Planning

Planning is a branch of artificial intelligence that deals with strategies or action sequences that are executed by robots, unmanned vehicles or intelligent agent. The solutions are usually complex, especially in multidimensional space. Planning is also relates to decision theory, which is used in other fields of studies such as statistics and psychology.

Planning can be either done online or offline, which entirely depends on the environment. If models are provided, planning can be done offline, allowing solutions to be found and evaluated before execution. However, in dynamically unknown environments, strategies need to be revised online. An example of this would be unmanned space programs.

Applications of planning are widely used in the world today. As mentioned before, unmanned spacecraft’s heavily rely on planning in order to achieve the mission they were assigned. Other uses of planners include Google Maps and factories. Google Maps uses planning to plan routes from reaching point A to B. In addition, it can dynamically change and reroute the user to a new path in the case of changing destinations, adding a new destination along the way, and even traffic congestion. Factories take advantage of planning by using them to plan specific instructions when to automate tasks when producing goods and optimizing for efficiency. With the aid of planners, the world today is much more efficient and robust thanks to the decision making by artificial intelligence.

There are many different kinds of planners, each differing by the approach of solving or using the constraints to reach a final goal or state. The 3 main types of planners that we will go over are: STRIPS, HTN and PDDL.

STRIPS

STRIPS, also known as Stanford Research Institute Problem Solver, is one of the most used automated planners.   
Most planners incorporate some style of STRIPS into their planning scheme. A STRIPS planner instance is composed of 3 major components:

* Initial state
* Specification of goal states-situation where the planner is trying to reach
* Set of actions which include a precondition and postcondition

In strips, the initial state, is the starting initial set of conditions that are all true. In order to reach the goal state, there are world states that are represented as a set of facts. World states can be thought as options for the planner to take to reach the goal. In STRIPS, the closed world assumption states that any fact not listed in a state is assumed to be false

Goals are also represented as a set of facts. The goal state is any state that contains all the goal facts for the condition to be met.

Actions in STRIPS are represented in 3 phases.

* PRE of preconditions of facts
* ADD of add effects facts
* DEL of delete effect facts

An action is only allowed in a new state when the preconditions are contained in the old state. Once the action has been executed, the planner enters a new state with a given set of new facts. From there , the planner either checks if it has reached its goal state, or repeats and continues on to reach the goal state.

In short, the objective of a STRIPS planner is to find a short action sequence reaching a goal state, or report that the goal is unachievable.

Hierarchical Network Planners

HTN are another type of planners used in automated planning. Similar to strips, HTN differ from STRIPS in which dependency among actions can be given in the form of networks

In HTN, tasks are broken into 3 categories:

* Primitive tasks, which roughly correspond to actions of STRIPS
* Compound tasks, which can be seen as composed of a set of simpler tasks
* goal tasks, which roughly correspond to the goals of STRIPS, but are more general

A primitive task by definition is an action that can be executed, a compound task is a complex task composed of a sequence of actions,  
and a goal task is the task of satisfying a condition.   
Primitive actions can be executed directly while compound and goal tasks require a sequence of primitive actions to be formed.   
Goal tasks however are specified in terms of conditions that to be made true,  
while compound tasks can only be specified in terms of other tasks through the task network.

Constraints among tasks are expressed in form of networks called task networks. A task network is a set of tasks and constraints among them,  
and can be used as a precondition for another compound task or goal to be attainable.   
With this setup, one can express a given task is doable only if a set of other actions within the network are done,  
and done with such a way that the constraints are satisfied.

An example would be that a condition is necessary for a specific primitive action to be executed.  
When this is used as the precondition for a compound task or goal, it means that the compound or goal task requires the primitive action to be executed,  
and that the condition must be true for its execution to successfully achieve the compound or goal task.

Planning Domain Definition Language

The PDDL was a attempt to be the standard for classical programming tasks. PDDL is composed of the following components:

* Objects- Things in the world that interest us
* Predicates- Properties of objects that we are interested in; can be true or false
* Initial State- The state of the world that we start in
* Goal specification- Things that we want to be true
* Actions/Operators- Ways of changing the state of the world

In PDDL, planning tasks are separated into 2 files

* A domain file for predicates and actions
* A problem file for objects, initial state and goal specification

In the Domain file, the title of the file specifies the domain name, followed by the Objects and the predicates. After this are the list of actions with parameters, with preconditions and effects, similar to STRIPS. The effects of actions could also be conditional

In the problem file, the title of the file specifies the problem name. Followed by the problem name is the domain name, which must match the corresponding domain with the corresponding domain file. The file then specifies the initial state, followed by the goal specification.

As described above, PDDL is very similar to STRIPS with the preconditions, effects, initial state and goal state. The only major difference is that the problem is separated into a different file. Nevertheless, PDDL is a excellent planner to use when deciding to choose a planner.

Narrative Generation

Narrative generation is interesting type of planning, in which the made plan simulates deviations as the plan is executed. The planner replans based on the new state, and continues on its way to its goal. Sometimes, the end goal is changed in order make the specific goal achievable.

For example, a plane takes off from a certain airport; all is going well after take off until a passenger gets extremely sick. The planner simulates the sick passenger deviation, as a regular trip would just end with nothing significantly happening. From there, the planner can decide to change the end destination, continue on to the designated end designation, or fly back to the starting airport. The narrative generator has many options as this point, and can simulate the different goal endings at well!