**A\* Search Algorithm in Artificial Intelligence**

A\* is an extended Breadth-first search (BFS) Algorithm that prioritizes the shortest routes that it can reach first and then the other routes. Our goal is to reach the endpoint in the shortest span of time. It is a part of Informed searching where we already have some knowledge about the problem statements and the domains. This knowledge is known as heuristic. It is optimal and complete. Optimal means that A\* is sure to give the best solution that is available for the problem. Complete means that A\* can find all the possible solutions that exist for the problem. A\* decides to take up a step only if it is convincingly sensible and reasonable as per its functions.

**A\* Algorithm Equation**

To find the shortest path, we can use three parameters F(n), G(n), H(n), which follow the below equation:

F = G+ H

F is used to find the least cost from one node to the other and is responsible to find the optimal path between source and destination.

G is the cost of moving from node to another and it changes for every movement from one node to other. This is an actual cost.

H is the heuristic path between current node to destination. It has no actual cost but it is the assumption cost from the node to destination. It is like a node is saying to the other node that it will help in reaching the goal for an estimate cost. By default, the heuristic value of the goal state is zero.

**Drawbacks of A\* algorithm**

A\* is slow and also the space it requires is a lot as it saves all the possible paths that are available.

The speed execution of A\* search is highly dependent on the accuracy of the heuristic algorithm that is used to compute h(n).

**Comparison between Dijkstra’s algorithm and A\* algorithm**

Dijkstra does not know when to stop as it does not know which is the best path and tends to compute inefficiently.

A\* computes the best possible way and stops when the best route is found. This makes the computations effective and efficient.