DADABHOY INSTITUTE OF HIGHER EDUCATION
DIHE

ARTIFICIAL INTELLIGENCE

TEACHER: DR. ARSALAN KHAN

FINAL TERM EXAMINATION

FALL 2021

HASSAN JAVED
BSCS116/I-14/M01063

DATE: 27-JAN-2022

Question # 01

BREADTH SEARCH:

Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures. It starts at the tree root (or some arbitrary node of a graph, sometimes referred to as a 'search key'), and explores all of the neighbor nodes at the present depth prior to moving on to the nodes at the next depth level.

Uniform Cost Search:

UCS is different from BFS and DFS because here the costs come into play. In other words, traversing via different edges might not have the same cost. The goal is to find a path where the cumulative sum of costs is the least.

Cost of a node is defined as:

cost(node) = cumulative cost of all nodes from root

cost(root) = 0

DEPTH SEARCH:

Depth-first search (DFS) is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking.

DFS (Depth First Search) Features

1. DFS starts the traversal from the root node and explore the search as far as possible from the root node i.e depth wise.
2. DFS algorithm works in two stages. In the first stage, the visited vertices are pushed onto the stack and later on when there is no vertex further to visit those that are popped-off.

3. DFS search can be done with the help of stack i.e LIFO implementations.
4. DFS is not useful in finding shortest path. It is used to perform a traversal of general graph and the idea of DFS is to make a path as long as possible and then go back (backtrack) to add branches also as long as possible.
5. DFS is comparatively faster when compared to BFS.
6. The time complexity of DFS is O(V+E) where V stands for vertices and E stands for edges.
7. DFS requires comparatively less memory to BFS.
8. Some Applications of DFS include: Topological sorting, finding connected components, finding articulation points (cut vertices) of the graph, solving puzzles such as maze and Finding strongly connected components.

HEURISTIC SEARCH:

- Heuristic or informed search exploits additional knowledge about the problem that helps direct search to more promising paths.
- A heuristic function, h(n), provides an estimate of the cost of the path from a given node to the closest goal state. Must be zero if node represents a goal state.

  -Example: Straight-line distance from current location to the goal location in a road navigation problem.

- Many search problems are NP-complete so in the worst case still have exponential time complexity; however, a good heuristic can:

  -Find a solution for an average problem efficiently.

  -Find a reasonably good but not optimal solution efficiently.

LEARNING AGENTS:

Learning agent in AI is the agent which has ability to learn from its past experience. Learning agent perceive the environment and keep track every information. It starts from very basic knowledge and then able to learn incrementally from their environment.

Learning agent is composed of four components:

- Learning element: Learning element is responsible for the agent to learn from environment
- Critic: Learning element gather feedback from critic which explain how far agent is performing well according to the performance
- Performance element: Responsible for selecting external action
- Problem generator: Responsible for recommending actions that will help agent to new experience.

Question # 02

INFORMED SEARCH:

The algorithms of a uniformed AI do not consist of any additional data/ info regarding the goal node. It only contains the information provided during the definition of a problem. The plans required to reach from the start state to the goal state differ only on the basis of the length and order of the provided actions.

For example, Breadth-First Search and Depth-First Search.

Features of Uninformed Search in AI:

- It does not contain any additional data/ info.
- The information is provided to the AI during the definition of a problem.
- It can reach the goal state on the basis of the length and the order of actions performed.

- The AI doesn’t utilize any knowledge to search for the solution to a problem.
- A few examples of these include BFS (Breadth-First Search) and DFS (Depth-First Search).
- This type of AI takes more time to generate a solution for any problem.
- It is always complete.
- The total cost incurred is generally more than that of Informed Search in AI.
- It consumes a fairly moderate time to do the search.
- The implementation is lengthy.
- No suggestion is present for finding any solution.

INFORMED SEARCH:

The algorithms of an informed search contain information regarding the goal state. It helps an AI make more efficient and accurate searches. A function obtains this data/info to estimate the closeness of a state to its goal in the system.

For example, Graph Search and Greedy Search.

Features of Informed Search in AI:

- It consists of information regarding the goal state.
- It makes a search more efficient.
- A function obtains the data/info regarding the closeness of the current state of a search to its goal state.
- It utilizes knowledge for implementing the searching process.
- A few examples include graph search and greedy search.
- It incurs less cost.
- It may be complete or incomplete.
- A solution can be found much quicker.
- This type of search consumes less time.
- Implementation of such an AI is short and quick, not at all lengthy.
- The AI gets a direct suggestion about the solution of the search/ problem.

Question # 03

Both goal-based and utility-based agents have goals. However, having goals isn't effective (or efficient) enough, given that a goal-based agent may have several actions that can lead to the goals, but not all these actions are equally effective. So there's the need for an agent to perform the most effective action. And this is done by a utility-based agent.

That said, for an agent that exhibits the utility function, it maps each state after each action being taken nor performed efficiently and effectively.

Example

Consider two drones GG and UU, where GG is a goal-based and UU a utility-based agent. (The two drones have onboard computerized chips, so there is no need for ground control). These drones are sent on a mission and they have a goal. Both drones detect the given goal, but GG does not know which of its available actions is more efficient or effective. However, UU, based on its utility function, can select the most efficient or effective action.

Question # 04

ITERATIVE IMPROVEMENT ALGORITHM

Greedy techniques iteratively construct an optimal solution by building optimal solutions from smaller problems. Iterative improvement techniques build an optimal solution by iterative refinement of a feasible solution for the complete problem.

Feasible solutions are solutions that satisfy the constraints of the problem, for example using the denominations in the making change problem.

The objective function is the function that problem seeks to maximize or minimize.

Iterative improvement is frequently used in numerical problems, for example root finding or finding the maximum of a function. We will concentrate on iterative improve to graph problems.

Iterative improvements have difficulties:

1. Finding the initial solution (guess to the solution) can be easy, for example the empty set, or on the other hand it can be difficult.

2. The algorithm for refinements the guess may be difficult. The refinement must remain feasible and improve the objective function. Meaning they should not jump around and possibly diverge from the optimal solution.

3. The refinement may find a local optimal solution but not the global optimal solution, for example find the maximum by always going uphill.