4)

For p in [0, p0] as above, estimate the average or expected length of the shortest path from start to goal. You may discard unsolvable maps. Plot density vs expected shortest path length. What algorithm is most useful here?

**Compare 2 different A\* algorithm:**

|  |  |  |  |
| --- | --- | --- | --- |
| Euclidean\_Distance dim=512 |  |  |  |
| p | path\_length | p | path\_length |
| 0 | 1023 | 0.09 | 1025 |
| 0.01 | 1023 | 0.1 | 1023 |
| 0.02 | 1023 | 0.12 | 1023 |
| 0.03 | 1023 | 0.13 | 1031 |
| 0.04 | 1023 | 0.14 | 1035 |
| 0.05 | 1023 | 0.15 | 1029 |
| 0.06 | 1023 | 0.16 | 1037 |
| 0.07 | 1023 | 0.17 | 1039 |
| 0.08 | 1023 | 0.18 | 1039 |

A screenshot of a social media post

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Manhattan\_Distance dim=512 |  |  |  |
| p | path\_length | p | path\_length |
| 0 | 1023 | 0.1 | 1023 |
| 0.01 | 1023 | 0.11 | 1023 |
| 0.02 | 1023 | 0.12 | 1023 |
| 0.03 | 1023 | 0.13 | 1023 |
| 0.04 | 1023 | 0.14 | 1023 |
| 0.05 | 1023 | 0.15 | 1023 |
| 0.06 | 1023 | 0.16 | 1023 |
| 0.07 | 1023 | 0.17 | 1023 |
| 0.08 | 1023 | 0.18 | 1023 |
| 0.09 | 1023 | 0.19 | 1023 |

A screenshot of a social media post

Description automatically generated

Cause Bi\_Directional BFS cost lots of time to run dim=512 we use dim=100

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bi\_Directional\_Breadth\_First\_Search dim-100 |  |  |  |  |  |
| p | meeting\_node | path\_length | p | meeting\_node | path\_length |
| 0 | (50,49) | 199 | 0.1 | (33, 66) | 199 |
| 0.01 | (45, 54) | 199 | 0.11 | (40, 59) | 199 |
| 0.02 | (46, 53) | 199 | 0.12 | (72, 27) | 199 |
| 0.03 | (49, 50) | 199 | 0.13 | (74, 25) | 199 |
| 0.04 | (46, 53) | 199 | 0.14 | (61, 38) | 199 |
| 0.05 | (45, 54) | 199 | 0.15 | (72, 27) | 199 |
| 0.06 | (37, 62) | 199 | 0.16 | (24, 75) | 199 |
| 0.07 | (20, 79) | 199 | 0.17 | (42, 58) | 199 |
| 0.08 | (79, 20) | 199 | 0.18 | (29, 71) | 199 |
| 0.09 | (44, 55) | 199 | 0.19 | (60, 38) | 199 |

A screenshot of a social media post

Description automatically generated

**Above We can conclude that BiDirectional BFS is time-wasting and A\* algorithm perform better**

5)

Is one heuristic uniformly better than the other for running A∗ ? How can they be compared? Plot the relevant data and justify your conclusions.

I develop UniformSearch Algorithm, supposing heuristic is 0 and compare with A\* algorithm

**compare three different algorithm time cost:**

set dim=512 and p is ranged from 0 to 0.6 with step 0.05:

total time cost using Manhattan search is 8.99621868133545

total time cost using Euclidean search is 9.80755090713501

total time cost using Uniform search is 11.38677978515625

set dim=512 and p is ranged from 0 to 0.3 with step 0.025:

total time cost using Manhattan search is 12.999441146850586

total time cost using Euclidean search is 12.308625221252441

total time cost using Uniform search is 16.55923104286194

set dim=512 and p is ranged from 0 to 0.2 with step 0.02:

total time cost using Manhattan search is 12.800921201705933

total time cost using Euclidean search is 14.627891063690186

total time cost using Uniform search is 16.26413106918335

Dijkstra is a special case for A\* (when the heuristics is zero)

From the data we can conclude that uniformsearch(heuristic uniformly) spend more time than A\* algorithm

6) Do these algorithms behave as they should?

Yes BdBFS cost lots of time and A\* algorithms perform better

8) On the same map, are there ever nodes that BD-DFS expands that A∗ doesn’t? Why or why not? Give an example, and justify.

Yes Different algorithm may generate different path and the nodes visited is different.

Using copy.deepcopy library to duplicate 3 same mazes and use 3 algorithms to calculate the sum of visited node

Here the snapshot of result:

Dim=256 p=0.2 (Here 100 represent path)

Original Path

A black sign with white text

Description automatically generated

For Mahattam A\*

A close up of a keyboard

Description automatically generated

For Euclidean A\*

A close up of a keyboard

Description automatically generated

For Bd-BFS:

A picture containing text

Description automatically generated