



# e-Yantra Robotics Competition - 2018

## NS Task 1 Report <5405>

Author Name(s)	Rashmil Panchani
Team ID	5405
Date	28/11/2018

### Q1. Describe the path planning algorithm you have chosen.

The path planning algorithm that we have used in this task is A\* Algorithm because it is fairly flexible

and can be used over wide range of contexts. It's like Dijkstra's algorithm in that it can be used to

find a shortest path. It's like Greedy Best-First-Search in that it can use a heuristic to guide itself.

Thus it is a combination of Dijkstra's and Best-First-Search algorithms. It uses the sum of 'cost' (distance from source to the current node, defined by  $g(x)$ ) as well as 'heuristic' (estimated distance from current node to destination, defined by  $h(x)$ ).

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

We assumed the arena to be a weighted graph with nodes as the vertices of the graph. Each node has been

assigned a co-ordinate and the weight is calculated by using the distance formula between two nodes.

Since the heuristic can be any function, we defined the heuristic to be the distance between two nodes using the distance formula.

The neighbours of the node are stored in a vector and the connections between them are initialized using an adjacency matrix such that if connection exists then the value is '1', else it is '0'. This provides

us with the path when starting and ending node has been specified.

The nodes in the path are stored in a list and the current and the next node is retrieved from it and depending on the orientation of bot, right turn and left turn commands are executed.

Whenever an obstacle is encountered the bot goes back to the previous node and deletes the connection between

the nodes in which the obstacle is placed so that it won't traverse the same path again. On reaching the previous node, it will again follow A\* algorithm to find a new path.

On encountering the nuts, pick function is executed and then the bot is directed towards the dropping zone

depending on the colour of the nut by following A\* algorithm.

In this manner the bot will pick and place all the nuts and finally come back to the starting position.

Video link : <https://youtu.be/ixvJDzh1ZvE>