Ahsanullah University of Science and Technology

Department: Computer Science and Engineering Course

Assignment Fall 2021

Course Number: CSE4107

Course Title: Artificial Intelligence

Student ID: 18.02.04.142

Student Name: S.M. TASNIMUL . HASAN

Student Section: B

Date of Submission: 28.07.2022

Signature of the Student: Tasnimul

Problem 1 : Why is Greedy Best first search said to be incomplete?

Answer : The greedy best first search algorithm tries to explore the node that is closest to the goal. This algorithm evaluates nodes by using the heuristic function h(n). That is, the evaluation function is equal to the heuristic function, $f(n) = h(n)$. With the help of best first search at each step, we can choose the most promising node. In this algorithm we expand the node which is closest to the goal node and the closest cost is estimated by heuristic function.

Greedy best first search is incomplete even if the given state space is finite. The worst case time complexity of greedy best first is $O(b^m)$ and the worst case space complexity is $O(b^m)$. where m is the maximum depth of the search space. This algorithm can behave as an unguided depth-first search in the worst case scenario. It can get stuck in a loop as DFS and this algorithm is not optimal. In finite state space the program can get stuck in loops unless we use a closed list, that's why this greedy best first search is said to be incomplete.

Problem 2 : Present an analysis targeting negative aspects of the A* search strategy.

Answer : A* search is the most commonly known form of best-first search. It is one of the best and popular technique used in path-finding and graph traversal. A* search algorithm finds the shortest path through the search space using the heuristic function. In this 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.

$$f(n) = g(n) + h(n)$$

Here,

$f(n)$ = Estimated cost of the cheapest solution

$g(n)$ = Cost of reach node n from start state

$h(n)$ = Cost of reach node n to goal node

Negative aspects of the A* search strategy :

A* search algorithm does not always produce the shortest path as it mostly based on heuristics and approximation. The performance of A* search is dependent on accuracy of heuristic algorithm used to compute the function h(n).

A search algorithm is said to be admisible if it is guaranteed to return an optimal solution. If the heuristic function used by A* is admisible, then A* is admisible. When A* terminates its search, it has found a path from start to goal whose actual cost is lower than the estimated cost of any path from start to goal through any open node. When the heuristic is admisible, those estimates are optimistic. So A* can safely ignore those nodes because they cannot possibly lead to a cheaper solution then the one is already has. In other words, A* will never overlook the possibility of a lower cost path from start to goal and so it will continue to search until no such possibilities exists.

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.

Problem 3 : Perform a comparative study between Hill-climbing local search and Genetic algorithms.

Answer : Hill climbing algorithm is a local search algorithm which continuously moves in the direction of increasing elevation / value to find the peak of the mountain or best solution to the problem. It terminates when it reaches a peak value where no neighbours has a higher value. On the other hand, a genetic algorithm (GA) is a search based optimization technique based on the principles of Genetics and natural selection. This algorithm is used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems in research and in Machine Learning.

Hill climbing algorithm is a technique which is used for optimizing the mathematical problems. One of the widely discussed examples of Hill climbing algorithm is Traveling Salesman Problem in which we need to minimize the distance travelled by the salesman. It is also called greedy local search as it only looks to its good immediate neighbour's state and not beyond that. From the several methods of TSP, Hill Climbing algorithm has good performance in local searching. Starting from defining the group, deciding the better search area up to iterating from level to level, the result of each level is taken from best and then compared, so they can get more optimal. This TSP method is used to determine the nodes that has been given the distance among other nodes by comparing the existing node based on selection of the shortest distance from the initial position.

Besides hill climbing to solve TSP problem, there is also genetic algorithm. Genetic algorithm is powerful and flexible meta-heuristic as well as the relatively new type of algorithm by adopting the idea of natural selection and genetic changes naturally. This algorithm is known as a tool that can solve combinatorial in optimization problems such as TSP. Genetic algorithm worked by solving the problem of appropriate population selection that will later perform iterations (generations by applying the 3 basic operations: selection, crossover and mutation).