International Journal of Sciences & Applied Research

www.ijsar.in

Performance comparison of sampling-based algorithms for path planning of a mobile robot

Sivaranjani Arthanari*, Vinod B.

Department of Robotics and Automation Engineering, PSG College of Technology, Coimbatore-641004, India.
Corresponding author: *Sivaranjani Arthanari, Department of Robotics and Automation Engineering, PSG College of Technology, Coimbatore-641004, India.

Abstract
With the emergence of mobile robots in commercial applications, it is a challenge for the research community to achieve efficient path planning and solve major issues in micro logistics. Mobile robots can move from the initial position to goal position without any collisions in the physical environment. The processing time, path length, and the collision-free path are the main constraints to get the optimal path of the mobile robot. Sampling-based algorithms are used to get the optimal path. This paper presents a performance comparison of sampling-based algorithms like A*, Bidirectional Rapidly Exploring Random Tree, Potential field, Probabilistic Road Map (PRM) and Rapidly Exploring Random Tree (RRT) which is used for path planning. The processing time and the path length for each algorithm are determined for the given map in a 2D environment with static obstacles. The A* algorithm gives the shortest path and the PRM algorithm takes the least processing time when compared to other algorithms.

Keywords: Path Planning; Sampling-Based Algorithms; Mobile Robot, MATLAB

1. Introduction

In the navigation of a mobile robot, path planning is most important. Mobile robot path planning solves three major problems [14]. The first one is the mobile robot should drive from the starting position to the goal position. The second one is, with a specific algorithm the robot should obtain the goal position by avoiding obstacles in an environment. The third one is the completion of the tasks as far as possible to optimize the robot trajectory as previously mentioned. A good path planning technology can reduce the wear and capital investment of the mobile robot.

The path planning problem is specified by two configurations named q_{init} and q_{goal}. q_{init} is called the initial configuration and q_{goal} is called the goal configurations. Any path in the free configuration that joins these two configurations is a solution to the path planning problem [2].

The general path planning method is as follows:

- Mark configurations q_{init} and q_{goal} in free configuration at random;
- Retain those configurations which lie in Cfree (along with q_{init} and q_{goal}) as the nodes of a graph P and connect adjacent configurations by links of P.

- Return yes if q_{init} and q_{goa}l belong to the same connected component of P.
- Return no if no path has been found after having generated configurations.

The answer yes is always correct, that is, whenever the planner returns yes, there exists a collision-free path connecting q_{init} to q_{goal} [2].

Several sampling-based approaches have been proposed and successfully applied to the problem of path planning. In sampling-based approaches, once the samples have been decided, where it will be placed, the next problem is to determine whether the configuration is in the collision. It is a critical component of sampling-based planning. In most planning applications, the majority of processing time is spent on collision checking. The five sampling-based approaches A*, Bidirectional Rapidly Exploring Random Tree, Potential field, Probabilistic Road Map (PRM) and Rapidly Exploring Random Tree (RRT) has been identified for the path planning of a mobile robot based on the research. Before the RRT, two popular approaches were introduced. Those are randomized potential field algorithm and the PRM algorithm. These techniques do not naturally extend to nonholonomic planning problems. So, the RRT algorithm was introduced for path planning to handle the nonholonomic constraints [11, 15].

2. Related work

RRT algorithm finds a valid path in a complex domain most of the time. Bin Feng et. al presented a paper on PRM and RRT algorithm for path planning. They concluded RRT algorithm is not feasible in the dynamic environment thus the generated path cannot guarantee safety in the dynamic domain. So that the safety RRT algorithm is generated to work in a dynamic environment with any obstacles (Bin Feng et al., 2014). It is more applicable to the real field environment such as RoboCup SSL competition.

MEA* and RRT*-AB algorithms are the advanced versions of A* and RRT algorithm. The path length, execution time, and memory requirements are the main important constraints in the path planning of the mobile robot. Noreen et al. [14] presented the performance comparison of MEA* and RRT*-AB algorithms. MEA* approach gives the shortest path and it takes less memory and computational time. For off-line application scenarios, the MEA* algorithm can be used because of its increased computational efficiency. RRT*-AB works in high-dimensional complex problems and it outperforms the MEA* algorithm.

PRM, A*, and Genetic Algorithm are used to find the shortest path for the multi-robot problem. In PRM, the sampling points are completely random and it is distributed throughout the configuration free space, Cfree. The drawbacks in the PRM are, the selected node is not in the ideal uniform distribution and the sampling points cannot completely cover the entire free space, and their distribution is random [7, 18]. The node enhancing method and geometric optimization are used to overcome the drawbacks. Safaa et al. [16] used an improved genetic algorithm in addition to the shortest path algorithms to make these paths more optimal to guarantee the shortest paths. First, the PRM algorithm is used to create the map in the free configuration space to avoid static obstacles [2, 4, 16]. Then by using the A* algorithm and the map created, the near-optimal path for each robot is found which represents the shortest path. This shortest path is ensured by the Genetic Algorithm.

Potential Field methods are more powerful and have intelligent algorithms. They can be used for more than two dimensions in different workspaces. Disha presents the concept of potential field method for motion of robot to move from starting position and reach the goal by avoiding the dynamic obstacles. The basic concept of this method is to fill the robot's workspace with an artificial potential field in which the robot is attracted to the target and is repulsed away from the obstacles. The main drawback of using this method is local minima and it occurs when the summation of all the forces is zero. The navigation function is designed for handling these drawbacks [3].

In autonomous navigation, the time it takes for the robotic system to recognize objects with sensors and the time the algorithm takes to process that information are the key concepts. Robotics system toolbox in MATLAB and ROS are used to navigate the mobile robot in addition to the generation of algorithms. PRM and Fast Marching Square were the two algorithms used by Galli et. al in the navigation of a mobile robot. The disadvantages in PRM are the solution it finds does not have to be the optimal path and it produces a non-homogeneous distribution of the samples in the space. FM^2 path planning algorithm looks for the optimal path between two points. There are drawbacks to the trajectories obtained. That is great abruptness of turns and trajectories that get very close to the obstacles. The main problem in this is execution time. Lack of fluency can occur for execution delays due to algorithms that require computational effort and can delay the sending of information. Several experiments were performed to test the effectiveness of algorithms. In PRM the range of optimal values is between 0.3 to 0.6 m [6].

RRTs have been widely used in autonomous mobile robot path planning. The challenges in RRTs-based planners are identifying the paths rapidly and passing through narrow passages in a robot's configuration space [1, 4, 16, 17]. Akash U et. al developed the ROS enabled robot with a slam to navigate autonomously. A particle filter algorithm is used in SLAM, to avoid collisions. In autonomous robot path planning, Rapidly Exploring Random Tree algorithm is used because of its ability to avoid obstacles efficiently [1].

3. Methodology

Path-planning requires a map of the environment and the mobile robot must be aware of its location concerning the map. MS paint is used to draw the map and it is created with static obstacles. A 2D robot workspace consists of the black and white area, represents the environment of the robot which is shown in Figure 1. The initial position is represented as S inside the black area and the goal position is marked in a white dot inside the small black dot. The white area represents the free space where the robot can move without any problem. The black area represents the obstacles where the robot cannot move. Based on the literature survey, the A*, Bidirectional Rapidly Exploring Random Tree, Potential field, Probabilistic Road Map and Rapidly Exploring Random Tree algorithms have been identified to experiment for the given map with static obstacles and tested in MATLAB.

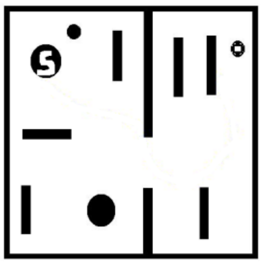


Figure 1. Representation of Map

3.1. A Algorithm*

A* algorithm is a graph search algorithm mostly used in robotics to find the shortest path in navigation. The A* algorithm would take a very large computation time on a high-resolution map. The map resolution is hence reduced, while the resolution is an algorithm parameter. In general, the higher the resolution of the map, the better is the results with excessive computation time. A* works by making a lowest-cost path tree from the start node to the goal node. A* uses a function f(n) that gives an estimate of the total cost of a path using that node. It expands paths that are already less expensive by using this function:

$$f(n) = g(n) + h(n) \qquad (1)$$

where $f(n)$ is total estimated cost of the path through node n, $g(n)$ is cost so far to reach node n and $h(n)$ is estimated cost from n to goal also called a heuristic function. This is a function that will try to estimate the cost to reach the target node. A good heuristic function will achieve that fewer nodes will have to be visited to find the target. While Dijkstra's algorithm would expand to all sides, A* will search only in the direction of the target.

3.2. Bidirectional Rapidly Exploring Random Tree Algorithm

The better performance can usually be obtained by growing two trees, one from the starting position and the other from the goal position. The expansion of Bidirectional RRT graph which is represented as P_a and P_b is shown in the Figure 2.

3.3. Potential Field Algorithm

A potential field algorithm uses the artificial potential field to control the mobile robot around in the configuration space. The output of a potential function can be taken as the energy and it is a negative gradient as the net force acting on the mobile robot. All obstacles repel the robot with a magnitude inversely proportional to the distance. The goal attracts the robot. The robot simulates from the highest potential to the lowest potential. The goal node has the lowest potential while the initial node will have the maximum potential. The resultant potential, accounting for the attractive and repulsive components is measured and used to move the robot [3]. The local minima is the main problem in the potential field algorithm.

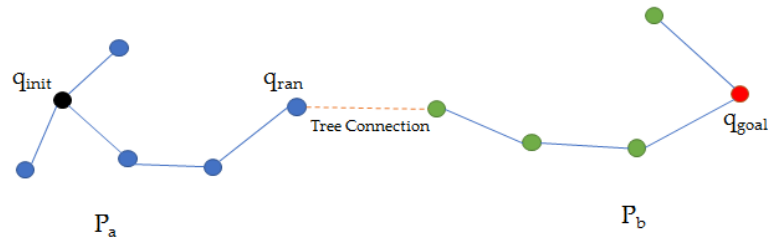


Figure 2. Expansion of Bidirectional RRT

The bidirectional RRT algorithm for path planning of a mobile robot is given below.

```
BIDIRECTIONAL_RRT ($q_{init}$, $q_{goal}$)
1   $P_a$.init ($q_{init}$); $P_b$.init ($q_{goal}$);
2   for k=1 to K do
3           $q_{ran}$ ← RANDOM_CONFIG ();
4           if not (EXTEND ($P_a$, $q_{ran}$) = Trapped) then
5               if (EXTEND ($P_b$, $q_{new}$) = Reached) then
6                   Return Path ($P_a$, $P_b$);
7           SWAP ($P_a$, $P_b$);
8   Return Failure;
```

3.4. Probabilistic Roadmap Algorithm

Probabilistic techniques build a roadmap in Configuration space (C space). The algorithm has two stages: one is the learning phase (construction and expansion) and the query phase. The learning phase aims to randomly draw a small graph across the workspace. All the vertices and edges of the graph should be collision-free so that a robot may use the same graph for its motion planning. The PRM selects several random points in the workspace as the vertices. To qualify to be a vertex, a randomly selected point must not be inside the obstacles. The algorithm then attempts to connect all pairs of randomly selected vertices. If any two vertices can be connected by a straight line, the straight line is added as an edge. To determine the path between the initial point and goal point the graph search algorithm is applied. Lydia E. Kavraki invented this PRM algorithm [3,16].

3.5. Rapidly Exploring Random Tree Algorithm

The path of the mobile robot is planned by building a tree initially starting from the position of the robot. A point is randomly sampled in the space and it is checked if that point collides with an obstacle in the space. If the sampled point has no collisions, it is then checked if the straight-line path between the sampled point and the nearest existing point in the tree has any collisions. If this straight-line path has no collisions, the sampled point is added to the tree with the nearest point as its parent node. If there is a collision, this point is not considered.
Each time after a node is added to the tree and if the node is less than some threshold distance from the goal position, it is checked if the goal can be reached in a straight-line path from the added node. If the goal position is reachable, the recently added node is added to the goal position in the tree. At this point, the path planning is complete. If the goal position is still unreachable, additional points are sampled. The expansion of the RRT graph which is represented as P is shown in Figure 3.

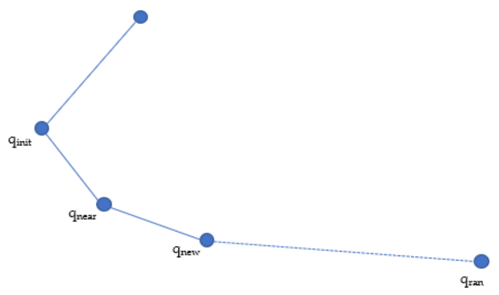


Figure 3. Expansion of RRT

The RRT algorithm for path planning of a mobile robot is given below.

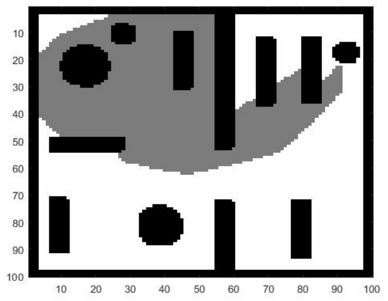
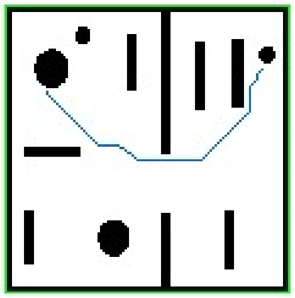
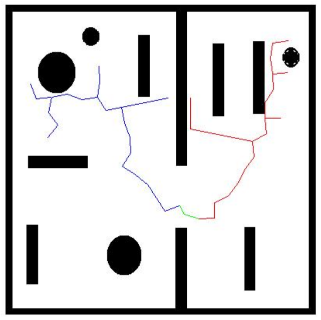
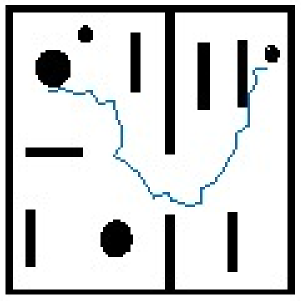
```
-----------------------------------------------------------------------------------------------------------
BUILD_RRT ($q_{init}$)
1       P. init ($q_{init}$);
2       for k=1 to K do
3               $q_{ran}$ ← RANDOM_CONFIG ();
4       EXTEND (P, $q_{ran}$);
5       Return P;
EXTEND (P, q)
1       $q_{near}$ ← NEAREST_NEIGHBOUR (q, P);
```

```
2   if NEW_CONFIG (q, $q_{near}$, $q_{new}$) then
3       P. add_vertex ($q_{new}$);
4       P. add_edge ($q_{near}$, $q_{new}$);
5       if $q_{new}$ = q then
6           Return Reached;
7       else
8           Return Advanced;
9   Return Trapped;
```
----

4. Discussion

Sampling-based path planning algorithms are simulated in the MATLAB 2019a for the given map with static obstacles. Each algorithm is simulated individually and the final path is shown in Table 1. Simulations of the sampling-based approaches are carried out several times. The processing time and the path length are displayed at the command window.
The comparison of each algorithm based on the path length and the processing time is shown in Table 2. The distance taken is in SI units and the processing time is in seconds. Based on the comparison the A* algorithm gives the shortest path but the processing time taken is high of all the algorithms. RRT algorithm gives the longest path. PRM algorithm takes minimum processing time. The mobile robot size is 10 x 10 units and the robot speed is 10 units per unit time. The step size of the robot is 20. The comparison of algorithms based on path length and the processing time is shown in Figure 4 and Figure 5.

Table 1: Expansion and Final path of Sampling-based Algorithms

Sampling-based Algorithms	**Expansion**	**Final Path**
A*		
Bidirectional RRT		

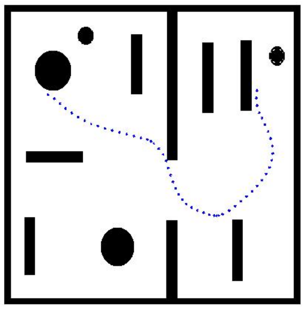
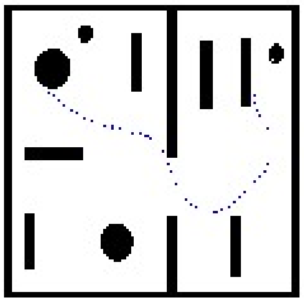
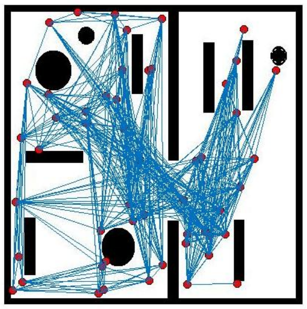
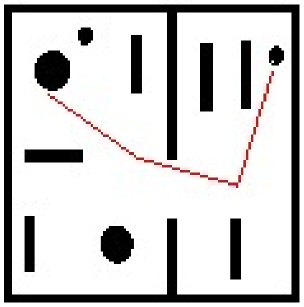
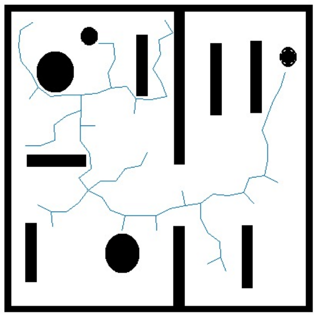
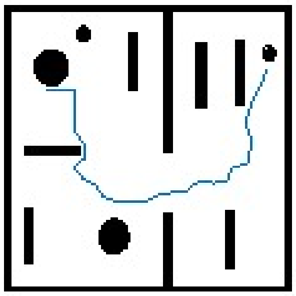
Potential Field		
PRM		
RRT		

Table 2: Path Length and Processing Time of Sampling-based Algorithms

Sampling-based Algorithms	**Path length in metrics**	**Processing Time in metrics**
A*	422.56	121.74
Bidirectional RRT	577.93	2.21
Potential Field	503.57	2.38
PRM	453.04	0.78
RRT	585.17	5.84

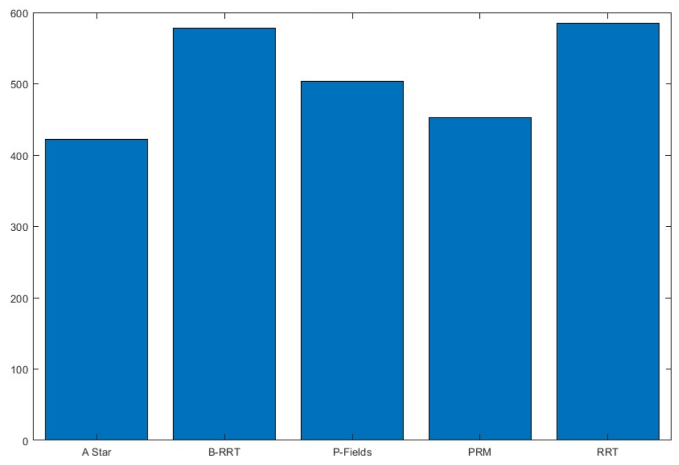


Figure 4. Comparison of Algorithms Based on Path Length

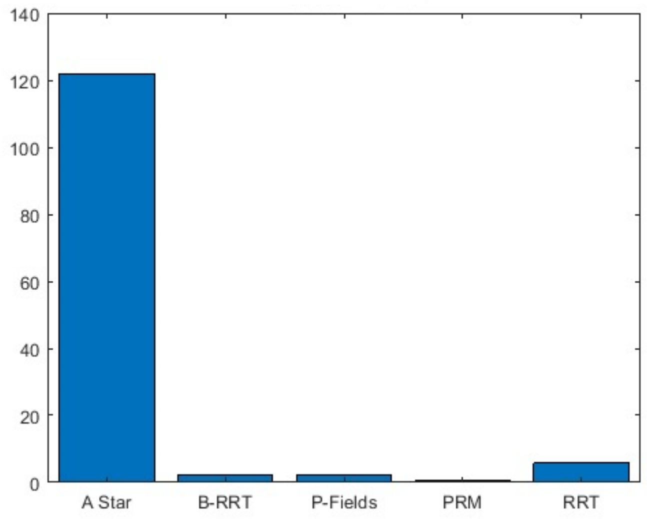


Figure 5. Comparison of Algorithms Based on Processing time

5. Conclusion

In this paper, the performance comparison of sampling-based algorithms is presented. The path length and the processing time were found and compared for the created map with static obstacles. Simulation results show that the A* algorithm gives the shortest path but the processing time taken by it is very high of all the algorithms. Probabilistic Road Map algorithm takes the least processing time to reach the goal position. So, the implementation of each algorithm depends on the application. Overall when compared to path length and the processing time, the PRM algorithm takes the least processing time and shortest path. Further, the algorithms will be tested in a dynamic environment.

References

1. Akashet al., 2019. Autonomous Navigation with Collision Avoidance using ROS. Journal of Remote Sensing GIS & Technology, Vol 5, No. 2, 14 – 18.
2. Barraquand et al., 1997. A random sampling scheme for path planning. International Journal of Robotics Research, Vol. 16, no. 6, 759–774.

3. Disha, 2018. Path Planning for Mobile Robots using Potential field Method. Research Gate.
4. Elbanhawi and Simic, 2014. Sampling-Based Robot Motion Planning: A Review. IEEE - Translations and content mining, Vol 2, 56-77.
5. Feng and Liu, 2014. An improved RRT based path planning with safe navigation. Applied Mechanics and Materials, Vol. 494-495, 1080-1083.
6. Galli et al., 2017. Path planning using Matlab-ROS integration applied to mobile robots. IEEE International Conference on Autonomous Robot Systems and Competitions.
7. Gang and Wang, 2015. PRM path planning optimization algorithm research. WSEAS transactions on systems and control, Vol 10, 148-153.
8. Goerzen et al., 2010. A survey of motion planning algorithms from the perspective of autonomous UAV guidance. Journal of Intelligent and Robotic Systems: Theory and Applications, 57, 65–100.
9. Kavraki, et al., 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation, 12, 566–580.
10. Kuwata, et al., 2008.Path Planning Challenges for Planetary Robots. IEEE Intelligent Robotic Systems, 22–27.
11. LaValle, 1998. Rapidly-Exploring Random Trees: A New Tool for Path Planning; Technical Report; Computer Science Dept., Iowa State University: Ames, Iowa.
12. LaValle, 2006. Planning Algorithms; Cambridge University Press: Cambridge, UK.
13. Noreen et al., 2016. Optimal Path Planning using RRT* Based Approaches: A Survey and Future Directions. International Journal of Advanced Computer Science & Applications, 7, 97–107.
14. Noreen et al., 2019. A Path- Planning Performance Comparison of RRT*-AB with MEA* in a 2-Dimensional Environment. Symmetry, https://doi.org/10.3390/sym11070945, 11, 945.
15. Qureshi and Ayaz, 2015. Intelligent Bidirectional Rapidly-exploring Random Trees for Optimal Motion Planning in Complex Cluttered Environments. Robotics and Autonomous Systems, 68, 1–11.
16. Safaa et al., 2014.Probabilistic Roadmap, A*, and GA for Proposed Decoupled Multi-Robot Path Planning. IRAQI Journal of Applied Physics.
17. Wang et al., 2010. Triple RRTs: An effective method for path planning in narrow passages. Advanced Robotics, Vol 24, 943–962.
18. Zhanget al., 2018a.Path planning for mobile robot based on modified rapidly exploring random tree method and neural network. International Journal of Advanced Robotic System.
19. Zhanget al., 2018b. Path Planning for the Mobile Robot: A Review. Symmetry,10, 450.