Some of them are BRR (for blue, blue, red), RRRB,

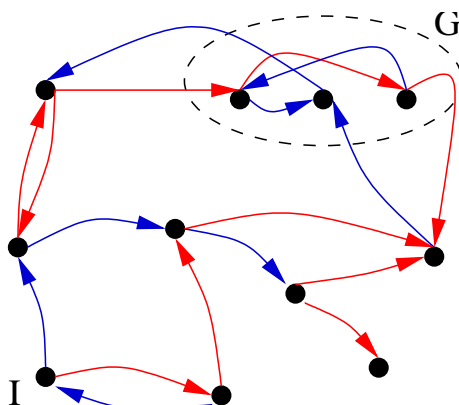


Figure 1.2: A nondeterministic planning problem

BRRRBRBRBRBRBRBRBRBR. The unique shortest plan is clearly BRR, as there are no plans of length 2 and not other plans of length 3.

Adding nondeterminism complicates planning, because even when there is a path from the initial state to a goal state, there is no guarantee that any single sequence of actions reaches the goal states, or even that the goals can be reached when choosing the next action can be postponed to the point when it is to be taken. A practical example is winning 1 euro in roulette given an initial capital of 5 euro. Whatever way the game is played, there is a relatively high probability of failure (not to mention that the expected outcome of the game is to lose money; don't do it!)

Figure 1.2 illustrates what a nondeterministic planning problem is. We have added a red arrow to the second state above the initial state, as well as a new state to the bottom right corner with a nondeterministic transition leading to it.

The new arrows make the red action nondeterministic, and have an impact on which action sequences are plans. First, nothing starting with BBBR or RRBR is a plan, because these action sequences might end in the bottom right state from which goal states cannot be reached. Second, if first the actions B and R are taken, the only way to proceed is to take the action R, and this either leads to a goal state, or takes us back toward the initial state. One possible plan would be to first take action B, and then repeat the sequence RR until a goal state has been reached.

Of course, even in this case there still are plans that do not involve nondeterministic steps, for example the plan RRRB.

A third possibility is planning with nondeterminism, but without the possibility of uniquely determining what the current state is. Figure 1.3 extends Figure 1.2 by restrictions on *observability*. Now during plan execution it is possible only to recognize the current state based on the color green or black. All black states are observationally indistinguishable from each other, and likewise all green states. We may be able to infer something about the current state based on the actions already taken and the observations made earlier, but these do not in general allow to infer the current state unambiguously.

The plan in the previous case consisting of the action B followed by iteration of RR until a goal state is reached cannot be executed any more, because the two states reached with the second R are indistinguishable.

However, for this problem there is a closely related plan that works also with partial observability. First take actions BR. Then repeat RR until the current state becomes black. So, it does not

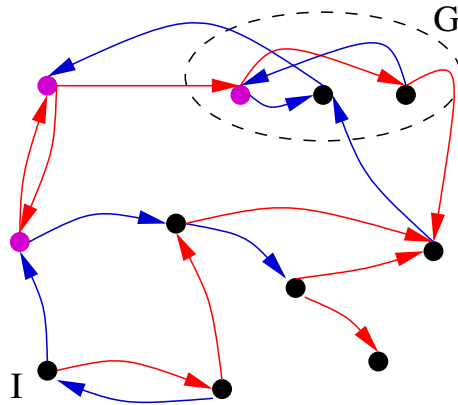


Figure 1.3: A nondeterministic planning problem with partial observability

simply suffice to reach a goal state, but one also has to be able to recognize that the goal state has been reached so that plan execution can terminate.

1.5 Related topics

Reasoning about action has emerged as a separate research topic with the goal of making inferences about actions and their effects [Ginsberg and Smith, 1988; Shoham, 1988; Sandewall, 1994a; 1994b; Stein and Morgenstern, 1994]. Important subtopics include the qualification and the ramification problems, which respectively involve deciding whether a certain action can be performed to have its anticipated effects and what are the indirect effects of an action. Both of these problems are of independent interest, both for their relations to the reasoning human beings do and for their importance in representing the world as required by any intelligent system for doing planning. In this lecture, however, we assume that a description of some actions is given, with all preconditions and direct and indirect effects fully spelled out, and concentrate on what kind of planning can be performed with these actions. The separation between planning and reasoning about actions is useful for both structuring systems that plan and act in a complicated world and for learning about these two topics.

Markov decision processes [Puterman, 1994] in operations research is essentially a formalization of planning. In contrast to AI planning, work in that area has used explicit enumerative representations of transition systems, like those used in Section 2.1, and as a consequence the algorithms have a different flavor than most planning algorithms do. However, most of the recent work on probabilistic planning, as discussed in Chapter 5, is based on Markov decision processes.

Discrete event systems (DES) in control engineering have been proposed as a model for synthesizing controllers for systems like automated factories [Ramadge and Wonham, 1987; Wonham, 1988], and this topic is closely related to planning. Again, there are differences in the problem formulation, with state spaces represented enumeratively or more succinctly, for example as Petri nets [Ichikawa and Hiraishi, 1988] or vector additions systems [Li and Wonham, 1993].

Synthesis of programs for reactive systems that work in nondeterministic and partially observable environments is similar to planning under same conditions. Program synthesis has been considered for example from specifications of their input-output behavior in different types of temporal logics [Vardi and Stockmeyer, 1985; Kupferman and Vardi, 1999].

1.6 Early research on AI planning

Research that has to current AI planning started in the 1960's in the form of programs that were meant simulate problem solving abilities of human beings.

One of the first programs of this kind was the General Problem Solver by Newell and Simon [Ernst *et al.*, 1969]. GPS performed state space search and used a heuristic that estimated differences between the current and goal states.

In the end of 1960's Green proposed the use of theorem-provers for constructing plans [Green, 1969]. However, because of the immaturity of theorem-proving techniques at that time, this approach was soon mostly abandoned in favor of specialized planning algorithms. There was theoretically oriented work on deductive planning that used different kinds of modal and dynamic logics [Rosenschein, 1981], but these works had little impact on the development of planning algorithms. Deductive and logic-based approaches to planning gained popularity again only in the end of 1990's as a consequence of the development of more sophisticated programs for the satisfiability problem of the classical propositional logic [Kautz and Selman, 1996].

The historically most influential planning system is probably the STRIPS planner from the beginning of the 1970's [Fikes and Nilsson, 1971]. The states in STRIPS are sets of formulae, and the operators change these state descriptions by adding and deleting formulae from the sets. Heuristics similar to the GPS system were used in guiding the search. The definition of operators, with a *precondition* as well as *add* and *delete* lists, corresponding to the literals that respectively become true and false, and the associated terminology, has been in common use until very recently. The add list is simply the set of state variables that the action makes true, and the delete list similarly consists of the state variables that become false.

Starting in the 1970's the dominating approach to domain-independent planning was the so-called partial-order, or causal link, or nonlinear planning, [Sacerdoti, 1975; McAllester and Rosenblitt, 1991], which remained popular until the mid-1990's and the introduction of the Graphplan planner [Blum and Furst, 1997] which started the shift away from partial-order planning to types of algorithms that had been earlier considered infeasible, even the then-notorious total-order planners. The basic idea of partial-order planning is that a plan is incrementally constructed starting from the initial state and the goals, by either adding an action to the plan so that one of the open goals or operator preconditions is fulfilled, or adding an ordering constraint on operators already in the plan in order to resolve potential conflicts between them. In contrast to the forward or backward search strategies in Chapter 3, partial-order planners tried to avoid unnecessarily imposing an ordering on operators too early. The main advantages of both partial-order planners and Graphplan are present in the SAT/CSP approach to planning which we will discuss in detail in Section 3.5.

In parallel to partial-order planning, the notion of hierarchical planning emerged [Sacerdoti, 1974], and it has been deployed in many real-world applications. The idea in hierarchical planning is that the problem description imposes a structure on solutions and restricts the number of choices the planning algorithm has to make. A hierarchical plan consists of a main task that can be decomposed to smaller tasks that are recursively solved. For each task there is a choice between solution methods. The less choice there is, the more efficiently the problem is solved. Furthermore, many hierarchical planners allow the embedding of problem-specific heuristics and problem-solvers to further speed up planning.

A collection of articles on AI planning starting from the late 1960's has been edited by Allen *et al.* [Allen *et al.*, 1990]. Many of the papers are mainly of historical interest, and some of them

outline ideas that are still very much in use today.

1.7 This book

My intention in writing these lecture notes was to cover planning problems of different generality and some of the most important approaches to solving each type of problem. It goes without saying that during the last several decades of planning research a lot of work has been done that are not covered by these notes.

Important differences to most textbooks and research papers on planning is that I use a unified and rather expressive syntax for representing operators, including nondeterministic and conditional effects. This has several implications on the material covered in this book. For example, many people may find it surprising that I do not use a concept viewed very central for deterministic planning by some, *the planning graphs* of Blum and Furst [1997]. This is a direct implication of the general syntax for operators I use, as discussed in more detail in Section 3.9. It seems that any graph useful graph-theoretic properties planning graphs have lose their meaning when a definition of operators more general than STRIPS operators is used.

One of the important messages of these notes is the importance of logic (propositional logic in our case) in representing many of the notions important to all forms of planning ranging from the simplest deterministic case to the most general types of planning with partial observability. As we will see, states, sets of states, belief states and transition relations associated with operators are in many cases represented most naturally as propositional formulae. This representation shows up once and again in connection of different types of planning algorithms, including backward search in classical/deterministic planning, planning as satisfiability, and in implementations of nondeterministic planning algorithms by means of binary decision diagrams.

In addition to generalizing many existing techniques to the more general definition of planning problems, many of the algorithms are either new or have been developed further from earlier algorithms. I cite the original sources in the literature sections in the end of every chapter. Some of my contributions can be singled out rather precisely. They include the following.

1. The definition of regression for conditional and nondeterministic operators in Sections 3.2.2 and 4.1.4.
2. The algorithm for computing invariants in Section 3.6. The computation of mutexes in Blum and Furst's [1997] planning graphs can be viewed as a simple special case of my algorithm, restricted to unconditional operators only.
3. The algorithm for planning with full observability in Section 4.3.2. This algorithm is based on a similar but more complicated algorithm by Cimatti et al. [2003].
4. The representation of planning without observability as quantified Boolean formulae in Section 4.4.2.
5. The framework for non-probabilistic planning with partial observability in Section 4.4.3.
6. The complexity results in Section 4.5.3, most importantly the 2-EXP-completeness result for conditional planning with partial observability.