Prithvi_Poddar_17191_astar_dstar

March 20, 2021

1 Report on A* and D* search algorithms

2 Prithvi Poddar (17191)

2.1 A* implementation

In the outputs of the A* algorithm, **grey** nodes will be the nodes that were explored by the algorithm and the **black** line will be the shortest path from the Start (**marked by an x**) to the Goal (**marked by a solid black circle**)

2.1.1 5 obstacles environment

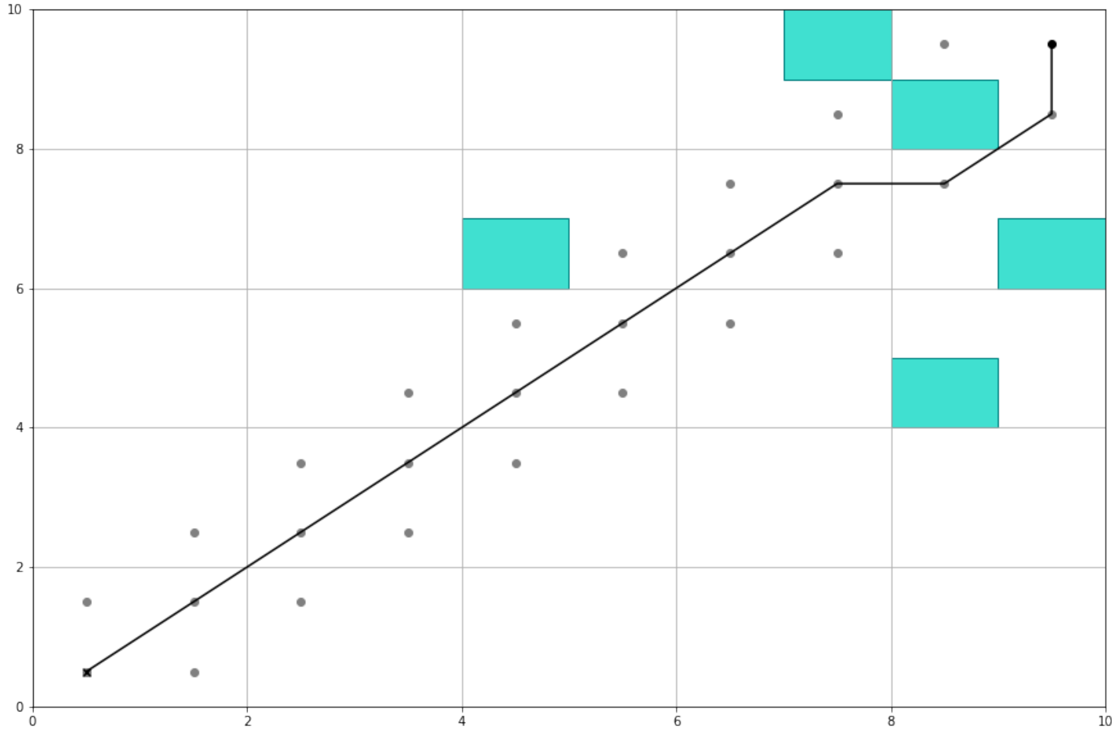
[1]:
```
from astar import *
```

[20]:
```
n_row=10
n_col=10
n_obs=5
start=[9,0]
goal=[0,9]
n_obs = 5
astar = A_star(n_row=n_row, n_col=n_col, n_obs=n_obs, start=start, goal=goal)
```

[21]:
```
astar.generate_path()
```

```
Path found
```

[22]:
```
astar.visualize()
```



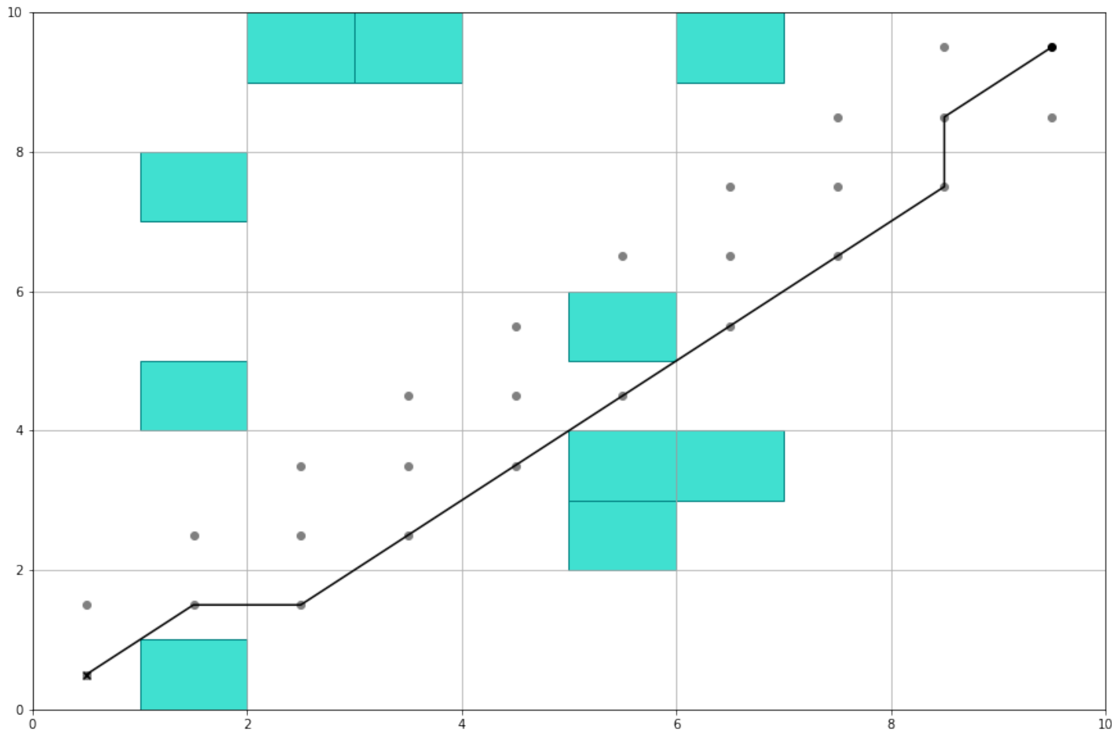
2.1.2 10 obstacle environment

```
[29]: n_row=10
n_col=10
start=[9,0]
goal=[0,9]
n_obs = 10
astar = A_star(n_row=n_row, n_col=n_col, n_obs=n_obs, start=start, goal=goal)
```

```
[30]: astar.generate_path()
```

```
Path found
```

```
[31]: astar.visualize()
```



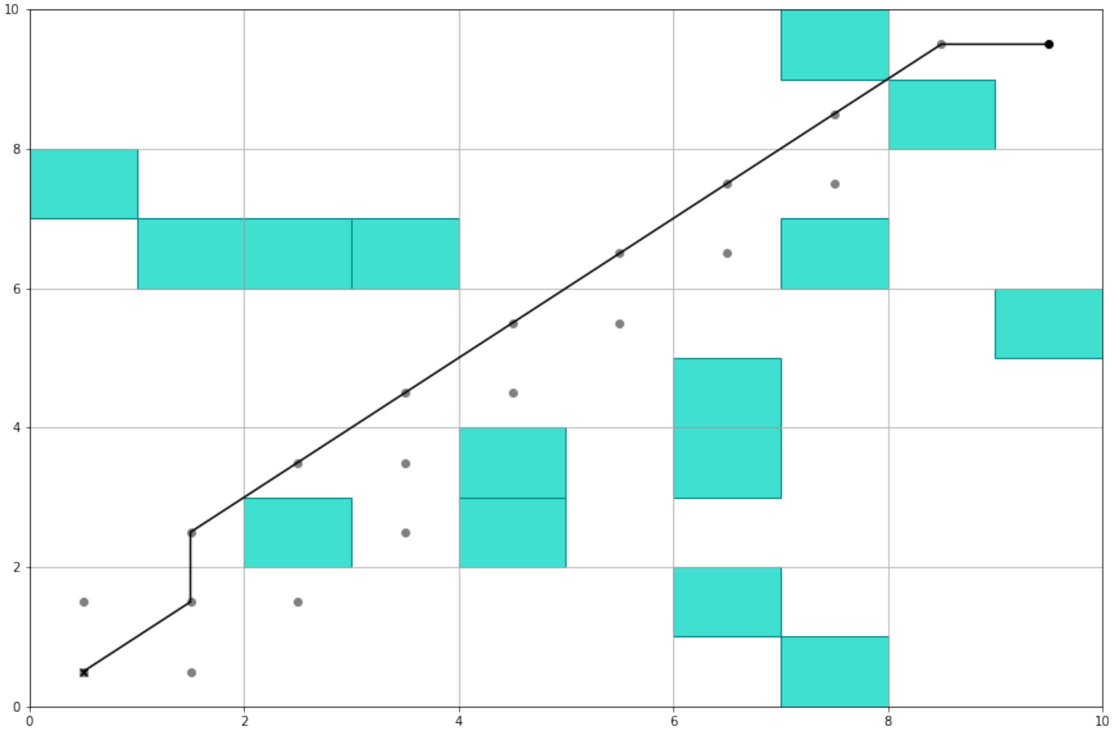
2.1.3 15 obstacle environment

[32]:
```
n_row=10
n_col=10
start=[9,0]
goal=[0,9]
n_obs = 15
astar = A_star(n_row=n_row, n_col=n_col, n_obs=n_obs, start=start, goal=goal)
```

[33]:
```
astar.generate_path()
```

```
Path found
```

[34]:
```
astar.visualize()
```



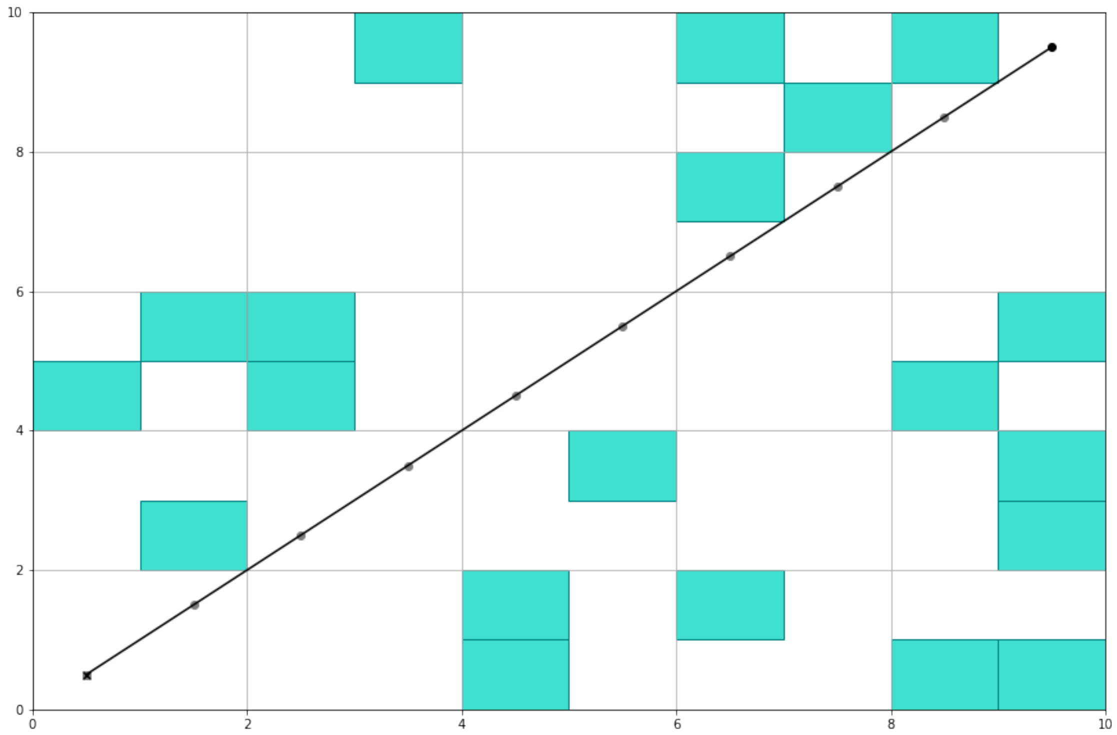
2.1.4 20 obstacle environment

[35]:
```
n_row=10
n_col=10
start=[9,0]
goal=[0,9]
n_obs = 20
astar = A_star(n_row=n_row, n_col=n_col, n_obs=n_obs, start=start, goal=goal)
```

[36]:
```
astar.generate_path()
```

```
Path found
```

[37]:
```
astar.visualize()
```



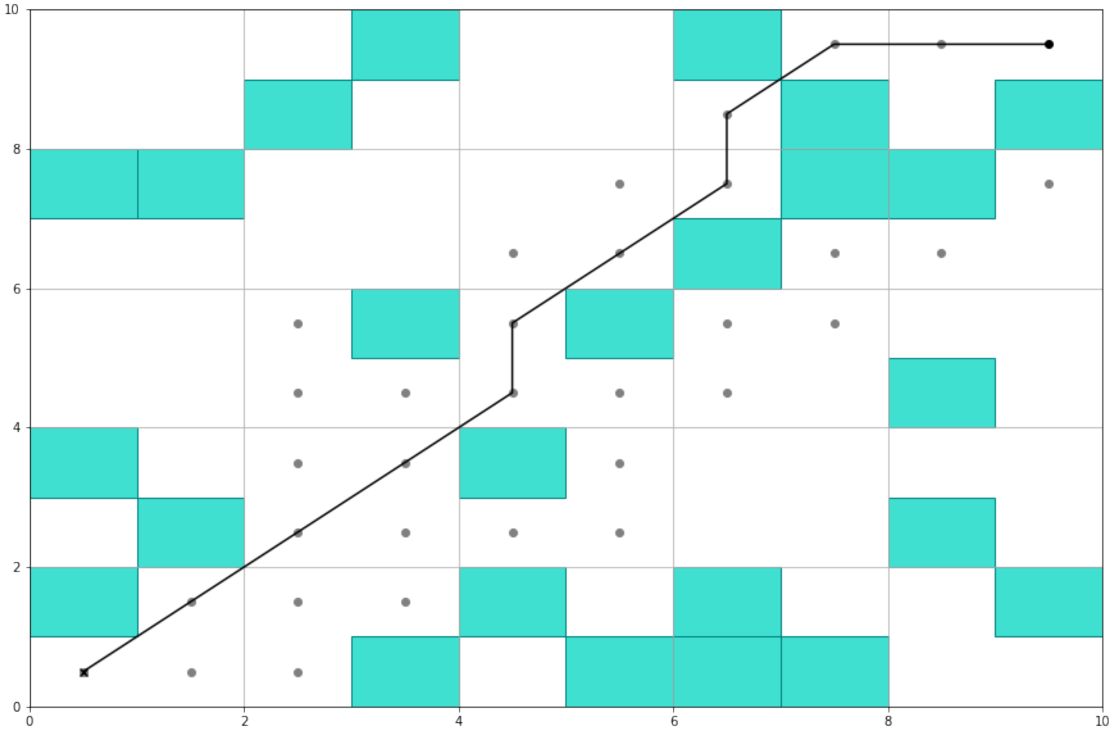
2.1.5 25 obstacle environment

[40]:
```
n_row=10
n_col=10
start=[9,0]
goal=[0,9]
n_obs = 25
astar = A_star(n_row=n_row, n_col=n_col, n_obs=n_obs, start=start, goal=goal)
```

[41]:
```
astar.generate_path()
```

```
Path found
```

[42]:
```
astar.visualize()
```



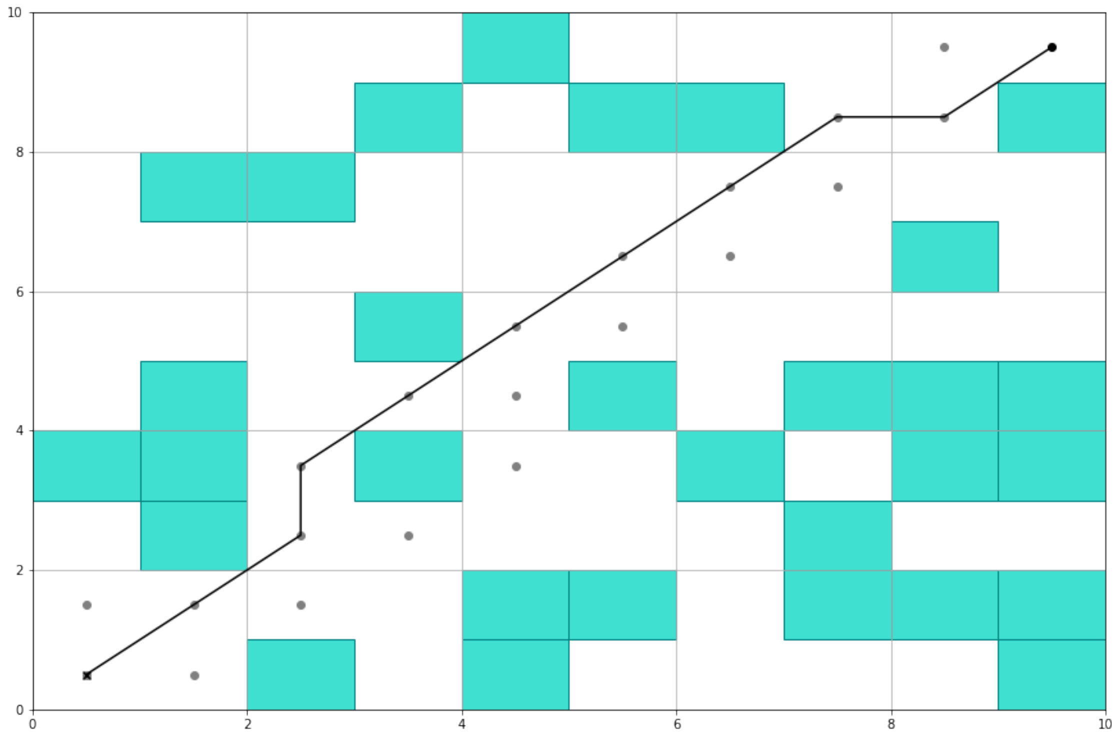
2.1.6 30 obstacle environment

[43]:
```
n_row=10
n_col=10
start=[9,0]
goal=[0,9]
n_obs = 30
astar = A_star(n_row=n_row, n_col=n_col, n_obs=n_obs, start=start, goal=goal)
```

[44]:
```
astar.generate_path()
```

```
Path found
```

[45]:
```
astar.visualize()
```



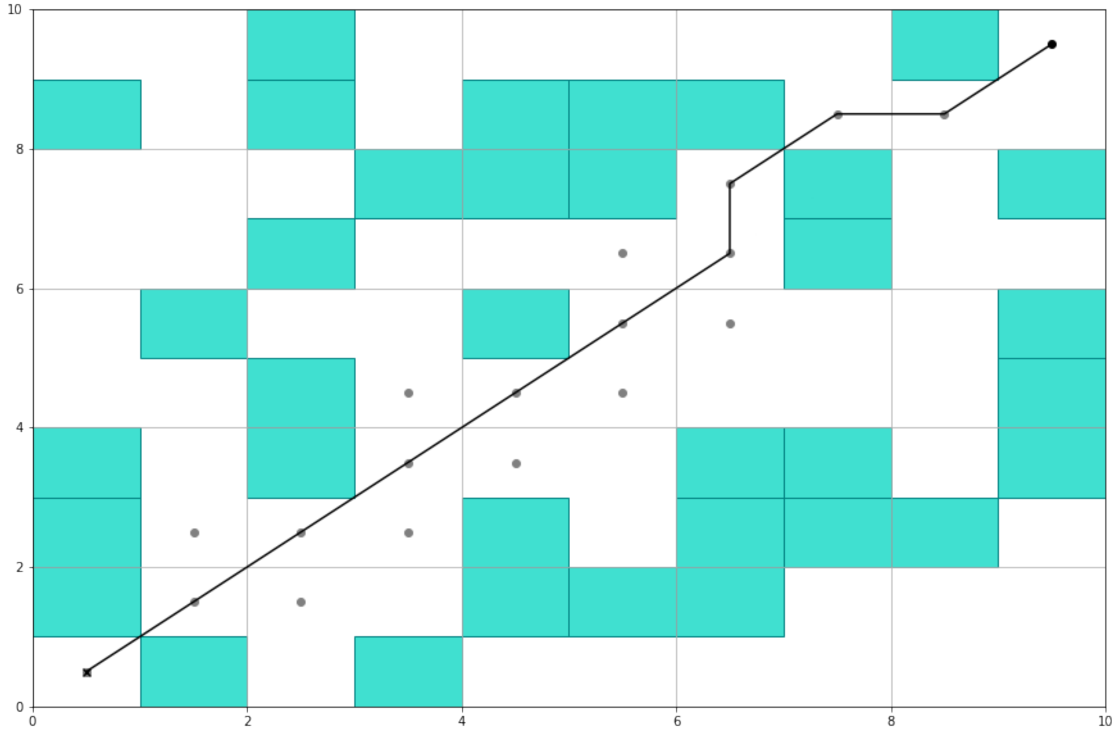
2.1.7 35 obstacle environment

[46]:
```
n_row=10
n_col=10
start=[9,0]
goal=[0,9]
n_obs = 35
astar = A_star(n_row=n_row, n_col=n_col, n_obs=n_obs, start=start, goal=goal)
```

[47]:
```
astar.generate_path()
```

```
Path found
```

[48]:
```
astar.visualize()
```



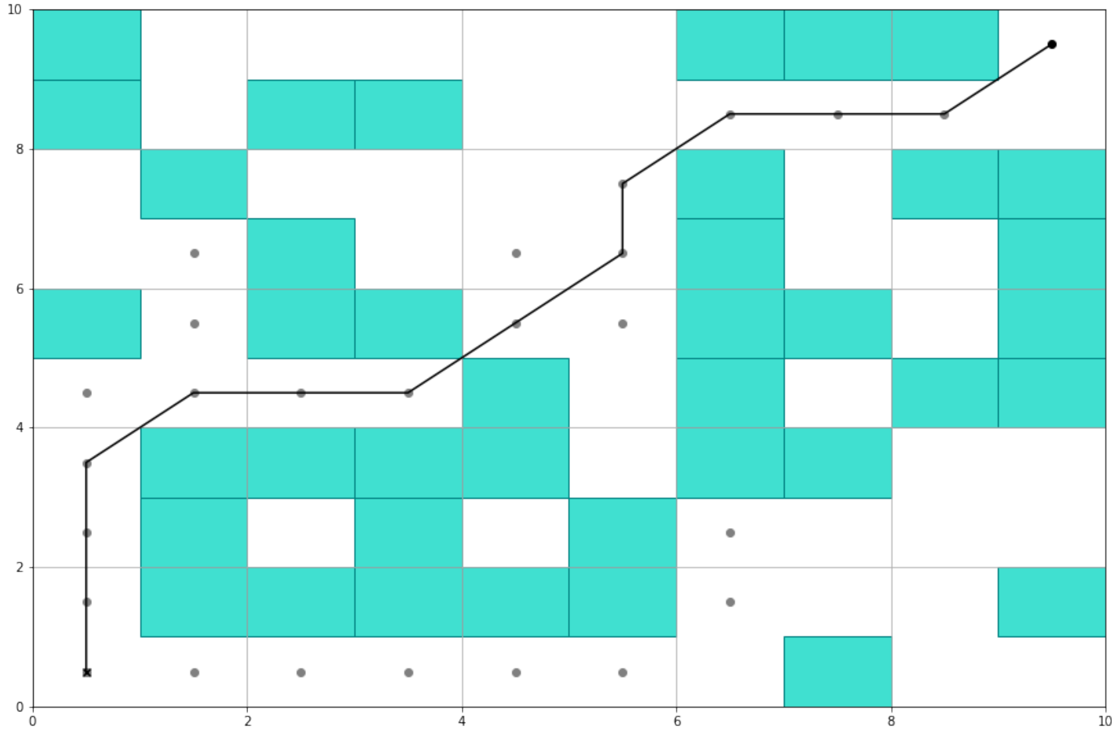
2.1.8 40 obstacle environment

[49]:
```
n_row=10
n_col=10
start=[9,0]
goal=[0,9]
n_obs = 40
astar = A_star(n_row=n_row, n_col=n_col, n_obs=n_obs, start=start, goal=goal)
```

[50]:
```
astar.generate_path()
```

```
Path found
```

[51]:
```
astar.visualize()
```



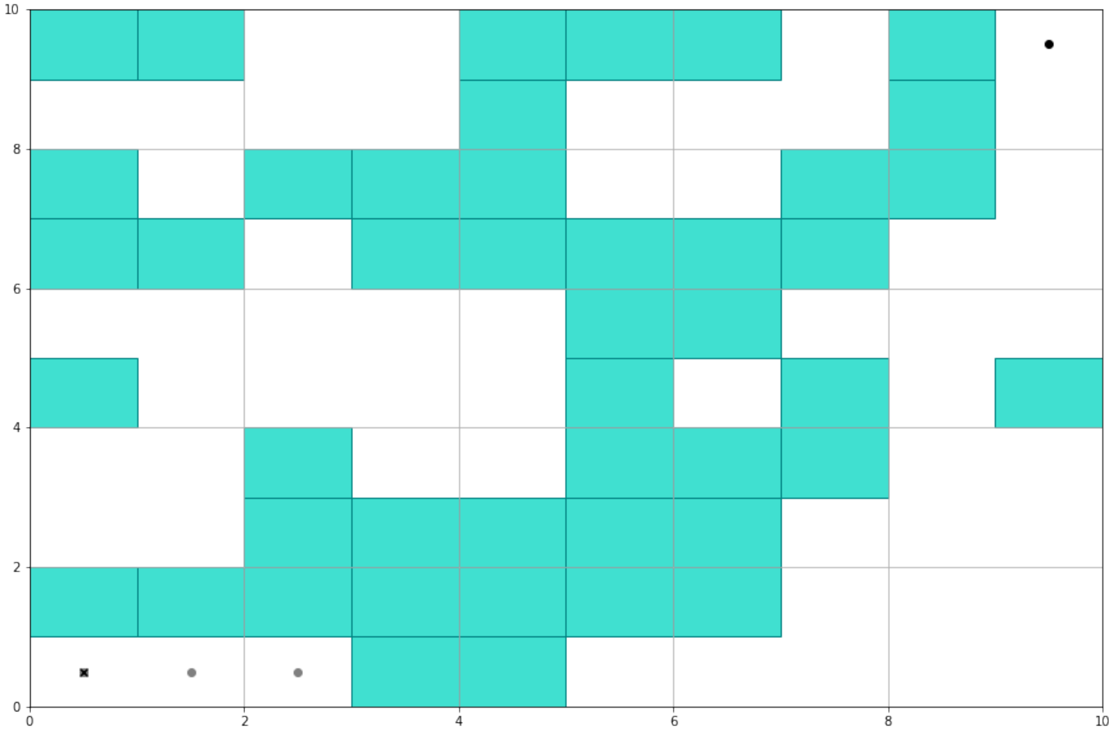
2.1.9 45 obstacle environment

```
[58]: n_row=10
      n_col=10
      start=[9,0]
      goal=[0,9]
      n_obs = 45
      astar = A_star(n_row=n_row, n_col=n_col, n_obs=n_obs, start=start, goal=goal)
```

```
[59]: astar.generate_path()
```

```
Goal is not reachable
```

```
[60]: astar.visualize()
```



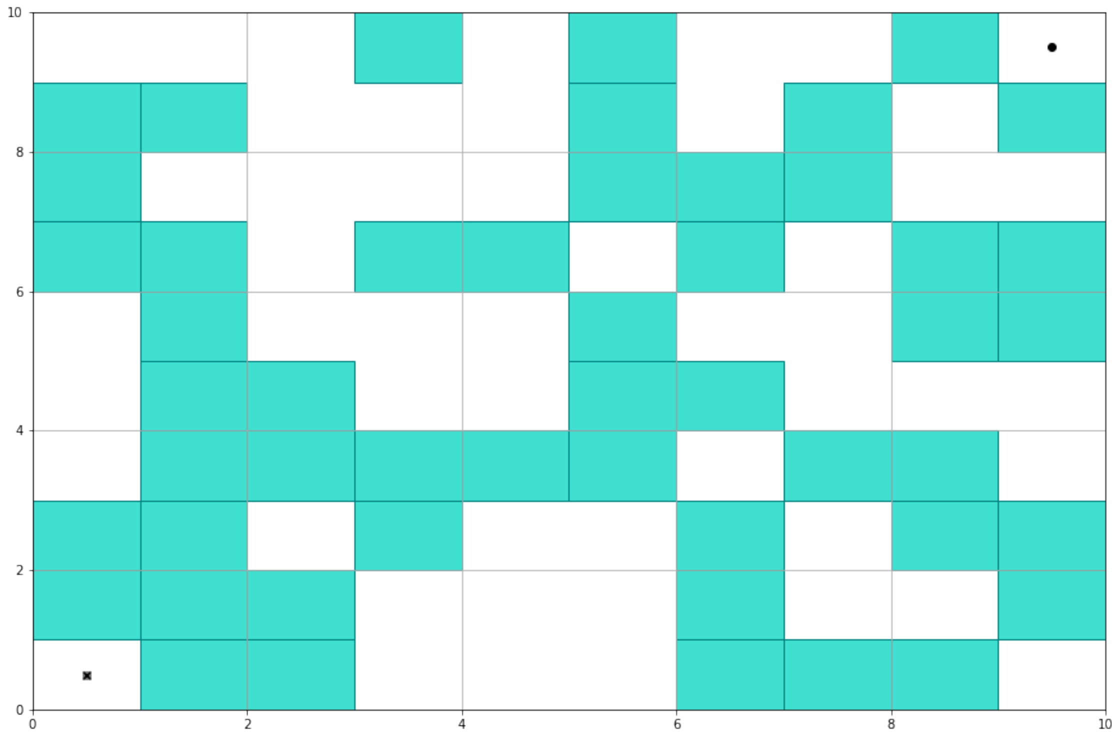
2.1.10 50 obstacle environment

[55]:
```
n_row=10
n_col=10
start=[9,0]
goal=[0,9]
n_obs = 50
astar = A_star(n_row=n_row, n_col=n_col, n_obs=n_obs, start=start, goal=goal)
```

[56]:
```
astar.generate_path()
```

```
Goal is not reachable
```

[57]:
```
astar.visualize()
```



Hence with increasing obstacles we see that there is a chance that the goal might become unreachable in which case the algorithm is able to tell us if the goal is unreachable.

```
[ ]:
```