Manoj Kumar Selvaraj
UID: 120511257

1.

BUG 1 Algorithm:

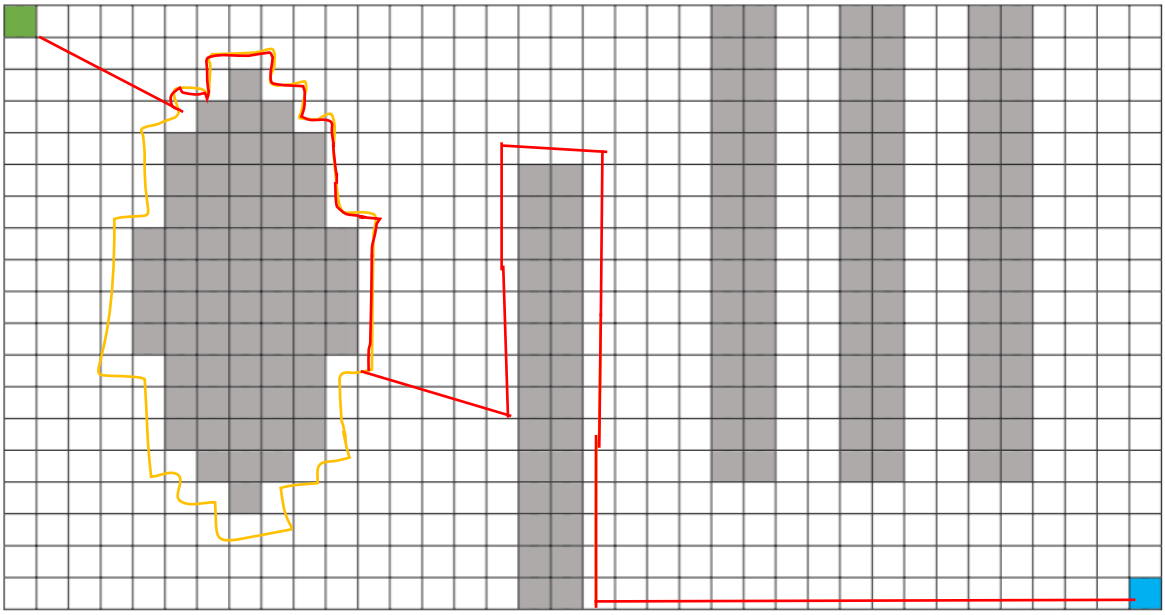
Steps:

1. Head towards the goal.
2. If the obstacle is in the way circumnavigate it and remember the closest point to the goal.
3. Go to that closest point by following the obstacle.
4. Continue the steps till goal is reached.

Legend:

Green point: Start
Blue point : Goal
Grey : Obstacles
White : Free space
Orange : Actual distance travelled
Red : Final Path

Here, I am considering a clockwise turn(left)
Once the robot follows the 1st rectangle, goal point is reached. So circumnavigating the obstacle is not completed.



BUG 2 Algorithm:

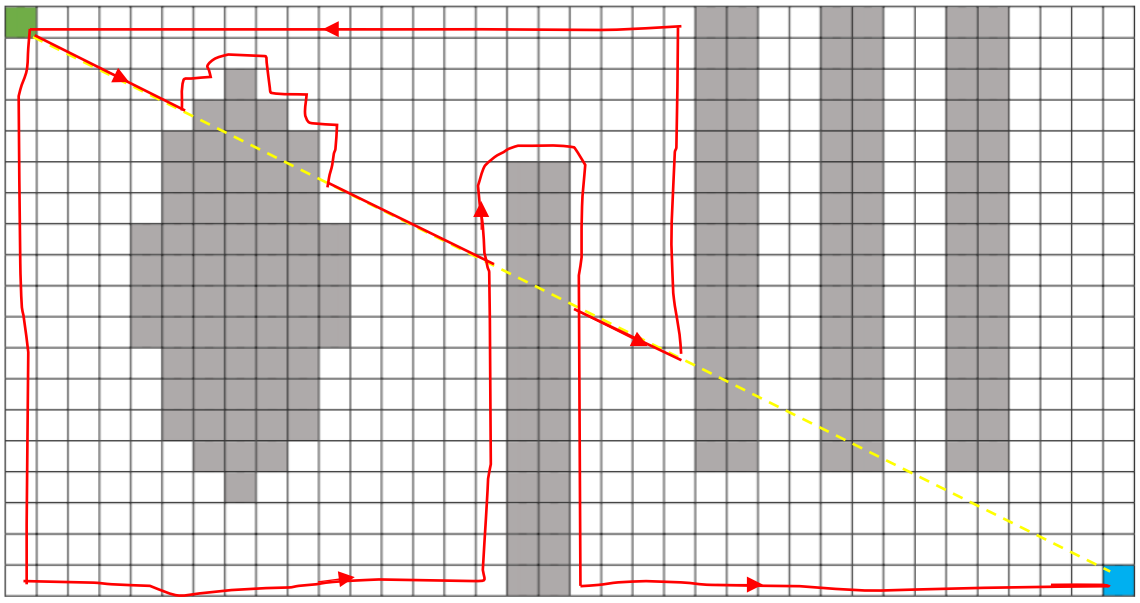
Steps:

1. Head towards goal on m-line.
2. If the obstacle is in way , follow the obstacle until you see the m-line again which is closer to the goal point.
3. Leave the obstacle and go towards the goal.

Legend:

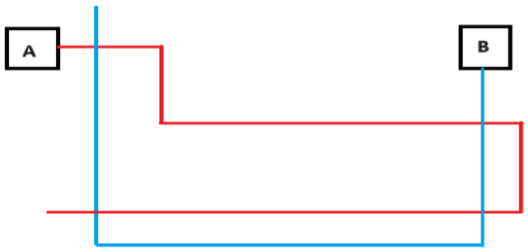
Green point : Start
Blue point : Goal
Grey : obstacles
White : Free space
Yellow : m-line
Red : Final Path

Here initially the robot follows the m-line, once obstacle is encountered it moves in clockwise(left) direction till m-line is met. Once the goal point is reached final path is found.



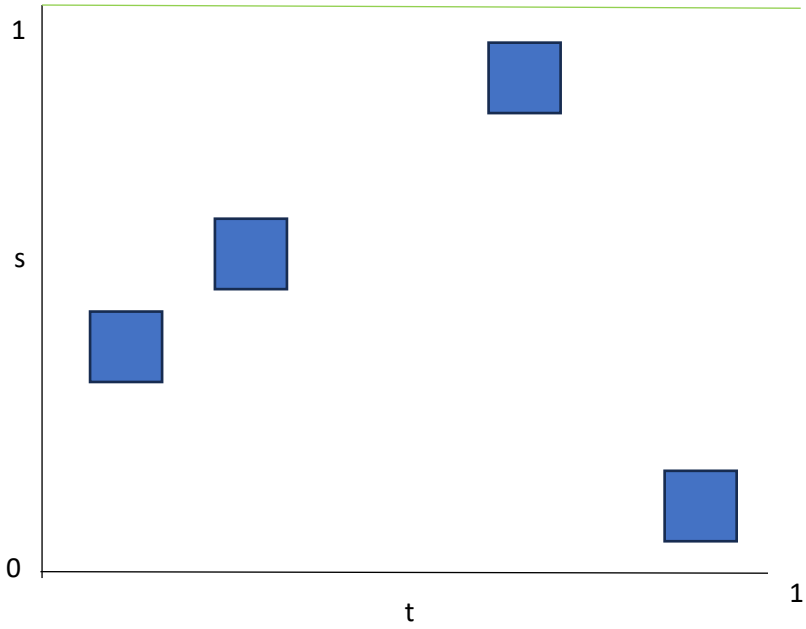
Here by Comparing the final paths of BUG1 and BUG2, it is evident that BUG1 path is shorter compared to BUG2.So, in this given map BUG1 algorithm performs better compared to BUG2.

2.

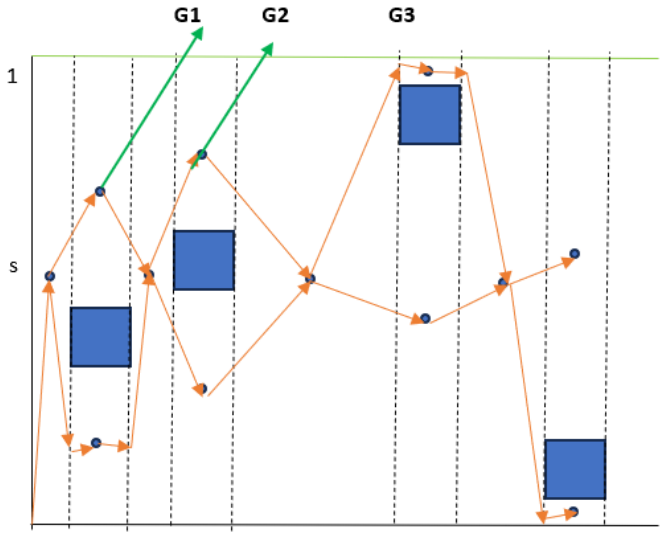


Assuming A robot moves form 0-s-1 and B moved from 0-t-1

State space:



Velocity tuning using vertical cell decomposition:



Here the G1, G2 ang G3 are the possible goals. The feasible solution would be G1 since it has only forward velocity.

3.

Bi Directional Astar:

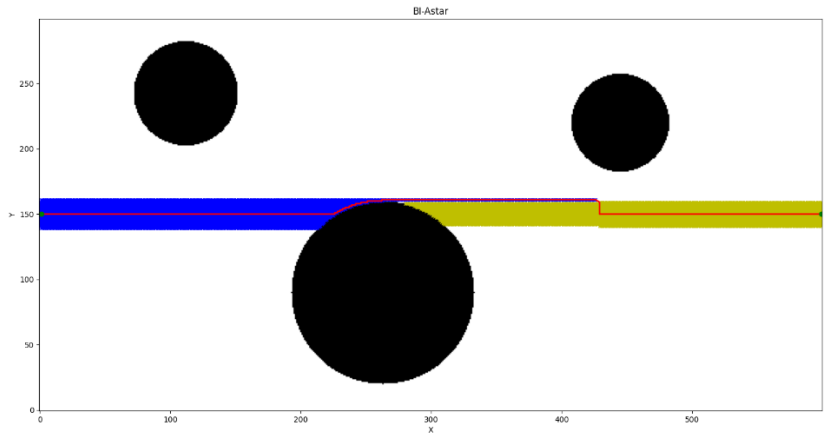
To run bi directional astar *python3 bi-astar.py*

Output:

Time taken for BI-Astar: 0.10818648338317871 seconds

Legend:

Back - obstacles
Blue - Tree1 explored nodes
Green - Tree2 explored nodes
Red - Final Path



Potential Field:

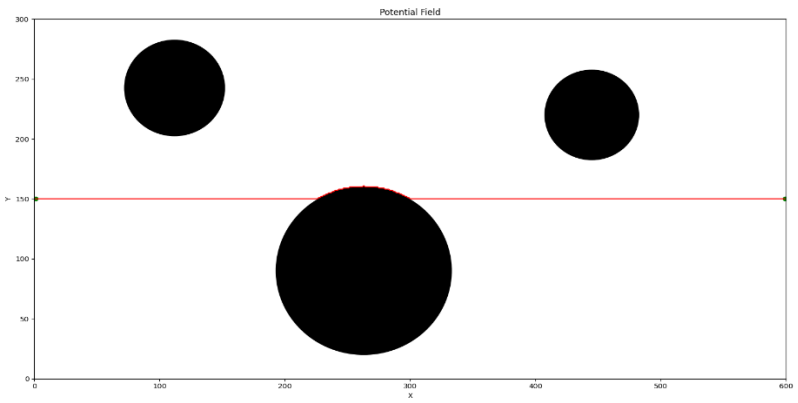
To run Potential Field *python3 potfield.py*

Output:

Time taken for Potential Field Algorithm: 0.59212327003479 seconds

Legend:

Back - obstacles
Red - Final Path



From the two algorithms,
1. It is clear that Bi-directional A-star runs faster than potential field, since Bi-directional A star converges from 2 ends.
2. Space complexity of Bi-directional A-star is more, as it stores all the nodes and parent.
3. Potential Field algorithm avoids the usage of cost function as the robot moves based on attractive and repulsive forces.
4. Potential field is suitable in dynamic environments where the algorithm prioritizes local optimization using current position, where as Bi-directional A-star is suitable for finding global optimal solution.

4.

Motivation:

Many robotics applications depend on multiple robots collaborating for (SLAM)simultaneous localization and mapping. In a centralized multi-robot SLAM system, all robots share their measurements with a central-server(distributed-server) for global optimisations. However, existing optimisation algorithms such as iSAM2 face challenges in efficiently handling measurements from multiple-robotic systems.

Method:

The author propose MR-iSAM2, an efficient incremental smoothing and mapping algorithm designed for centralized multi-robot SLAM. It relies on a novel-data-structure called (MRBT)Multi-Root Bayes Tree. The MRBT consolidates multiple Bayes Trees (from iSAM2) with the same underlying structure into one tree with various Root nodes. Different Robot's measurements contribute to growing the tree around a distnct root, thereby better capturing the distributed nature of the multi-robot problem.

MR-iSAM2 conducts incremental updates by integrating new measurements from each robot around its corresponding root in the MRBT. Techniques from iSAM2, such as fluid re-linearization and partial state updates, are adapted to preserve the MRBT structure during the updates. Moreover, the algorithm facilitates inter-robot information propagation by transmitting messages through the dual-directional edges connecting the multiple roots.

Advantages:

- Information can still propagate between roots for accurate global inference.
- Incremental updates are efficient by focusing only on the relevant root.
- MRBT represents multi-robot SLAM problems more naturally than a single Bayes tree.
- Growing the tree in separate branches for each robot better preserves sparsity.

Results:

The author assess MR-iSAM2 using synthetic multi-robot datasets (CityTrees) and a real 2D multi-robot dataset (UTIAS). Compared to the iSAM2 baseline, MR-iSAM2 demonstrates:

- Similar accuracy in terms of absolute trajectory error.
- Significantly fewer variables re-eliminated and cliques recalculated per update.
- Up to 10 times faster update times in multi-robot scenarios.
- Smaller average clique sizes, better capturing sparsity.

Conclusion:

MR-iSAM2 is an efficient incremental smoothing and mapping algorithm tailored for centralized multi-robot SLAM systems. By employing the multi-root Bayes tree representation and updating around the roots of different robots, MR-iSAM2 outperforms iSAM2 significantly in computational efficiency while maintaining similar accuracy. The multi-root approach yields promising results and holds potential for extension to other multi-robot inference problems.