01. Heuristics function and 02. Best first search algorithm

CSE-0408 Summer 2021

Ashraf uddin Mamun [UG02-44-17-044]
Department of Computer Science and Engineering
State University of Bangladesh (SUB)
Dhaka, Bangladesh
mamun336929@gmail.com

I. Overview

Heuristics function: Heuristic is a function which is used in Informed Search, and it finds the most promising path. It takes the current state of the agent as its input and produces the estimation of how close agent is from the goal. The heuristic method, however, might not always give the best solution, but it guaranteed to find a good solution in reasonable time. Heuristic function estimates how close a state is to the goal. It is represented by h(n), and it calculates the cost of an optimal path between the pair of states. The value of the heuristic function is always positive.

II. Proposed methodology:

Heuristics don't always lead to a lower cost. However, those that don't overestimate the true or the lowest possible cost of a solution are called admissible heuristics. This characteristic can guarantee the optimality of the solution. An admissible heuristic can be found by simplifying the original problem in terms of its constraints, reducing it to a less constrained problem. As an example, let's consider the eight-puzzle problem:

III. Rules for solving problems with Heuristic Function:

Use of heuristic function in a heuristic search algorithm leads to following properties of a heuristic search algorithm: Admissible Condition: An algorithm is said to be admissible, if it returns an optimal solution. Completeness: An algorithm is said to be complete, if it terminates with a solution (if the solution exists). Dominance Property: If there are two admissible heuristic algorithms A1 and A2 having h1 and h2 heuristic functions, then A1 is said to dominate A2 if h1 is better than h2 for all the values of node n. Optimality Property: If an algorithm is complete, admissible, and dominating other algorithms, it will be the best one and will definitely give an optimal solution.

IV. Advantages:

I. It can provide some quick and relatively inexpensive feedback to designers.
II. You can obtain feedback early in the design process.
III. Assigning the correct heuristic can help suggest the best corrective measures to designers.
IV. You can use it together with other usability testing methodologies.
V. You can conduct usability testing to further examine potential issues.

V. Disadvantages:

I. It requires knowledge and experience to apply the heuristics effectively.
II. Trained usability experts are sometimes hard to find and can be expensive.
III. You should use multiple experts and aggregate their results.
IV. The evaluation may identify more minor issues and fewer major issues.

VI. Overview of Best first search algorithm:

Best first search algorithm: If we consider searching as a form of traversal in a graph, an uninformed search algorithm would blindly traverse to the next node in a given manner without considering the cost associated with that step. An informed search, like Best first search, on the other hand would use an evaluation function to decide which among the various available nodes is the most promising (or 'BEST') before traversing to that node.

VII. Proposed methodology:

The Best first search uses the concept of a Priority queue and heuristic search. To search the graph space, the BFS method uses two lists for tracking the traversal. An 'Open' list which keeps track of the current 'immediate' nodes available for traversal and 'CLOSED' list that keeps track of the nodes already traversed.

VIII. Rules for solving problems with Best first search algorithm:

Step 1: Place the starting node into the OPEN list.
Step 2: If the OPEN list is empty, Stop and return failure.
Step 3: Remove the node n, from the OPEN list which has the lowest value of h(n), and places it in the CLOSED list.
Step 4: Expand the node n, and generate the successors of node n.
Step 5: Check each successor of node n, and find whether any node is a goal node or not. If any successor node is goal node, then return success and terminate the search, else proceed to Step 6.
Step 6: For each successor node, algorithm checks for evaluation function f(n), and then check if the node has been in either OPEN or CLOSED list. If the node has not been in both list, then add it to the OPEN list.
Step 7: Return to Step 2.

IX. Advantages:

I. Best first search can switch between BFS and DFS by gaining the advantages of both the algorithms.
II. This algorithm is more efficient than BFS and DFS algorithms.

X. Disadvantages:

I. It can behave as an unguided depth-first search in the worst-case scenario.
II. It can get stuck in a loop as DFS. III. This algorithm is not optimal.

Acknowledgment

I would like to thank my honourable**Khan Md. Hasib Sir** for his time, generosity and critical insights into this project.