Machine, Data and Learning (CS7.301)

Spring 2022, IIIT Hyderabad
07 Feb, Monday (Lecture 8)

Taught by Prof. Vikram Pudi

Solving Problems by Searching

Problem-Solving Agents

A problem-solving agent, generally speaking, takes in a perception of the state of the world and returns an action to solve a certain problem.

Problem Types

A task may be deterministic or nondeterministic, and it might be fully observable, partially observable or non-observable.
A deterministic and fully observable task can be formulated as a single-state problem; a non-observable task is a sensorless or conformant problem; a nondeterministic and/or partially observable task is a contingency problem; and an unobservable problem is an exploration problem.

Problem Formulation

A problem is defined by:

- an initial state
- actions or a successor function (a set of action-state pairs)
- a goal test
- a path cost

A solution to a problem is a sequence of actions leading from the initial state to a goal state.

The state space must be abstracted from the state space of the real word, which is too complex for realisability.

Searching Algorithms

There are many ways to search through a state space.

One common type is tree search algorithms, which proceed by generating successors of previously explored states.

Search strategies are evaluated according to their completeness, complexity (time and space), and optimality.

Breadth-first search (expand shallowest unexpanded node) is one search algorithm which is complete and optimal. It has time and space complexity $O(b^{d+1})$.

Uniform-cost search expands the least-cost unexpanded nodes. It is complete and optimal, and has time ans space complexity $O(b^{\text{ceil}\left(\frac{c^*}{\varepsilon}\right)})$, where c^* is the cost of the optimal solution.

Depth-first search proceeds by expanding the deepest unexpanded node. It is *not* complete and *not* optimal. However, it has time complexity $O(b^m)$ and space complexity $O(bm)$. It can be modified to a depth-limited search.