Advanced Search Algorithm Implementation

-Sailesh Rajagopalan

D*

Explain in your own words, how does D* replan a path by updating the Cost?

D* follows dijkstra to calculate the path but unlike that it allows for adding the nodel in a list from the goal to start, nodes can be represented in and with respect to D* the replan follows as process state, prepare repair, repair replan, The replan is as:

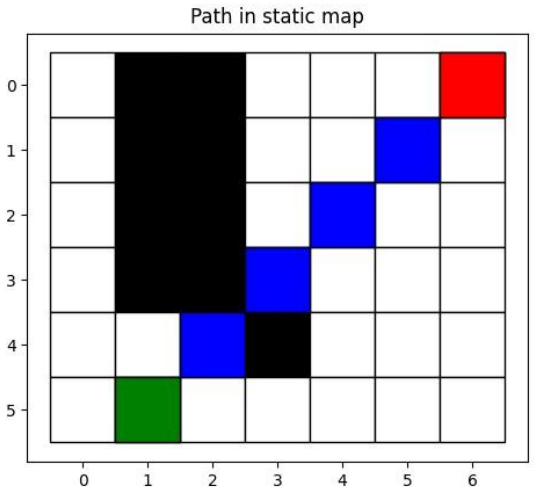
- The heuristic of the goal is set to zero
- The Goal node will be added to the open and by leveraging algorithms like A* or Dijkstra we can find the feasible path
- Using the path which is found, and using sensors we have the robot can traverse, every state that exists the nodes are repaired and each of the affected nodes are evaluated
- This is a repetitive process which continues till the robot is able to reach the goal

Why does D* can replan faster than A* or Dijkstra?

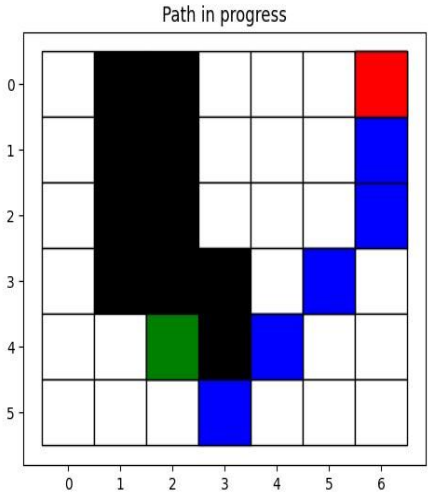
D* follows a methodical repair and replan with enables the path locally, by taking into account the previous path calculations in the map and it changes it based on that making it less computationally expensive, A* and Dijkstra's shortcomings are in the phase as they repeat the process from base 1, as they calculate from the first it is very expensive computationally.

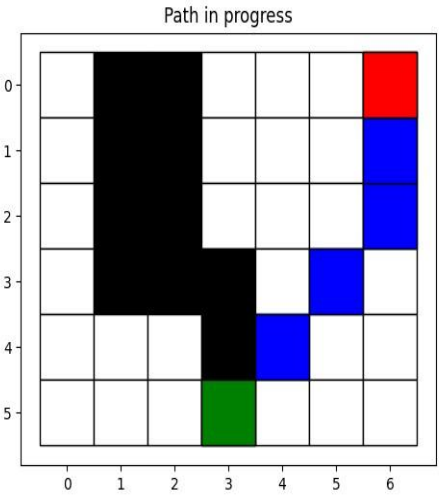
RESULT:

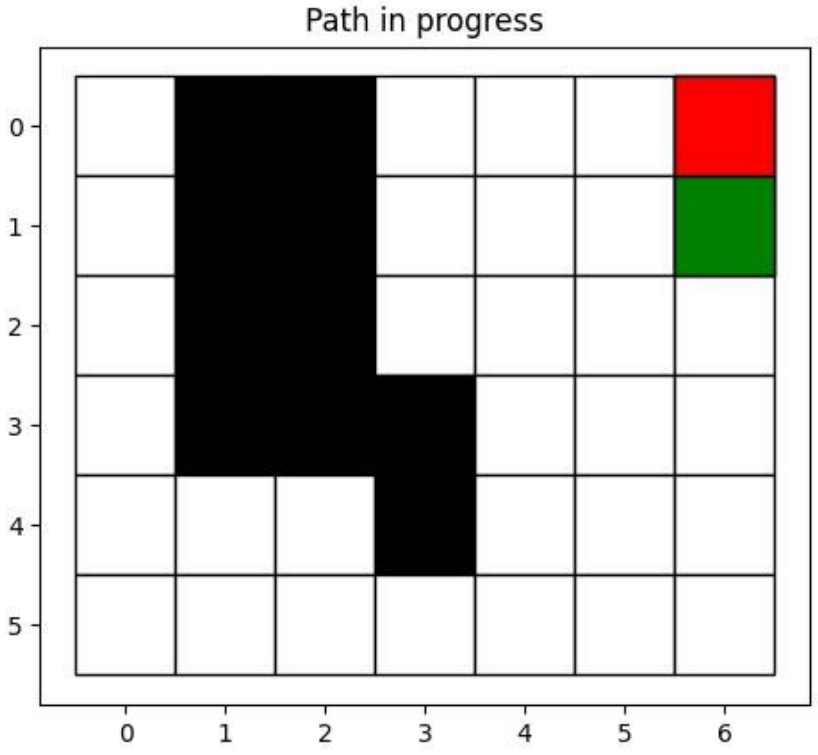
Map 1:



Dynamic Obstacle:

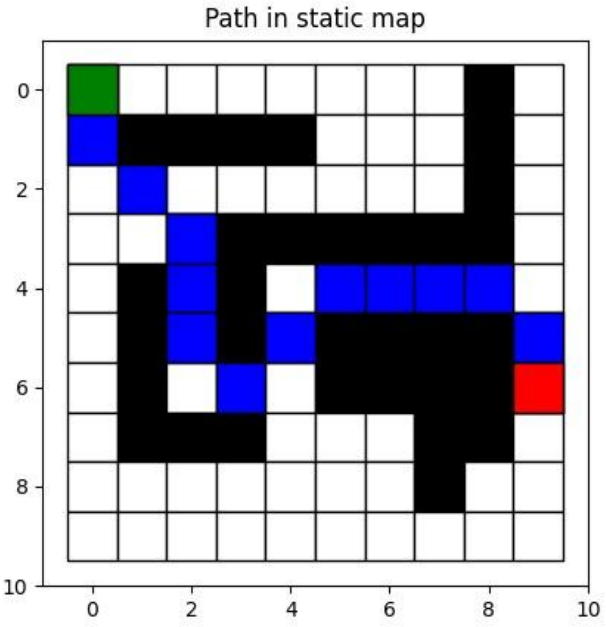

Path in progress
0
1
2
3
4
5
0
1
2
3
4
5
6


Path in progress
0
1
2
3
4
5
0
1
2
3
4
5
6

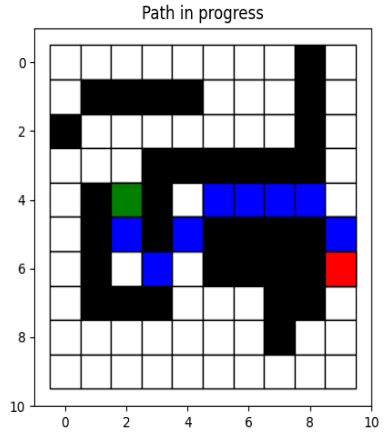

Path in progress
0
1
2
3
4
5
0
1
2
3
4
5
6

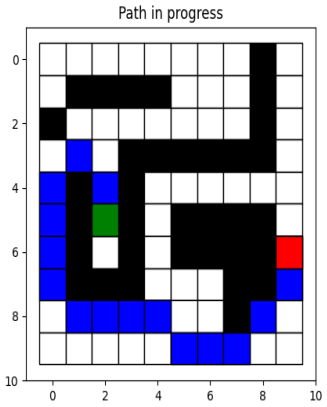
Goal

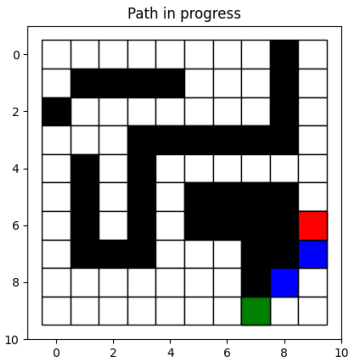
Map2:

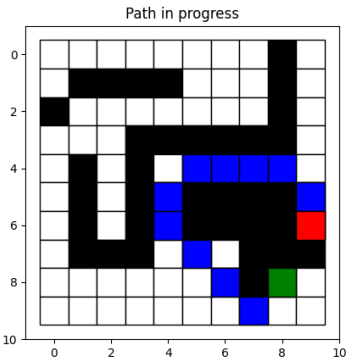


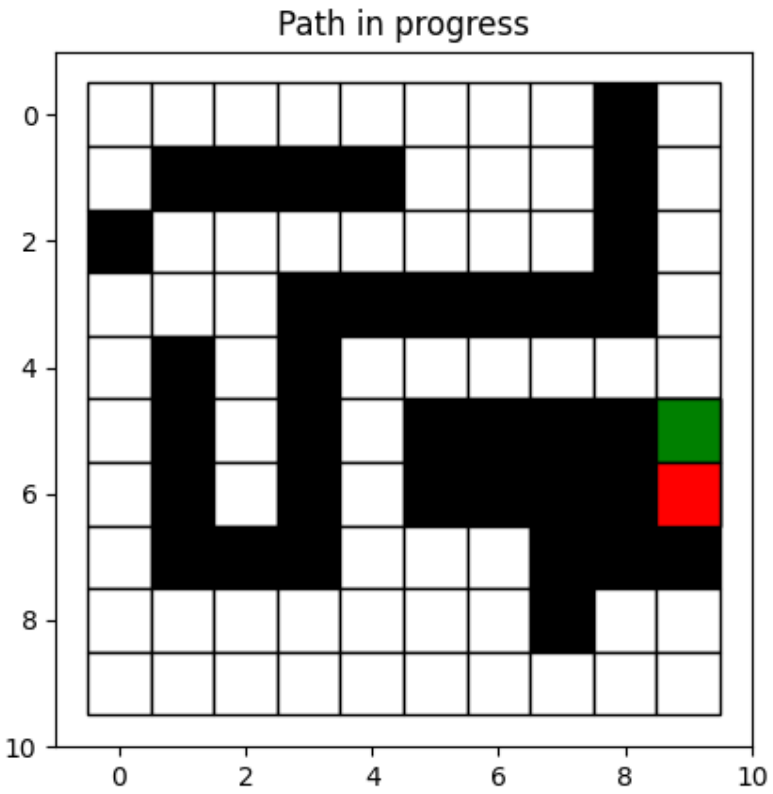
Dynamic Obstacle





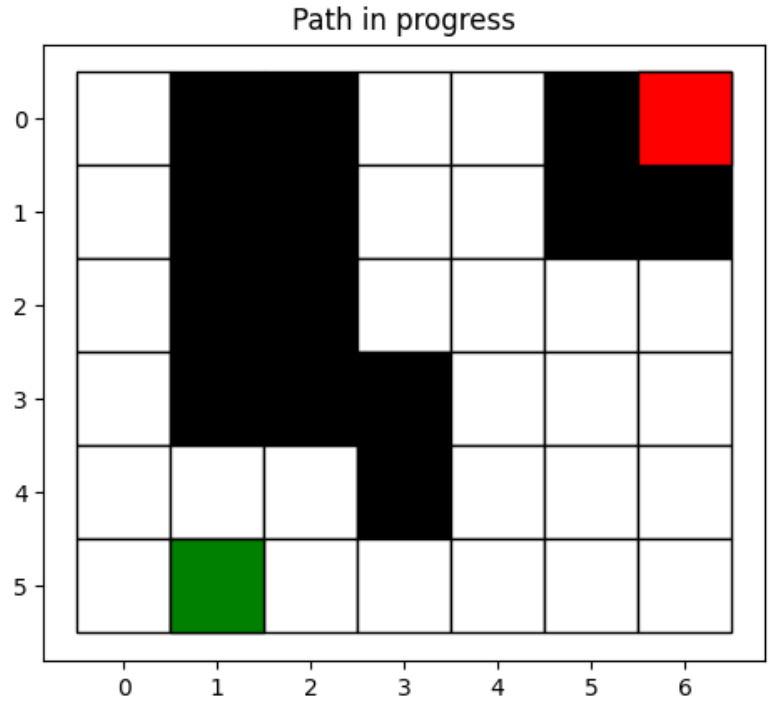





Path in progress
0
2
4
6
8
10
0
2
4
6
8
10

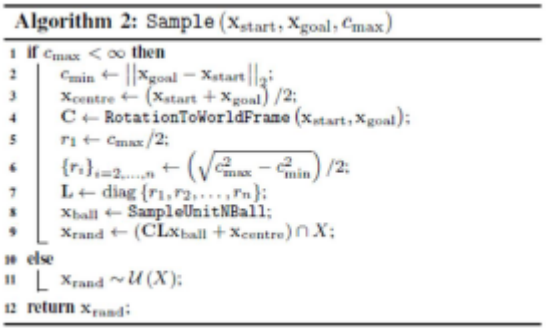
Goal

MAP 3:


Path in progress
0
1
2
3
4
5
0
1
2
3
4
5
6

No path is found

INFORMED RRT*



Algorithm 2: Sample $(x_{start}, x_{goal}, c_{max})$

1 **if** $c_{max} < \infty$ **then**
2 $c_{min} \leftarrow ||x_{goal} - x_{start}||_2$;
3 $x_{centre} \leftarrow (x_{start} + x_{goal})/2$;
4 $C \leftarrow$ RotationToWorldFrame (x_{start}, x_{goal});
5 $r_1 \leftarrow c_{max}/2$;
6 $\{r_i\}_{i=2,\dots,n} \leftarrow \left(\sqrt{c_{max}^2 - c_{min}^2}\right)/2$;
7 $L \leftarrow \text{diag}\{r_1, r_2, \dots, r_n\}$;
8 $x_{ball} \leftarrow$ SampleUnitNBall;
9 $x_{rand} \leftarrow (CLx_{ball} + x_{centre}) \cap X$;
10 **else**
11 $x_{rand} \sim \mathcal{U}(X)$;
12 **return** x_{rand};

What is the key differences between regular RRT* and informed RRT*?

Informed RRT* - The path exploration is controlled in a finite space, the tree structure that is generated only requires small number of nodes and provides an optimal path in a lesser number of iterations, As Informed RRT* is a focused it is done using an ellipsodal in the local region

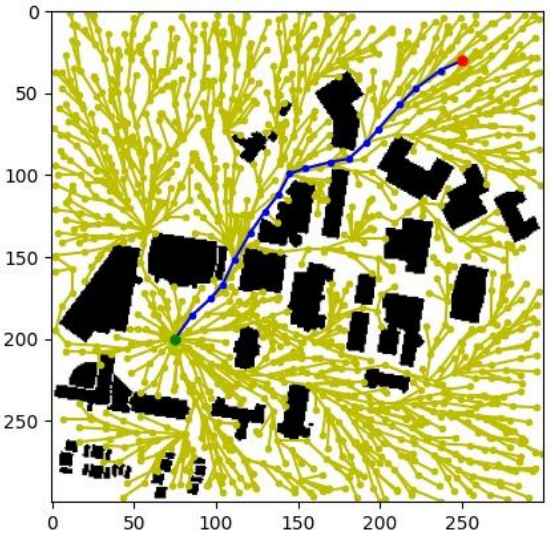
RRT* - RRT* does very high exploration on the sample space adn it required greater number of nodes compared to Informed RRT*, and unlike in the local region sampling is performed on the entire workspace.

RESULT:

```
"C:\Users\Sailesh Rajagopalan\PycharmProjects\AI\venv\Scripts\pyt
******** RRT Star *******
Iterations = 500 Cbest = 315.989308282746
Iterations = 1000 Cbest = 306.65208898687024
Iterations = 1500 Cbest = 302.98713093722125
Iterations = 2000 Cbest = 302.98713093722125
RRT star: It took 1441 nodes to find the current path
The path length is 291.34
*********Informed RRT Star*******
Iterations = 0 Cbest = 0
Iterations = 500 Cbest = 303.5944954333836
Iterations = 1000 Cbest = 285.8255119607477
Iterations = 1500 Cbest = 276.126086567143
Iterations = 2000 Cbest = 273.40950885958097
Informed RRT star:It took 1113 nodes to find the current path
The path length is 273.41
```

Hence we can conclude, that Informed RRT* will provide an optimal feasible path compared to other, as we have discussed above their differences, As we can see for 1000 iteration the Cbest = 306.7 for RRT* and 285.8 for Informed RRT*, hence providing a faster and low computationally expensive path with less nodes

RRT*



INFORMED RRT*