



e-Yantra Robotics Competition - 2018

NSTask1Report<NS#0263>

Author Name(s)	Rohit Vasant Patil, Brijesh Kumar Patra, Kunal Kishore Sahu, Debasish Panigrahy
Team ID	NS#0263
Date	28.11.2018

Q1. Describe the path planning algorithm you have chosen.

Ans—We have used BFS (Breadth First Search) algorithm for path following. The aim of BFS algorithm traverse the node as close as possible to the root node. Breadth-first search starts by searching a start node, followed by its adjacent nodes, then all nodes that can be reached by a path from the start node containing two edges, three edges, and so on. Formally, the BFS algorithm visits all vertices in a graph 'G' that are 'k' edges away from the source vertex 's' before visiting any vertex 'k+1' edges away. This is done until no more vertices are reachable from 's'.

For a graph $G = (V, E)$ and a source vertex 's', breadth-first search traverses the edges of graph 'G' to find all reachable vertices from 's'. It also computes the shortest distance to any reachable vertex. Any path between two points in a breadth-first search tree corresponds to the shortest path from the root 's' to any other node 'v'.

Q2. Describe the algorithm's specific implementation i.e. how have you implemented it in your task?

Ans— First of all we numbered all the nodes and represented the arena as a non-weighted graph in form of a matrix as follows :- If there is an edge between node 'i' and node 'j' then $graph[i, j] = 1$ and if there is no edge between node 'i' and node 'j' then $graph[i, j] = 0$.

All the nodes at pickup zone and deposit zones (D1 and D2) are stored in separate arrays.

Every time a path is found using BFS it is stored in the path [] array.

A variable current_node is used to keep track of current node while traversing the path.

A function (travel_path) is defined for traversing the path found by BFS algorithm. In this function we have defined for each node that in which direction and by how many nodes the

bot should move to reach the next node in path. The direction is specified based on the previous node and the next node.

In main Task_1_2 logic first using BFS we calculated the path from start node (node 0) to the first node in pickup zone array. Our main objective is to pick the nuts from PICK UP ZONE and place them in their appropriate DEPOSIT ZONES. While traversing the path if any obstacle is found the bot returns to the previous node and again calculate the new path for the destination and this time the edge with the obstacle is not considered i.e. if there is an obstacle between node 'i' and node 'j' then graph [i , j] is made 0.

After reaching a pickup node it checks for the NUT if nut is present then it detects its color and according to the color it goes to the deposit zone and places the NUT and if NUT is NOT present then it checks in the next pickup node. This is repeated until all NUTs are deposited in the appropriate deposit zones.

Once all the pickup zones are covered or both green and red nut are deposited in deposit zones the bot returns to the starting position.

Video URL: <https://youtu.be/1CHsk4eqCwU>