B. Sc. (H) Computer Science Semester VI
Core Paper XIII – Artificial Intelligence

Topic:

- **Means-Ends Analysis**
- **A* Algorithm**

Means-Ends Analysis

We have studied the strategies which can reason either in forward or backward, but a mixture of the two directions is appropriate for solving a complex and large problem. Such a mixed strategy, make it possible that first to solve the major part of a problem and then go back and solve the small problems arise during combining the big parts of the problem. Such a technique is called Means-Ends Analysis.

Means-Ends Analysis is problem-solving techniques used in Artificial intelligence for limiting search in AI programs.

It is a mixture of Backward and forward search technique.

The MEA technique was first introduced in 1961 by Allen Newell, and Herbert A. Simon in their problem-solving computer program, which was named as General Problem Solver (GPS).

The MEA analysis process centered on the evaluation of the difference between the current state and goal state.

How means-ends analysis Works

The means-ends analysis process can be applied recursively for a problem. It is a strategy to control search in problem-solving. Following are the main Steps which describes the working of MEA technique for solving a problem.

First, evaluate the difference between Initial State and final State.

Select the various operators which can be applied for each difference.

Apply the operator at each difference, which reduces the difference between the current state and goal state.

Operator Subgoaling

In the MEA process, we detect the differences between the current state and goal state. Once these differences occur, then we can apply an operator to reduce the differences. But sometimes it is possible that an operator cannot be applied to the current state. So we create the subproblem of the current state, in which operator can be applied, such type of backward chaining in which operators are selected, and then sub goals are set up to establish the preconditions of the operator is called Operator Subgoaling.

Note: Please solve example on the topic discussed above from reference book [*]. (pg 94-96)

*Elaine Rich, Kevin Knight, & Shivashankar B Nair, Artificial Intelligence, McGraw Hill, 3rd ed.,2009.

Refer http://www.vssut.ac.in/lecture_notes/lecture1423725949.pdf and https://www.javatpoint.com. Also solve the given examples.

A* Algorithm

A* search is the most commonly known form of best-first search. It uses heuristic function h(n), and cost to reach the node n from the start state g(n). It has combined features of UCS and greedy best-first search, by which it solve the problem efficiently. A* search algorithm finds the shortest path through the search space using the heuristic function. This search algorithm expands less search tree and provides optimal result faster. A* algorithm is similar to UCS except that it uses g(n)+h(n) instead of g(n).

In A* search algorithm, we use search heuristic as well as the cost to reach the node. Hence we can combine both costs as following, and this sum is called as a fitness number.

Algorithm of A*

Step1: Place the starting node in the OPEN list.

Step 2: Check if the OPEN list is empty or not, if the list is empty then return failure and stops.

Step 3: Select the node from the OPEN list which has the smallest value of evaluation function (g+h), if node n is goal node then return success and stop, otherwise

Step 4: Expand node n and generate all of its successors, and put n into the closed list. For each successor n', check whether n' is already in the OPEN or CLOSED list, if not then compute evaluation function for n' and place into Open list.

Step 5: Else if node n' is already in OPEN and CLOSED, then it should be attached to the back pointer which reflects the lowest g(n') value.

Step 6: Return to Step 2.

Advantages

A* search algorithm is the best algorithm than other search algorithms.

A* search algorithm is optimal and complete.

This algorithm can solve very complex problems.

Disadvantages

It does not always produce the shortest path as it mostly based on heuristics and approximation.

A* search algorithm has some complexity issues.

The main drawback of A* is memory requirement as it keeps all generated nodes in the memory, so it is not practical for various large-scale problems.

Note: Please solve example on the topic discussed above from reference book [*].(pg 78-79)

*Elaine Rich, Kevin Knight, & Shivashankar B Nair, Artificial Intelligence, McGraw Hill, 3rd ed.,2009.

Refer http://www.vssut.ac.in/lecture_notes/lecture1423725949.pdf and https://www.javatpoint.com. Also solve the given examples.

Assignment

Q1. Define the following terms:
i) Agent Program
ii) Ideal Rational Agent

Q2. Enlist four capabilities needed by a computer to pass the Turing Test.

Q3. Give the Percept, Actions, Goals and Environments of the following Agent types:
i) Medical Diagnosis System
ii) Taxi driver

Q4. Determine which of the following are valid wffs:
(i) ((((P → Q) →~R) ↔ S) ∨ (T ∧ U))
(ii) wife(P(x))
(iii) ~~P
(iv) PQ →

Q5. Following is a knowledge base of PROLOG:
likes(george,food).
likes(george,wine).
likes(browny,wine).
likes(browny,george).
How do you add the following rules in the knowledge base?
i) Browny likes anything that George likes.
ii) Browny likes anyone who likes wine.

Q6. Differentiate between the following environment types:
i) Static and Dynamic

ii) Discrete and Continuous

Q7. Under what situations can hill climbing fail to find a solution?

Q8. Describe the operation of the A* Algorithm using an example.

Q9. Write Operators, Preconditions and Results for the following problem using Means-Ends Analysis.
A robot has to move a desk with two things on it from one room to another.

Q10. Find the most general unifier for

(i) t1 = P(f(a), g(X))
t2 = P(Y, Y)

(ii) t1 = P(f(g(X,a)), X)
t2 = P(f(Y), b)