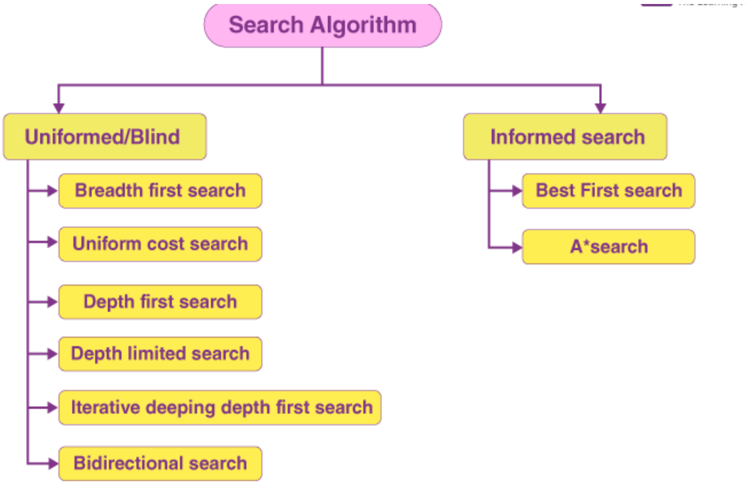
KEY ALOGARITHMS AND TECHNIQUES USED IN ARTIFICIAL INTELLIGENCE FOR SEARCH PROBLEMS

An algorithm is a set of rules and regulation used to solve a particular problem. Search algorithms are algorithms that help in solving search problems. A search problem is a problem that consists of a search state, start state and the goal state. Some of the types of search algorithms used for solving search problems are informed search and uninformed search algorithms.



UNINFORMED SEARCH

This is also known as the blind search or brute force algorithm. This a class of search algorithms that do not use specific knowledge or heuristics about the problem domain to guide their search hence generating the search tree. Uninformed search is a class of general purpose algorithms that operates in a brute force way. They do not have additional information about state or search space.

TYPES OF UNINFORMED SEARCH ALOGARITHMS USED FOR SOLVING SEARCH PROBLEMS.

Breadth-First Search algorithm

Breadth-first search is the name given to an algorithm that searches a tree or a graph in a breadth-first manner. Before going on to nodes of the next level, the Breadth-first search algorithm starts searching from the tree ‘s root node and extends all successor nodes at the current level. Breadth first search is implemented using the Queue which is FIFO meaning First In Last Out.

Properties of Breadth-first Search

Time complexity: The number of nodes transverse in Breadth first search till the shallowest node can be used to determine an algorithm ‘s time complexity i.e. where d represents the depth of the shallowest solution and b represents anode at each state.

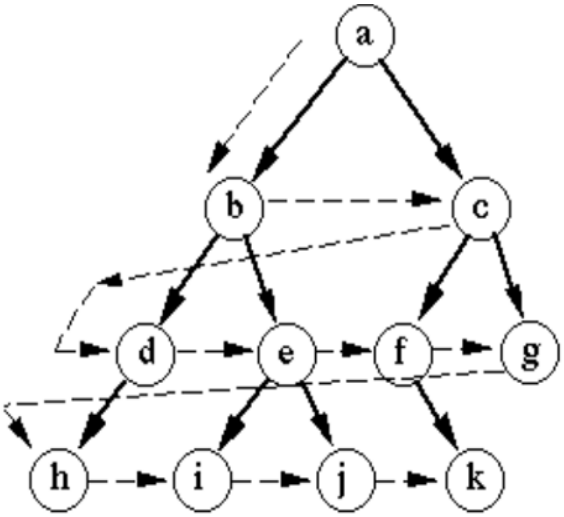
Space complexity: Space complexity of Breadth-first search algorithm is given by the memory size of frontier which is O (b^d).

Completeness: Breadth-first search is complete, which means it will discover the solution if the shallowest target node is at finite depth.

Optimality: Breadth-first search is optimal if the path cost is anon decreasing function of the node ‘s depth.

Example of breadth first algorithm

Assuming k is the goal state from this tree diagram



Using the above tree diagram, the breadth first search begins searching from the root node and extends to other successor nodes until reaching the goal state.

Therefore, the final search solution of BFS from the above tree diagram is a, b, c, d, e, f, g, h, I, j, k.

Advantages of Breadth-first search algorithm

It’s simple and easy to implement and understand because it has the shortest path from the starting point to the goal.

Easily programmable.

Applications of Breadth first search in different domains and applications

Breadth first search is applied in robotics for path planning and navigation tasks.

Breadth first search is applied in genetics and biology to model the relationships in biological systems representing interactions between different elements.

Breadth-first search is employed in image processing tasks, such as region growing or connected component labeling.

Breadth-first search is applied in game development to search for paths, explore game maps, and perform AI descions.

it is also applied to explore social networks, identifying connections and degrees of separation between individuals.

Breadth-first search is applied in Graph transversal to transverse graphs and explore all vertices at current depth before moving on to the next level.

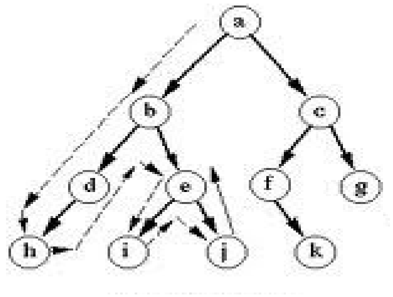
It is also used to find the shortest path between two nodes in an unweighted graph.

Depth-first search

Depth-first search algorithm is an algorithm for traversing or searching tree or graph data structures. It starts at the root node and explore as far as possible along each branch before backtracking. Depth first search uses a stack for implementation which is LIFO meaning Last In Last Out hence uses less memory than the Breadth first search. The time complexity of depth- first search is O(V+E) where V represents the number of vertices and E represents the number of edges in the graph.

Example of Depth-first search

Assuming j is the goal from this example



Depth first search starts its search from the root node until it hits a dead end then backtracks

Therefore, the final search solution for the DFS algorithm for the above tree diagram is a, b, d, h, e, I, j.

Application of the Depth- first search algorithm in different domains and applications

Detecting cycle in a graph to check the back edges of the graph is done using Depth-first search

Depth-first search is applied in the implementation of web crawlers to explore links on the website

It is also used in backtracking logarithms.

Depth first search is applied in maze generation to generate random mazes.

Depth first search can also be used in model checking ensuring that the model of the system meets a certain set of properties.

It is also used to check if the graph is bipartite

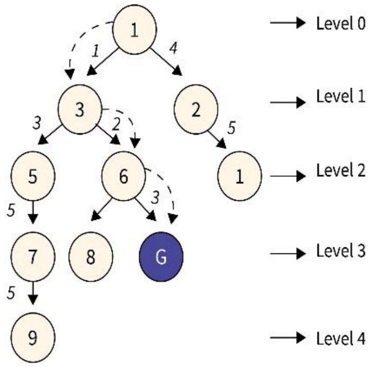
Depth first search is used for path finding to find the bath between two vertices.

Uniform cost search algorithm

Uniform cost search is an algorithm that explores a graph by gradually expanding nodes starting from the initial node and moving towards the goal node while considering the cost associated with each edge or step. It follows a priority queue where the one with the lowest cost is considered to find the source of the destination.

Example of uniform cost search algorithm

Assuming G is the goal



Since when using Uniform cost search, the lowest cost is considered when choosing the route to reach the goal state.

The route to the goal state for the above tree diagram is 1,3,6, G

Applications of uniform cost search in different domains and algorithm.

Uniform cost search is applied in resource allocation i.e. when resources have associated costs, uniform cost search can be applied to allocate resources optimally minimizing the overall cost.

It is also applied in network routings to find the least-cost path in computer networks there the edges represent communication links with associated costs

Uniform cost search is also applied in natural language processing in application like language translation to find the most effective translation of languages

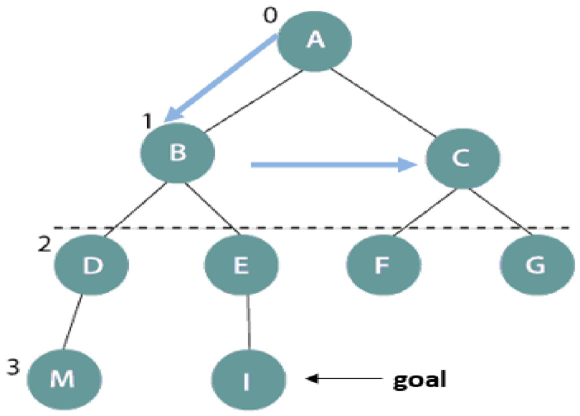
It is also applied in puzzle solving where each move has a different cost, such as sliding title puzzles.

Depth-limited search.

This an uniformed search algorithm where the search is limited at a certain level of the tree. It is almost the same as the depth first search algorithm with a depth limit before backtracking hence limiting the maximum depth of exploration. Depth limited search uses less memory compared to depth first search.

Example of the Depth limited search algorithm

Here the search is limited on level 2 before backtracking



The output is A, B, C.

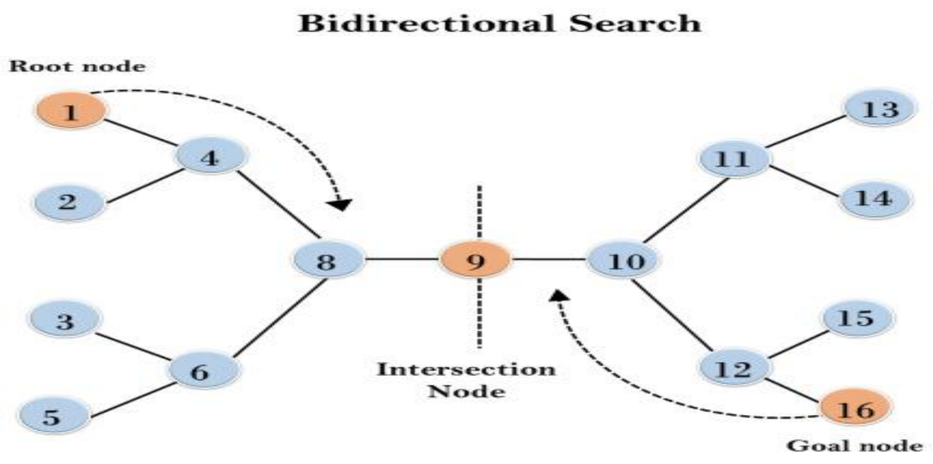
Applications of depth limited search in different domains and applications

It is applied in game playing like chess to limit the number of movements ahead hence making the game interesting

It is also applied in robotics path planning with in a limited distance.

Bidirectional search.

This is a search algorithm in which the search is conducted simultaneously from the initial state and the goal state. It has the forward search and the backward search.



INFORMED SEARCH ALOGARITHMS

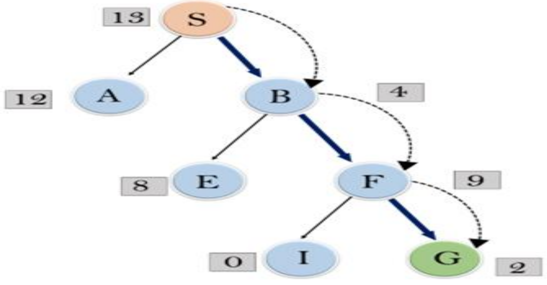
These algorithms are called also called heuristic search algorithms. It is used to expand the nodes using heuristic values. It contains the information about the goal state making its search easier thereby using specific knowledge to help it through its search of a problem space.

TYPPES OF INFORMED SEARCH ALOGARITHMS.

A* search algorithm.

This is an informed search algorithm that is used to find the shortest path through the search space using the heuristic function. The formula for A* is f(n)= g(n)+h(n) where f(n) is the total cost of the path through node n to the goal. g(n) is the cost of the path from the start node to node n, h(n) is the heuristic that estimates the cost from node n to the goal. It searches for the shortest path between the initial state and the final state.it is applied in network routing protocols to calculate the best route between two nodes.

Assuming G is the goal state



Considering the above formula of f(n)=g(n)+h(n), the route to the goal of the above A* algorithm is S (13), B (4), F (9), G (2) while considering the lowest total cost to reach the final destination which is the goal.

Best-First search

This is also known as the greedy search algorithm. It is implemented using a priority queue and heuristic search. It has two lists that is the open list and closed list. It is used in finding the shortest path from a given starting node to a goal node in a graphs

Other algorithms include the following

Binary search algorithms.

This is a type of algorithm that is used to find the position of a specific value contained in a sorted array and it works on the principal of divide and conquer hence becoming the best search algorithm.

Linear search algorithm.

This is a type of search algorithm that works by iterating through an array or a list until the target elements are found. Linear search logarithms is applied in phonebook search to find the person's name and phone number giving it a high efficient.

Constraint satisfaction problem algorithm

These are used to find a solution that meets a set of constraints and also finding values for a group of variables that fulfil a set of instructions or rules. They are also applied in resource allocation and scheduling to solve different constraints.

In conclusion the above algorithms are applied in different domains and application but some have the same tasks applications in those particular domains hence increasing the efficiency and effectiveness to solve different search problems.