Artificial INtelligence (CSC261)

Prakash Neupane

Godawarai College
Itahari, Sunsari
Affiliated to TU, IOST

03/09/2023

Recap!

- Unit 2 introduced reflex agents, which are the simplest type of agents that rely on a direct mapping from states to actions.
- However, reflex agents may not perform well in environments with a large state space that would require too much storage or learning time.
- In contrast, goal-based agents consider the future consequences of their actions and evaluate their desirability in reaching a specific goal.

Definition: Searching

- Searching refers to the process of finding a particular item or piece of information within a given set of data or information.
- In computer science, searching often refers to the process of finding a specific piece of data within a collection of data.
- For example, searching can be used to find a particular record within a database, a specific file within a folder, a word within a document, or a particular value within an array or list.

Definition: Searching

- There are various searching algorithms used in computer science to perform searching operations efficiently.
- Some of the commonly used searching algorithms include linear search, binary search and so on.
- The choice of searching algorithm depends on factors such as the size of the data set, the distribution of data, and the type of data being searched.
- Efficient searching is important in many areas of computer science, including database management, information retrieval, and artificial intelligence.

Searching Problem

- A searching problem is a type of problem-solving task where an agent needs to find a sequence of actions to go from an initial state to a goal state.
- Searching problems can occur in domains such as robotics, planning, and gaming.
- The problem-solving process for searching problems involves generating a search tree, where each node represents a state in the problem domain, and edges represent possible actions that can be taken to transition between states.
- The goal state is typically specified in advance, and the agent's objective is to find a path from the initial state to the goal state that satisfies the problem constraints and objectives.

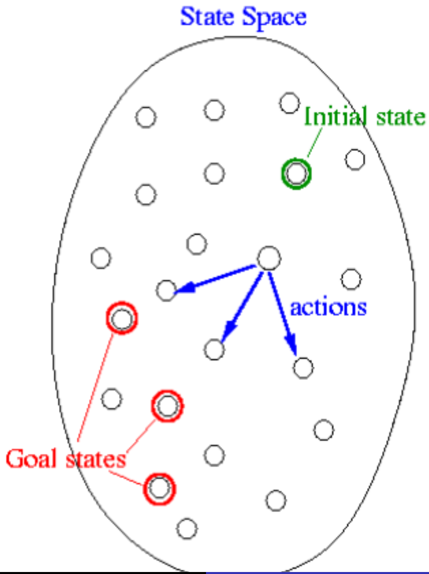
Searching Problem

- Searching problems can be challenging to solve, especially when the problem domain is complex, the state space is large, or the problem constraints are highly restrictive.
- Effective search algorithms and problem-solving techniques can help agents find optimal or near-optimal solutions to searching problems in a wide range of domains.

A search problem can be formally defined as follows:

- S: the full set of states
- So: the initial state
- G: set of goal states, G $\subseteq$ S
- A: S $\rightarrow$ S

Example: Search Problem



Problem Solving Agent

- Problem-solving agents are intelligent agents that analyze a problem, generate potential solutions, and select the best course of action to solve the problem.
- They can operate in a wide range of environments and can be used to address various types of problems, such as planning, scheduling, and decision making.
- Problem-solving agents use a range of techniques and algorithms to solve problems, such as search algorithms, constraint satisfaction, and optimization techniques.
- They may incorporate heuristics or domain-specific knowledge to improve their problem-solving efficiency and accuracy.

Problem Solving Agent

- The problem-solving process involves several steps, including **problem formulation, problem representation, search, and evaluation of potential solutions**.
- The agent identifies the problem and determines its constraints, objectives, and desired outcomes.
- It then converts the problem into a suitable representation that can be processed by the agent's algorithms.
- The agent then searches for potential solutions, evaluates them based on their desirability and feasibility, and selects the best solution to solve the problem.
- Problem-solving agents are critical in various fields, such as artificial intelligence, robotics, and computer science, where complex problems need to be solved efficiently and accurately.

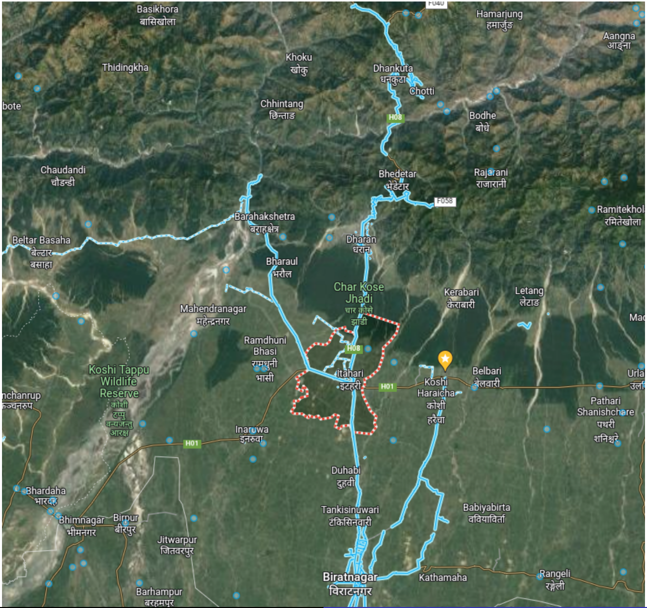
Problem as a state space search

- State space search is a problem-solving technique used in artificial intelligence and computer science to find a solution to a problem by exploring the set of possible states of a system.
- In state space search, the problem is represented as a set of states, with each state representing a particular configuration of the system.
- The search algorithm explores the state space by generating successor states for each state and selecting the ones that are most promising for finding a solution.

Problem formulation

- A problem can be formulated as a state space search by defining the following components:
  1. **Initial state:** This is the starting state of the problem, from which the search begins. It is the representation of the problem as it is given to the search algorithm.
  2. **Actions:** These are the possible operations that can be performed on the current state to move to the next state. Each action results in a transition from one state to another.
  3. **State Space:** This is the set of all possible states that can be reached by performing a sequence of actions starting from the initial state.
  4. **Goal Test:** This is a function that checks whether the current state is a goal state or not. The goal state is the state that satisfies the conditions of the problem.
  5. **Path Cost:** This is the cost associated with reaching a state. The path cost is a function that assigns a numerical value to each transition between states.

Route Finding Problem



Problem as a state space search

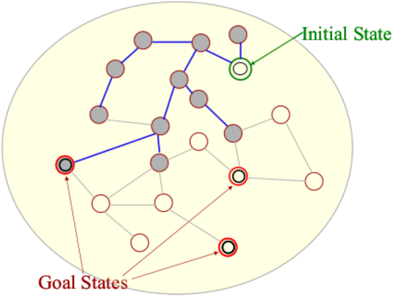
- A problem can be formulated as a state space search as follows:
  1. **States:** Locations (eg: Itahari, Biratnager, Dharan ...)
  2. **Initial state:** Starting point (eg: Biratnager)
  3. **Actions (Successor Function):** Transition form one location to another.
  4. **Goal Test:** Arrived at specific locations (eg: Vedatar)
  5. **Path Cost:** May includes total distance, financial cost, time, etc.

Well-defined Problem

- In AI, a search problem is a problem that can be solved by searching through a set of possible solutions to find the best or optimal solution.
- Well-defined search problems in AI typically have a defined:
  - starting state,
  - a set of possible actions or moves or successor functions,
  - Path and path cost
  - a goal state or condition, and
  - a set of rules that govern the legality of moves and transitions between states.

Example: Well-defined Problem

- **Pathfinding**: Given a map or a graph, the goal is to find the shortest path between two points.
- This problem can be solved using algorithms like Dijkstra's algorithm, A* search, or breadth-first search.



Search strategies

- In AI, search strategies are methods for systematically exploring a search space to find a solution to a well-defined problem.
- There are several common search strategies, each with its own strengths and weaknesses, and the choice of strategy often depends on the nature of the problem and the constraints involved.
- Here are some of the most common search strategies:
- Uninformed search (Blind)
  - Depth First Search, Breadth First Search, Depth Limited Search, Iterative Deepening Search, Bidirectional Search
- Informed Search (Heuristics)
  - Greedy Best first search, A* search, Hill Climbing, Simulated Annealing

Performance evaluation of search techniques

- Performance evaluation of search techniques in AI is an important step in determining the effectiveness of different search algorithms and their suitability for different types of problems.
  - **Completeness**: A search algorithm is considered complete if it is guaranteed to find a solution if one exists. Completeness is an important metric for problems where finding a solution is crucial, such as pathfinding or puzzle-solving.
  - **Optimality**: A search algorithm is considered optimal if it always finds the best possible solution. Optimality is an important metric for problems where there is a clear measure of goodness or quality, such as minimizing travel time or maximizing revenue.

Performance evaluation of search techniques

- Performance evaluation of search techniques in AI is an important step in determining the effectiveness of different search algorithms and their suitability for different types of problems.
  - **Time complexity**: This measures the amount of time it takes for a search algorithm to find a solution. Time complexity is an important metric for problems where efficiency is a concern, such as real-time systems or resource-constrained devices.
  - **Space complexity**: This measures the amount of memory or storage required by a search algorithm to find a solution. Space complexity is an important metric for problems where memory or storage is limited, such as embedded systems or mobile devices.

Performance evaluation of search techniques

- There are several notations used to measure time and space complexity of algorithms in artificial intelligence. These include:
  - **Big O notation (O)**: This notation expresses the upper bound of the growth rate of an algorithm's time or space complexity.
  - **Big Omega notation (Ω)**: This notation expresses the lower bound of the growth rate of an algorithm's time or space complexity.
  - **Big Theta notation (Θ)**: This notation expresses both the upper and lower bounds of the growth rate of an algorithm's time or space complexity.

References

S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Pearson, 2010.