# Artificial Intelligence & Machine Learning

Bodhayan Roy

Department of Mathematics,
Indian Institute of Technology Kharagpur

Lecture 4

# Components of a problem

A problem can be defined formally by five components:

- The initial state that the agent starts in.
- A description of the possible actions available to the agent.
- A description of what each action does; the formal name for this is the transition model.
- The goal test, which determines whether a given state is a goal state.
- A path cost function that assigns a numeric cost to each path.

# Components of a problem

The initial state, actions, and transition model implicitly define the state space of the problem: the set of all states reachable from the initial state by any sequence of actions. A path in the state space is a sequence of states connected by a sequence of actions.

A solution to a problem is an action sequence that leads from the initial state to a goal state. Solution quality is measured by the path cost function, and an optimal solution has the lowest path cost among all solutions.

# Infrastructure for search algorithms

Each node in the search tree stores the following information:

- STATE : the state in the state space to which the node corresponds.
- PARENT : the node in the search tree that generated this node.
- ACTION : the action that was applied to the parent to generate the node.
- PATH-COST : the cost, traditionally denoted by g(n), of the path from the initial state to the node, as indicated by the parent pointers.

# Measuring problem-solving performance

- Completeness: Is the algorithm guaranteed to find a solution when there is one?
- Optimality: Does the strategy find the optimal solution?
- Time complexity: How long does it take to find a solution?
- Space complexity: How much memory is needed to perform the search?

# Uninformed search strategies

- BFS: $O(b^{d+1})$ time, $O(b^d)$ space where the solution is at depth d, and b is the branching factor.
- Uniform-cost search: $O(b^{1+\lfloor C^*/\epsilon \rfloor})$ time and space, where $C^*$ is the cost of the optimal solution, and $\epsilon$ is the cost of each action.
- DFS: $O(b^m)$ time, $O(bm)$ space, where m is the maximum depth of the graph.

# Uninformed search strategies

## Depth-limited search

- DFS may fail in infinite search spaces.
- So we limit the depth of the search tree to some $l$, and do DFS.
- $O(b^l)$ time, $O(bl)$ space.
- Incomplete if $d > l$.
- Non-optimal if $d < l$.

# Uninformed search strategies

## Iterative-deepening search

- Keep on increasing $l$ starting from 1, and do Depth-limited search.
- Like BFS, it is complete when the branching factor is finite and optimal when the path cost is a nondecreasing function of the depth of the node.
- First level nodes generated once, second level nodes generated twice and so on.
- $O(b^d)$ time, $O(bd)$ space.

# Uninformed search strategies

## Bidirectional search

- Simultaneous BFS from the initial state and the goal.
- $O(b^{d/2})$ time and space.
- Applicable only when actions in the state space are reversible.