RBE550 – MOTION PLANNING
ASSIGNMENT: FLATLAND

Submitted by: Suketu Parekh

➢ Algorithm Overview:

a) Random Planner:
   1. Start at (1,1)
   2. For any random direction, check for already visited cell or obstacle
   3. Move to any connected random cell
   4. Start from 1. (Until 1000 iterations or reaching (128,128))

b) Breadth First Search (BFS):
   1. Start at (1,1)
   2. Add all the **valid** neighbouring cells of the current node into a queue
   3. Make a separate matrix for Parent cells
   4. Traverse through the queue until reached at (128,128)
   5. Traverse through **specific** Parent cells in reverse to get to (1,1)
   6. Add all the specific elements to the Path matrix

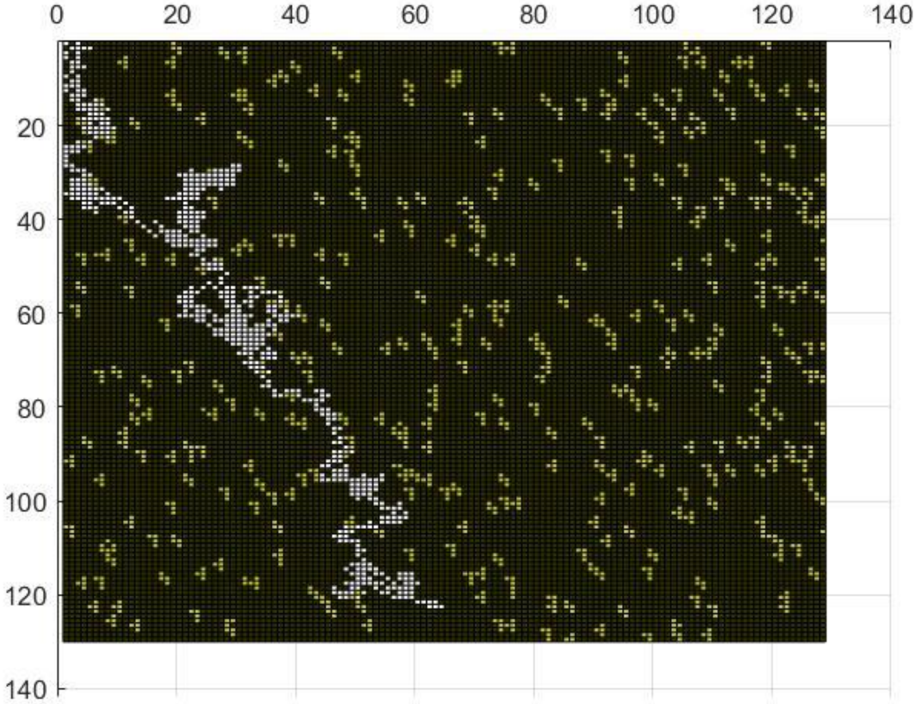
c) Depth First Search (DFS):
   1. Start at (1,1)
   2. Add one of the **valid** neighbours of current cell into a stack
   3. Move on to the top of the stack
   4. Check for validity of the next cell before pushing into stack
   5. Check for local **blockages** until (128,128) is reached

d) Dijkstra's Algorithm:
   1. Start at (1,1)
   2. Add all the **valid** neighbouring cells of the current node into a queue
   3. Make a **Cost** array to account for the cost of visiting each cell
   4. Make a separate matrix for Parent cells
   5. Traverse through the queue until reached at (128,128)
   6. Traverse through the path with the least cost of (128,128)

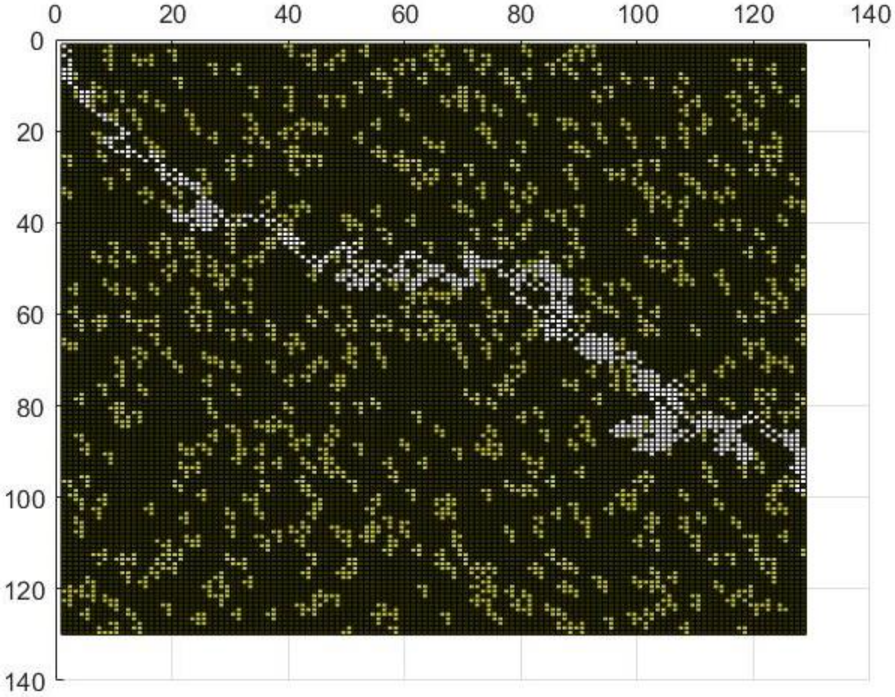
➢ Output Figures:

a) Random Planner

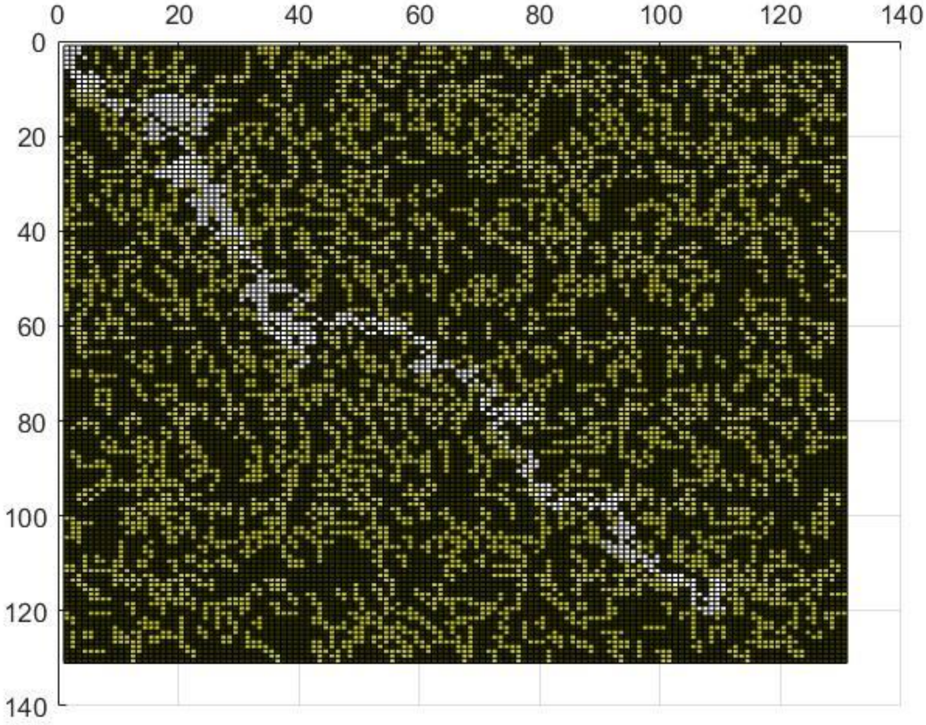
Occupancy = 10%



Occupancy = 20%

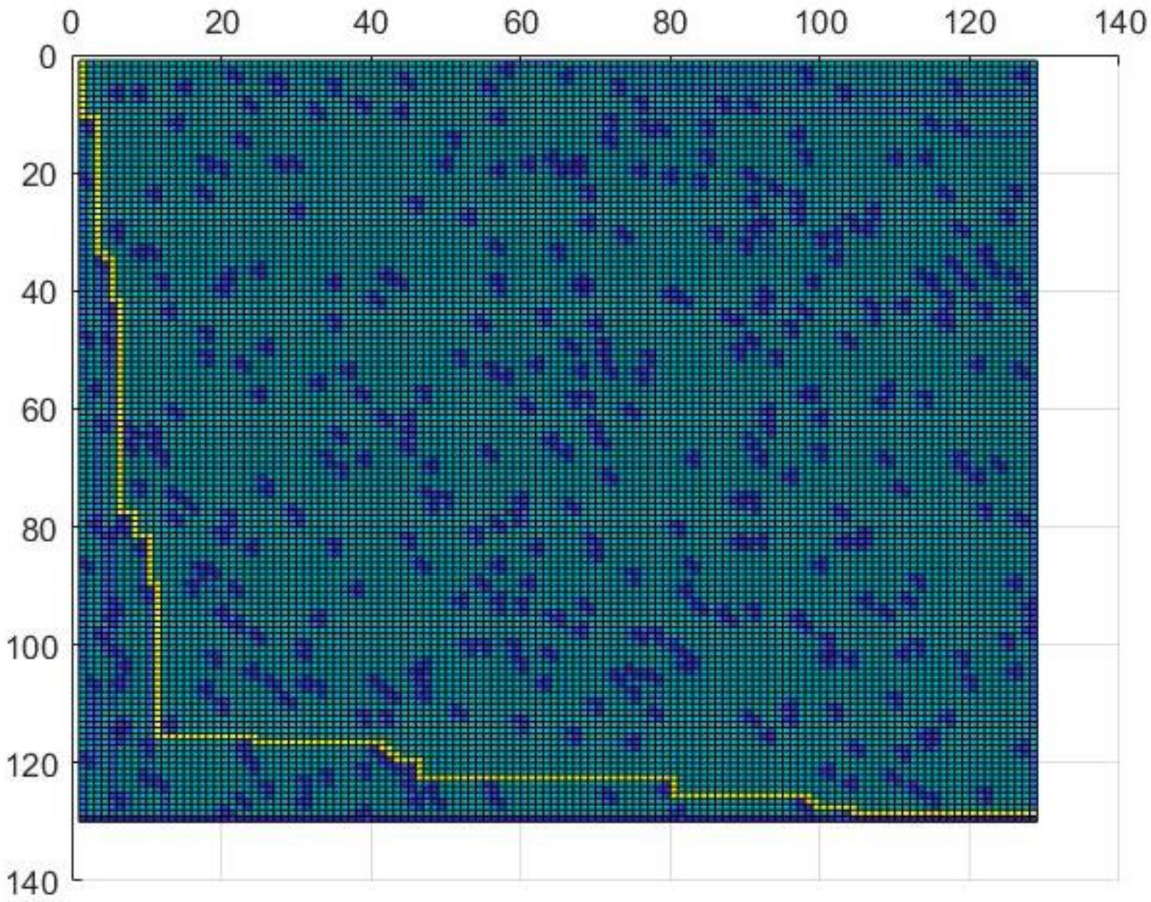


Occupancy = 50%

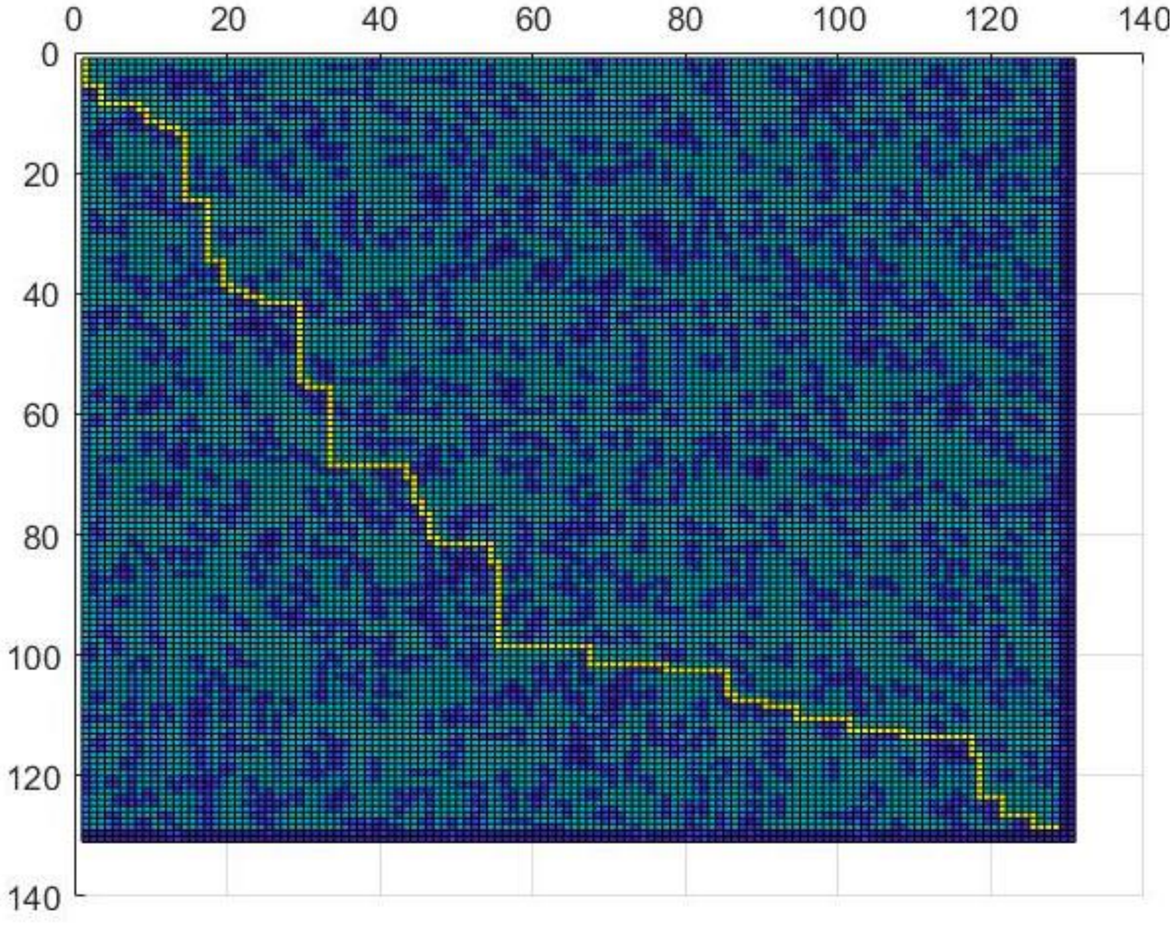


b) Breadth First Search (BFS)

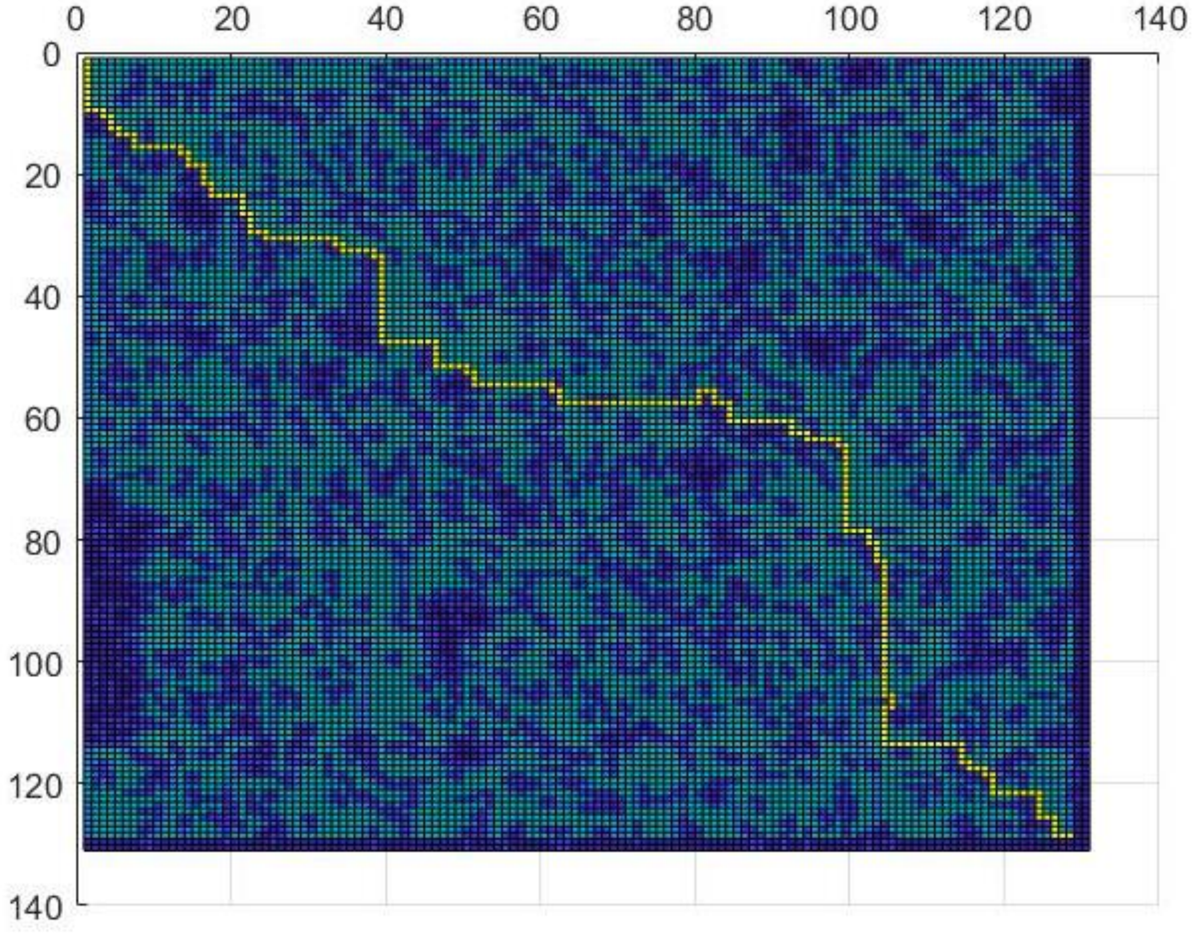
Occupancy = 10%



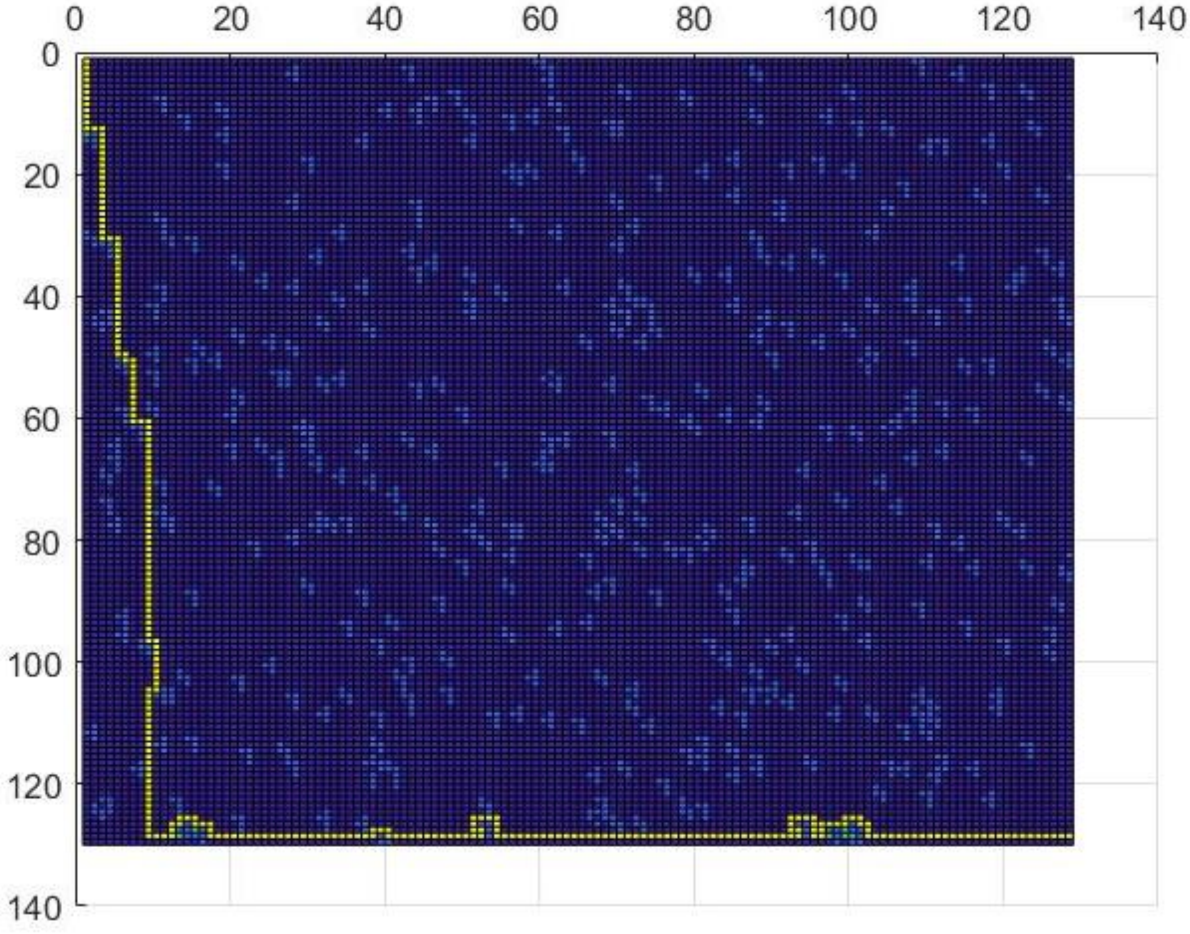
Occupancy = 30%



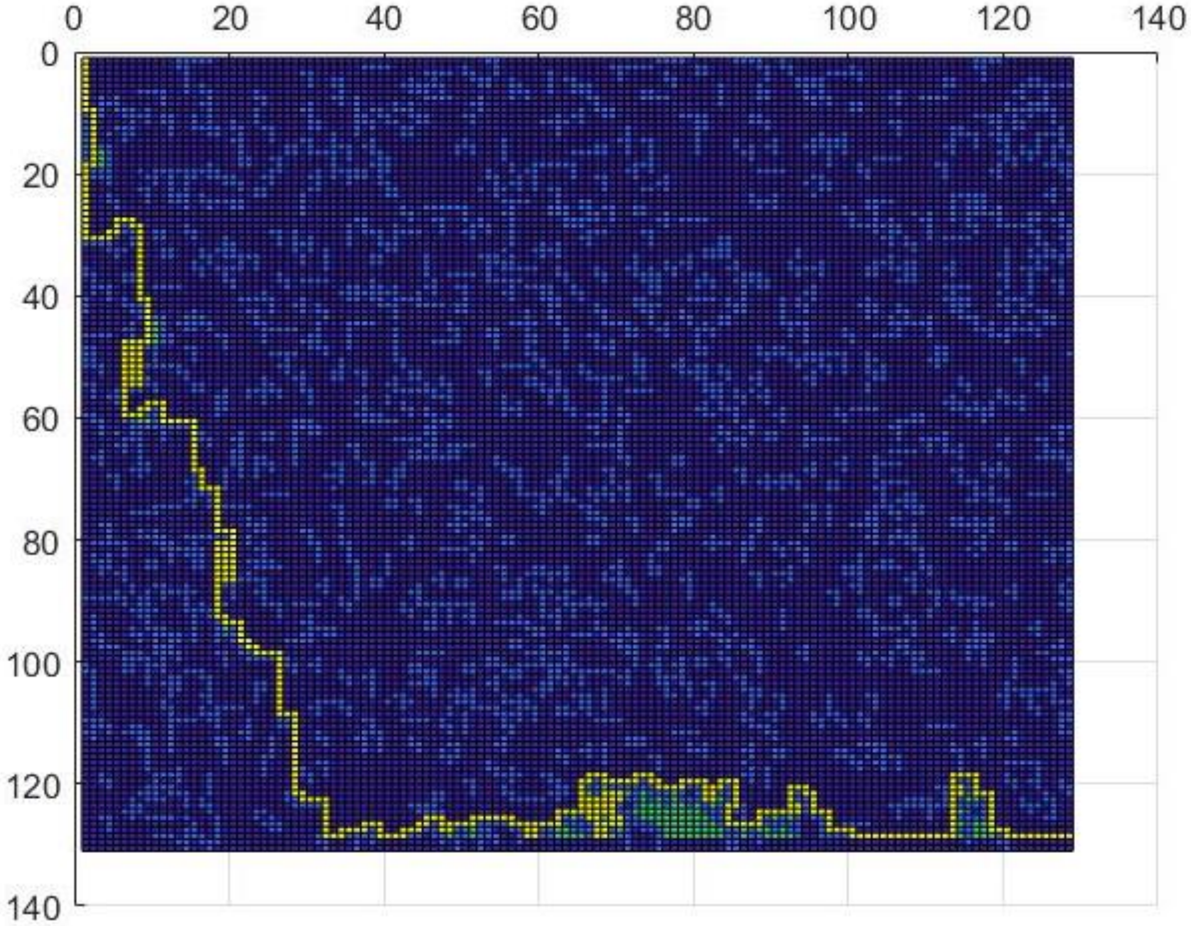
Occupancy = 40%



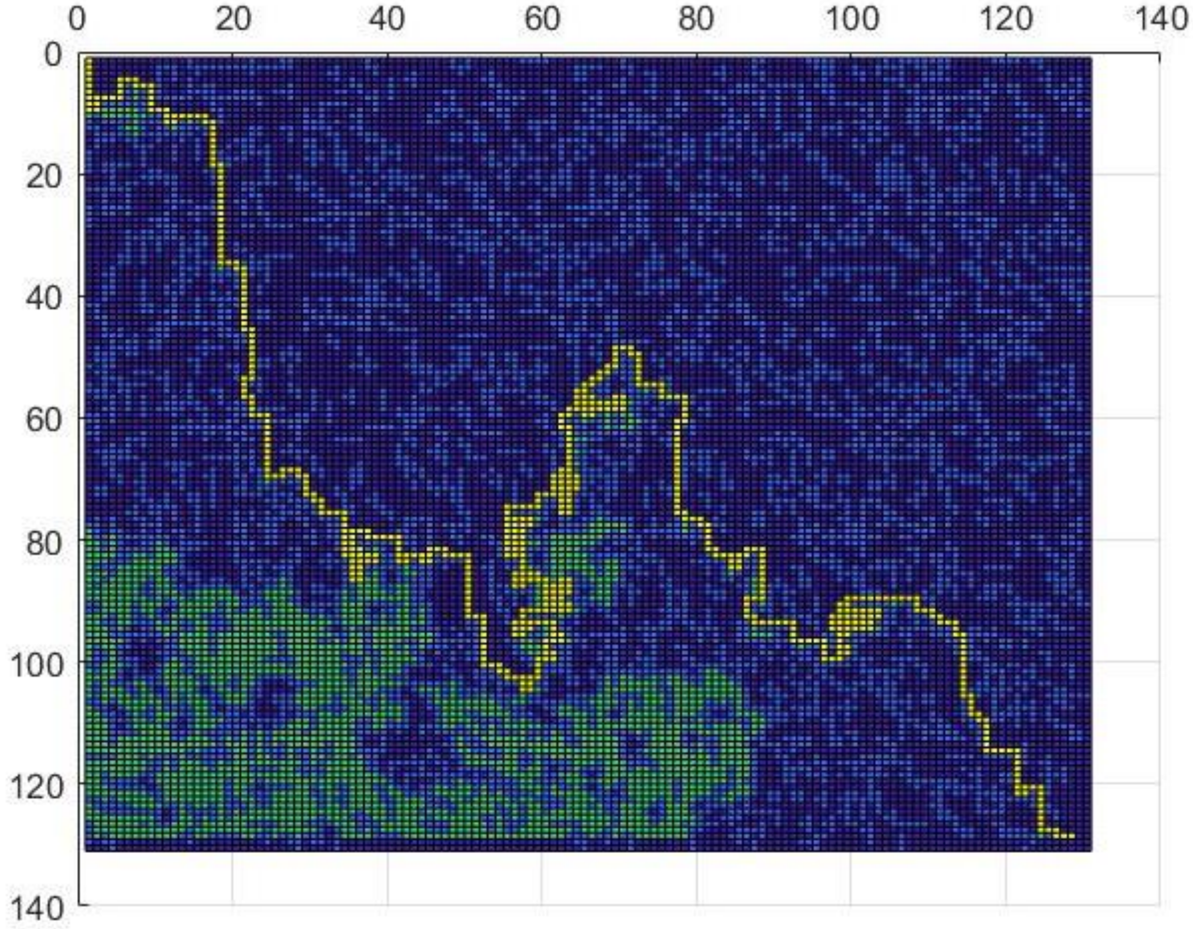
c) Depth First Search (DFS)
Occupancy = 10%



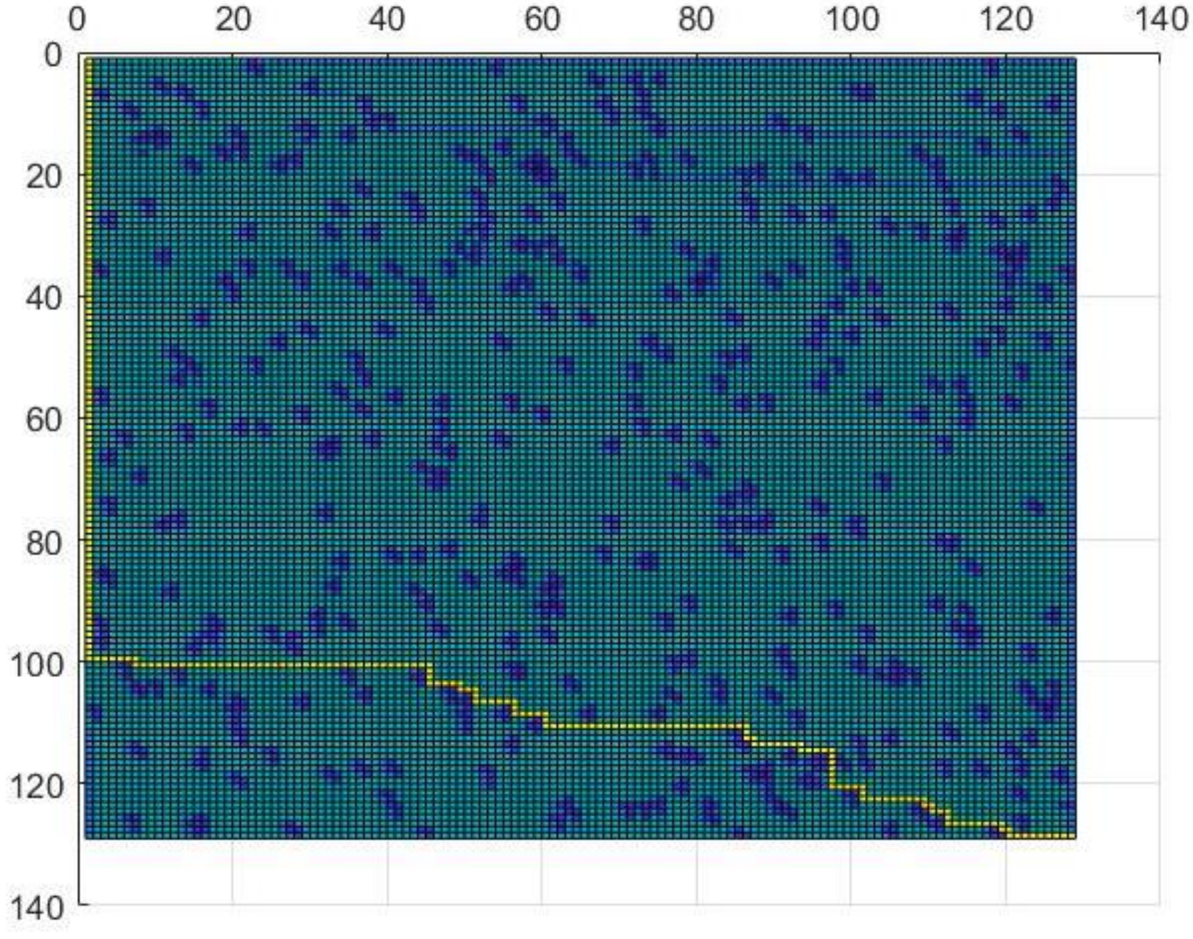
Occupancy = 30%



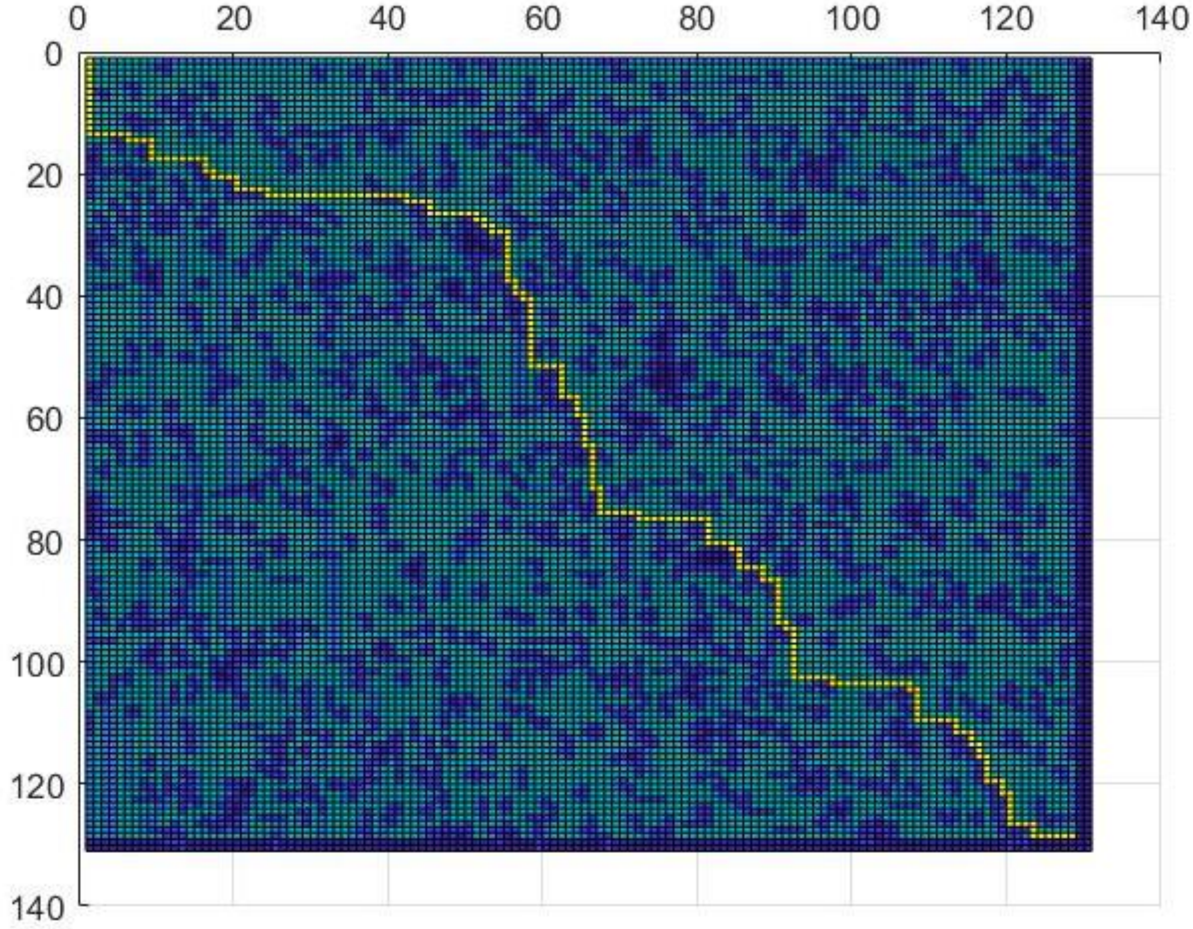
Occupancy = 50%



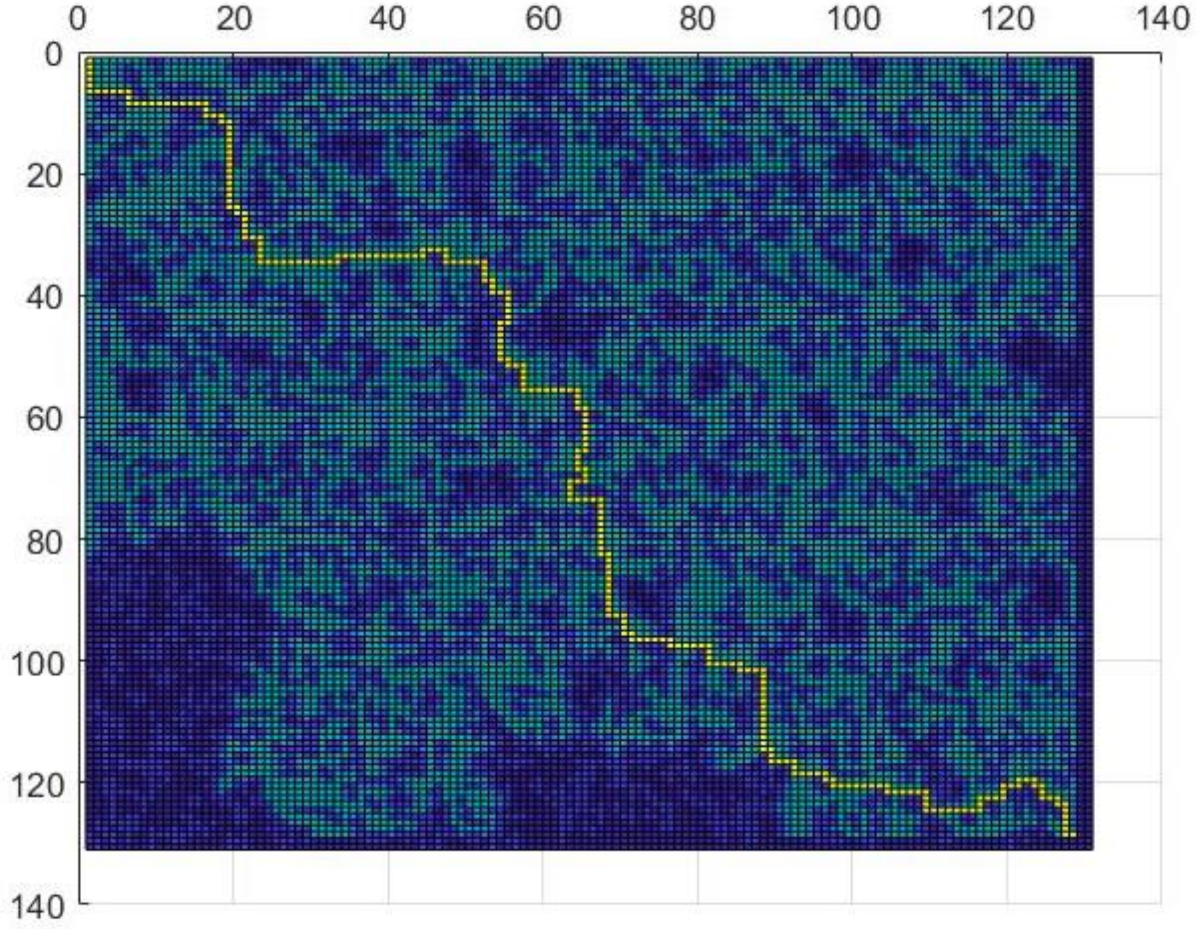
d) Dijkstra’s Algorithm
Occupancy = 10%



Occupancy = 30%



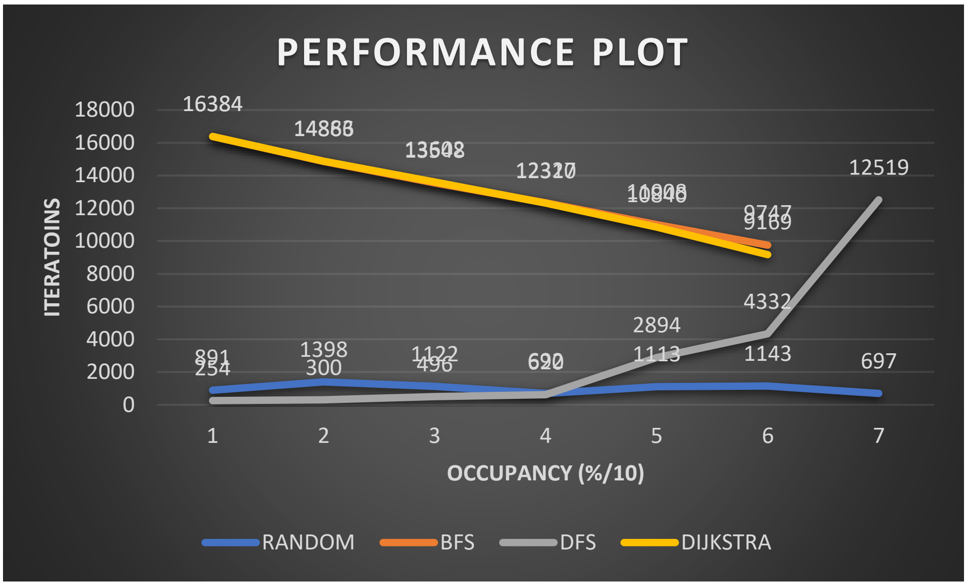
Occupancy = 50%



➢ Performance Plot:

DENSITY	**0**	**10**	**20**	**30**	**40**	**50**	**60**
RANDOM**	891	1398	1122	692	1113	1143	697
BFS	16384	14866	13548	12327	11008	9747	-
DFS	254	300	496	620	2894	4332	12519
DIJKSTRA	16384	14883	13602	12310	10840	9169	-

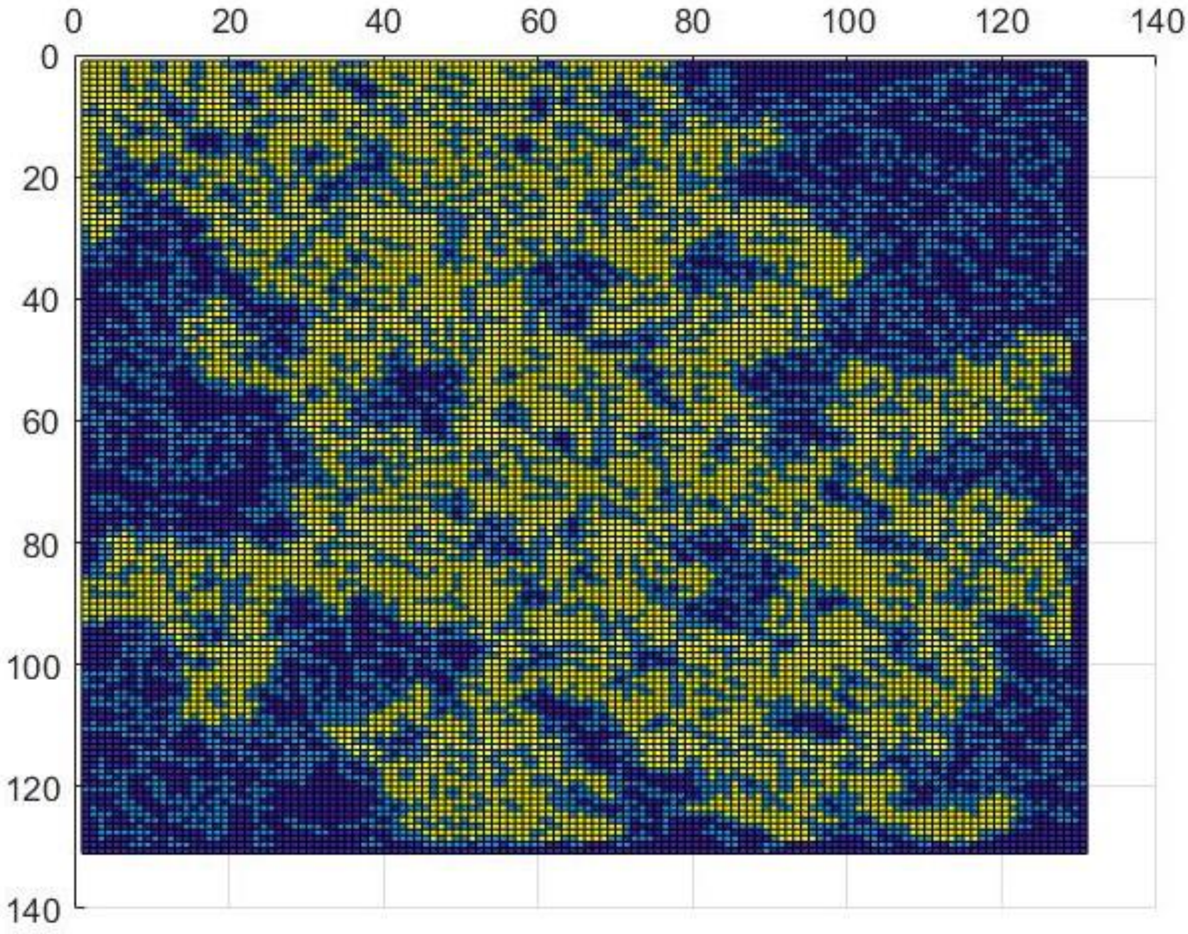
**** - Values extrapolated for (128,128)**



➢ Observations and Result:

i. Number of iterations of the Random Planner can be used as the control. It is more or less random as well.

ii. Most of the algorithms fail to detect a possible path for Occupancies above around 60%. This is because of absence of a possible path. The below figure depicts DFS failing (after about 10.5k iterations) at an occupancy of 60%.



iii. DFS performs better at lower occupancies and takes more and more iterations as the occupancy increases.

iv. BFS and Dijkstra’s algorithm performs almost equally well for all the situations. Both the algorithms take more iteration at lower occupancies and vice versa.