International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

A Study of Uniformed State Space Search in Artificial Intelligence

Dr. G. Vijaya Lakshmi

Assistant Professor, Department of Computer Science, Vikrama Simhapuri University, Nellore

Abstract: ***In Artificial Intelligence a state space search is used find the optimal solution for a given problem by systematically navigating all possible states from root state to the goal state. This technique is applied in various applications across many domains like Puzzle Solving, Path finding in games, Robotics, natural language processing, chess, solving puzzles, medical diagnosis etc. In this paper we study various uninformed state space search techniques with examples by finding their time and space complexities.***

Keywords: BFS, Depth limit, DFS, IDFS, state space search, uninformed search

1. Introduction

In artificial intelligence, a state space search is a method that is used to solve problems by exploring through a series of states and transitions until it reaches the goal of the state. It includes various features in finding the effective solution. They are

1) Exhaustiveness: Exploring all possible solution until it reaches the goal.
2) Completeness: If a search algorithm ensures that there is at least one solution for any random input, it is considered complete.
3) Optimality: searching through state space will gives the optimal solution.

2. State Space Search Techniques

In artificial intelligence, state space search is a core technique that navigate systematically all possible states to find the optimal solution. These state space search is classified into two types

- **Uninformed search:** also called as brute force search or blind technique, where it does not have any information to solve the problem.
- **Informed search:** also called as heuristic search, with the information it finds the solution from initial node to goal node.
- There are different Search Algorithms in AI. they are stated as below:

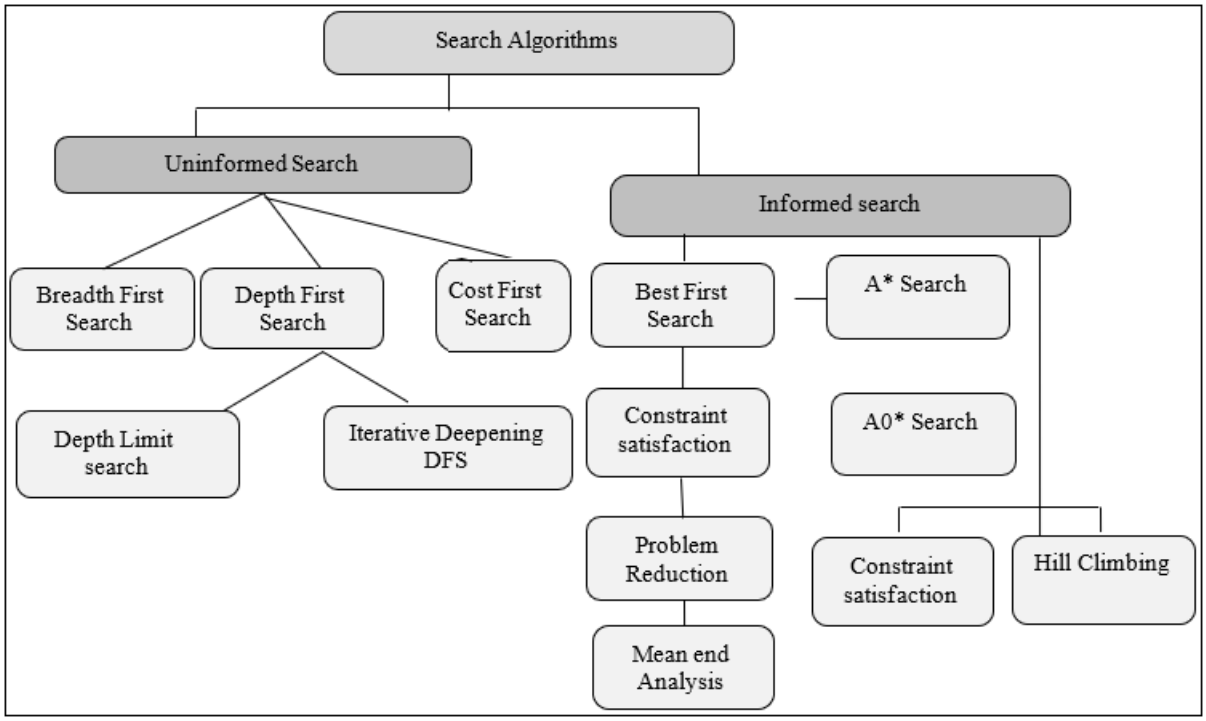

Figure 1: State Space Search Algorithms

In this paper, we discussed the informal search algorithms with the following given graph

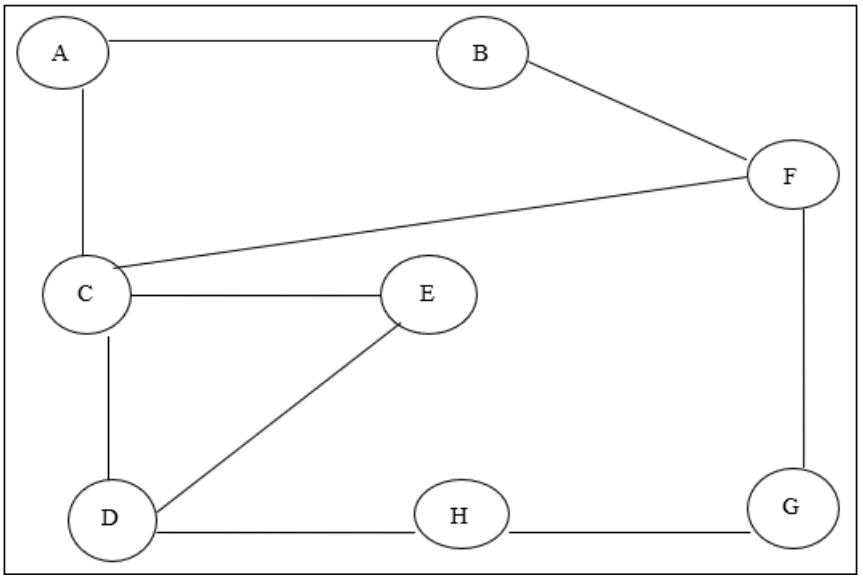

Figure 2: Graph

Depth First Search Algorithm
In Artificial Intelligence (AI) DFS, a graph traversal or tree traversal algorithm uses a stack data structure. It plays a significant role in problem - solving, path finding tasks, network analysis, game - solving problems like sudoku, web crawling etc. In this approach, it starts from initial node and from then it explores all the paths as far as possible before backtracking.

- Working of DFS algorithm is follows as:

1) Create a stack with a size equal to the graph's total number of nodes
2) Select any node to serve as the initial point of the traversal. Push the node into the stack
3) The third step is to explore its neighboring nodes & push the non - visited node at the top of the stack
4) Repeat step iii until there are no other nodes to visit.
5) Now pop the node which is at the top from the stack using backtracking.
6) Continue using steps iii, iv & v until the stack is empty.

For the given graph, the DFS search is as follows: A → B → F → C → E → D→ H → G

Drawbacks: it is not optimal because if it has huge levels and it goes on searching nodes till depth node till it finds the goal node. Therefore the time complexity is too high.

Time & Space Complexity: The time complexity is O (V + E), where V is the number of vertices and E is the number of edges & It store the stack of nodes on the path from the root to the current node the space complexity is O (V).

Depth - Limit Search
It is a DFS variation that explores a search tree or graph up to predetermined depth limit. In this algorithm it helps to avoid infinite paths in graphs with cycles. The time and space complexity will depend on the depth limit.
For the given tree if the start node is A, goal node is F and depth limit is 1, the output will be
A- > B - > F

Drawback: Finding a solution is unlikely to occur, and it is possible that many states will continue to recur. When the DFS algorithm searches deeply, it may enter into infinite loop.

Time & Space Complexity
The Time & space Complexity is O (b^l) is O (bxl) respectively where b is known as the branching factor (number of children at each node) and l is the given depth limit.

Iterative Deepening DFS
This algorithm is the combination of both DFS & BFS. It uses Stack to perform operations. Initially the depth limit will be zero. Until the goal node is found, depth - limited DFS is applied repeatedly, increasing the depth limit in each iteration.

For the given graph let the goal Node is E and depth limit is 1, then the path will be A - > C - > E

Drawbacks: the time complexity will be high for large It may explore at each and every depth level, leading to inefficiency and this algortithm is not suitable for graphs if there are unknown levels or depth

Time and space complexity
Consider a tree with a branching factor of "b" (the number of children of each node) and a depth of "d, " or bd nodes. . Nodes on the lowest level are extended once in an iterative deepening search, those on the next lowest level are extended twice, and so on, until the search tree's root is expanded d+1 times. Therefore, the Time and space complexity is O (b^d) & O (d) respectively.

Breadth First Search
Before traversing the nodes at the next depth level, the method known as Breadth - First Search (BFS) algorithm thoroughly examines every node at a specific depth. Beginning from the source node, BFS traverses all of its neighboring nodes first, moving level by level to their neighbors. BFS uses Queue data structure where newly explored nodes are added to the queue.

With this approach BFS effectively find the optimal path for the graphs or tree exploring all possibilities.

For the given graph the bfs path is: A - >B - >C - >F - >E - >D - >G - >H

Drawbacks: The major drawback is its memory consumption, as it needs to explore and store all nodes at the current depth level, which can grow exponentially in large search spaces.

Time & Space Complexity
The time complexity is O (V + E), and space complexity is O (V), where **V** is the number of vertices and **E** is the number of edges.

Cost First Search
Cost search is a tree or graph search algorithm used in artificial intelligence that finds the path by exploring the nodes from starting node to goal node it expands the nodes with lowest cumulative path cost . The UCS algorithm is a variant of also called Dijkstra's algorithm which works on a weighted graph that is edges have different costs). This algorithm explores from source node to goal node based on the cost to find the minimum cumulative cost of the path. It is also called a brute force approach since it do not have prior information about the path or node.

Drawbacks:
It will be Memory intensive since it stores all the path cost it explored which leads to memory intensive if the graph it too large and before traversal it requires prior knowledge of all the edge cost.
For instance, consider the cost path for the figure 1:

Table 1: Path cost

Nodes	Cost
A - >B	3
A - >C	5
C - >D	4
D - >H	1
H - >G	3
D - >E	9
C - >E	5
C - >F	10
F - >G	2
B - >F	13

In this algorithm we first choose from a table a priority node and assign the path cost as 0. Let the node be A and cost as 0. then it expands the minimum cumulative cost of the path until it reaches the goal node. then for the figure 1 the path will be A - >B - >F - >G.

Time & Space Complexity
Both time and space complexity are exponential in the size of the data. O (bd) where b is the branching factor of the search tree, and d is the depth of the goal node.

3. Conclusion

In this paper we studied various uninformed search algorithms with a given graph. This algorithms assists in finding the best path from root node to goal node. we determined the time and space complexity for every search algorithm. Besides, every algorithm has its own disadvantages These algorithms enable AI professionals to use them efficiently for a variety of tasks, such as route planning in logistics and path finding in games, web crawling etc.

References

[1] Artificial Intelligence: A Modern Approach, 4th US ed.
[2] Chowdhary KR, Chowdhary KR (2020) Introducing artificial intelligence.
[3] Handy Permana, S. D. et al. (2018) 'Comparative Analysis of Pathfinding Algorithms A, Dijkstra, and BFS on Maze Runner Game', IJISTECH (International Journal of Information System & amp; Technology), 1 (2), pp.1. doi: 10.30645/ijistech. v1i2.7
[4] Mehta, P. et al. A review on Algorithms for Path finding in Computer Games A Review on Algorithms for Pathfinding in Computer Games
[5] International Journal of Artificial Intelligence & Robotics (IJAIR) Optimization of Breadth - First Search Algorithm for Path Solutions in Mazyin Games E - ISSN: 2686 - 6269 Vol.3, No.2, 2021, pp.58 - 66.
[6] A. N. Putri, "Search the blind breadth first search algorithm in 3d game engine maze third person shooter android based on intelligent agent". Transformatika Journal, Vol.14, No.1, Jul.2016.
[7] R. Zhou and E. A. Hansen, "Breadth - first heuristic search, " Artif. Intell., vol.170, no.4–5, pp.385–408, 2006.
[8] Balaji et. al", International Journal of Creative Research Thoughts, "Navigating The World Of Graphs: Novel Applications Of Bfs And Dfs Algorthims, Volume 12, Issue 5 May 2024 | ISSN: 2320 - 2882.