

Research on robot path planning methods

Tianhao Chen^{1,3,†}, Guanhong Jiang^{2,†}

¹Department of Computer science and engineering, Shanghai University, 200444, China

²Department of Science, Hong Kong Baptist University, Hong Kong, 999077, China

³chentianhao1106@shu.edu.cn

[†]All authors contributed equally.

Abstract. Mobile robots are used extensively across a variety of industries and fields. How to discover a start-to-end path without colliding has become a hot topic in recent years due to the complexity and uncertainty of the workplace. In various environments, a path planning technique should demonstrate high efficiency and speed. And this can reduce the energy consumption of the robots and greatly increase their working efficiency. This paper will conclude the presently popular path planning algorithm. Based on the different features of these algorithms, they are divided into three types: traditional path planning algorithm, neural-work-based algorithm, and sampling-based algorithm. Based on the new papers in these years, detailed introduction of the algorithms and their variants will be given. At the end of the paper, the thesis is summarized and the future research trend is prospected.

Keywords: path planning, traditional algorithm, neural-work-based algorithm, sampling-based algorithm.

1. Introduction

Mobile robots have been widely applied in different industry and field. Due to the uncertainty and complex of the working environment, how to find a start-to-end path without collision becomes a hot topic nowadays. An effective path planning method should show high efficiency and speed in different environment [1]. And this can significantly improve the working efficiency of the robots and reduce the energy consumption of it.

In recent years, many review articles on path planning have been published. For example, Kruse et al. surveyed the socially-aware trajectory planning [2-3]. It puts greater emphasis on robot behavior during the navigation. Chik et al. divided path planners for robot navigation into a global planner and a local planner. Although a few researches have been made on the robot path planning, few have been done to make a comprehensive study on traditional and state-of-art path planning algorithms. Thus, a detailed survey on different path planning algorithms is presented.

This paper will conclude the presently popular path planning algorithm. Based on the different features of these algorithms, they are divided into three types: traditional path planning algorithm, neural-work-based algorithm, and sampling-based algorithm. Based on the new papers in these years, detailed introduction of the algorithms and their variants will be given.

2. Traditional methods

Path planning is a crucial area of study in robotics, with consequences for automated guidance, impediment avoidance, and best path selection. The goal of this section is to analyze traditional algorithms from two different angles, which are based on obstacle avoidance algorithm and based on graph network construction, and give a comparison and analysis.

2.1. Methods based on obstacle avoidance

Artificial Potential Field (APF), a popular obstacle avoidance algorithm, is a passive obstacle avoidance method in which the robot generates a virtual potential field around obstacles in its environment and guides it away from regions of the high potential field. The theory behind APF is that a robot is attracted to a goal and rebounded by obstacles in its environment [4]. The algorithm generates a potential field around the robot, where targets are assigned low potential values and obstacles are assigned high potential values. The robot then turns to the lowest potential value, which will Orient it toward the goal and away from the obstacle. APF has a wide range of applications in robotics, including mobile robots, unmanned aerial vehicles (UAVs), and autonomous underwater vehicles (AUVs).

The bug algorithm is a direct reactive obstacle avoidance method proposed by Lumelsky and Stepanov. By following the contour of the obstacle to bypass the obstacle, the robot advances toward the desired destination. Two appreciated variants are Bug1 and Bug2. The basic defect method is Bug1. To avoid obstacles, it navigated in a direct motion. When encountering an obstacle, the robot rotates in any direction until it finds a way out. While this is simple, it can lead to invalid routes and obstacles. The wiggly bug method is called Bug2. The Bug1 algorithm works by moving along the edge of an obstacle when it is encountered until it reaches the goal point. The advantages of this algorithm are that it is simple to understand and easy to implement, and it can also achieve good results in simple environments. However, it has the disadvantage that it may go around obstacles, resulting in an increased path length [5]. The Bug2 algorithm is a modified version of the Bug1 algorithm, which detects whether there is a shorter path to the goal point when moving along the edge of an obstacle. The advantage of the proposed algorithm is the ability to avoid circling obstacles, thus reducing the path length. However, compared with the Bug1 algorithm, the Bug2 algorithm is more complex to implement, and path jams may occur in complex environments.

Vector Field histogram (VFH) is a commonly used algorithm for robot obstacle avoidance. It was originally proposed by Borenstein and Koren as a method to help robots navigate in dynamic environments. The VFH algorithm works by creating a 2D histogram of the environment around the robot, which represents the likelihood of an obstacle at each location [4]. Using this histogram, the algorithm then generates a polar histogram representing the free-space orientation of the robot. Finally, the robot chooses a steering direction that maximizes free space and minimizes the possibility of collision with obstacles. The theory behind VFH is based on the idea of a vector field, a mathematical representation of the forces acting on an object in a physical system. In the case of VFH, the vector field represents the force exerted by the robot as it moves through the environment.

Compared with these three methods, the APF method can quickly deal with large-scale environments, and can deal with multi-objective path planning problems. And using the APF method, robots can avoid obstacles and find the shortest path, but it is easy to fall into local optimal solutions, and some skills are needed to avoid this situation. The APF method may produce a "vortex effect" and cause the robot to fail to reach the target point. Because of this, the APF method is most suitable for dealing with large-scale environments and multi-objective path planning problems. Since the APF method uses an artificial potential field to describe the interaction between the robot and the environment, it can quickly deal with large-scale environments and can deal with the path planning problem of multiple target points simultaneously [6]. When dealing with multi-objective path planning problems, APF is better than Bug Algorithm and VFH. The Bug algorithm is suitable for complex environments and can deal with obstacles, gaps, and other situations that appear in the environment. At the same time, the Bug algorithm is relatively simple and easy to implement, however, this algorithm may produce a "surround effect", which causes the robot to fail to reach the goal point. In addition, the Bug algorithm requires a high

initial estimation of the environment and needs to know the terrain and other information about the environment in advance. This is why Bug works best when dealing with complex terrain and situations where there are obstacles and gaps. The Bug algorithm uses the distance between pairs of points (robot and goal point) and the distance between the robot and obstacles to plan the path, so it can handle environments with complex terrain and problems such as obstacles, gaps, etc. The bug algorithm is better than APF and VFH when dealing with an environment with complex terrain and problems such as obstacles and gaps. Different from the remaining two, the VFH method can quickly and accurately detect obstacles in the environment, select the best path, and can be applied to complex environments, and can deal with various obstacles in the environment. However, this method has high requirements for environment modeling and needs to accurately estimate the position and shape of obstacles. When dealing with multi-objective path planning problems, the VFH method may only have local optimal solutions. The VFH method is most suitable for handling application scenarios that need to detect obstacles in the environment quickly and accurately. The VFH method uses lidar to scan the environment, then generates a histogram from the scanned data, and selects the best path based on the histogram. Because the VFH method can quickly and accurately detect obstacles in the environment and select the best path, it is more excellent than APF and Bug Algorithm in application scenarios that need to detect obstacles quickly and accurately in the environment.

In summary, the APF method is suitable for dealing with large-scale environments and multi-objective path planning problems. The bug algorithm is suitable for complex environments and situations that need to deal with obstacles and gaps in the environment. The VFH method can detect obstacles in the environment quickly and accurately and is suitable for dealing with complex environments. Which approach you choose depends on the specific scenario and problem requirements.

2.2. Methods based on graph network construction

Grid-based methods are a family of algorithms used in robotics and autonomous systems for planning the motion of a robot or vehicle. They operate by representing the robot's environment as a grid of cells, where each cell corresponds to a location in the space, and each cell is assigned a value that represents the accessibility or cost of that location. The history of grid-based methods can be traced back to the 1980s when researchers started to develop algorithms for robot path planning. In the early days, grid-based methods were simple and often used a binary representation to represent the obstacles in the environment. Over time, researchers developed more sophisticated grid-based methods that could represent the environment more accurately and efficiently. Another possible method is Voronoi Diagram. Paths are created by Voronoi Diagram methods along the diagram, which divides the world into regions based on how close barriers are to one another. To maximize its escape from obstructions, the robot moves along Voronoi edges.

Both the Grid-Based method and Voronoi Diagram are methods used for spatial analysis, but they each have some differences and advantages and disadvantages. The grid-based method is to divide the spatial region into uniform grids and treats each Grid as a unit. The advantage of this approach is that it can handle large-scale data because the computation of each lattice is relatively independent and can therefore be processed in parallel. Moreover, since the size of the lattice is fixed, the results of the proposed method are also predictable for data with different resolutions. The disadvantage is that since the size of each cell is fixed, it may lead to defects or errors in the analysis results, especially when the data density and distribution are uneven in different spatial regions. So, when the data is evenly distributed in the space or when the amount of data is large, the Grid-Based method can better process the data, because it can evenly divide the space area into grids and processes, which can take full advantage of parallel computing. In contrast, Voronoi Diagram is a geometrically based method that partitions a spatial region into polygonal cells centered at data points, where each cell contains all points that are at an equal distance to the nearest data point. The advantage of this method is that it can capture the characteristics of the data more accurately because it can adapt to different data distributions depending on the distribution of the data points. In addition, since the proposed method uses geometric shapes, it can better express spatial relationships and spatial similarities. The drawback is that the

computational complexity of the method is high, so it may not be able to handle large-scale data. So, when the data is unevenly distributed in space, the use of the Voroni Diagram can better capture the characteristics of the data, and when the data has a complex spatial distribution, the use of the Voroni Diagram can better represent the spatial relationship and spatial similarity.

To put it simply, traditional algorithms are frequently simple to comprehend and apply, making them a common option for many route-planning tasks. These algorithms have been thoroughly investigated and implemented in numerous situations, showing their efficacy in solving path-planning issues. They can also be modified or combined to better fit the needs of a given application. However, some conventional methods might produce inefficient or less secure routes for the automaton. Local minima or fluctuations in some algorithms, like the Artificial Potential Field technique, can impede the robot's progress. Additionally, some methods might be computationally difficult, particularly in settings with lots of obstacles or high-resolution models.

3. Neural-network based methods

3.1. Reinforcement learning

Reinforcement learning (RL) is an effective method that helps the agent make the best choice based on the past actions and results without any provided samples to learn in advance. In the field of path planning, this method uses feedbacks from the environment as the reward to stipulate the best path. Figure 1 shows a widely used RL model, the actor critic model.

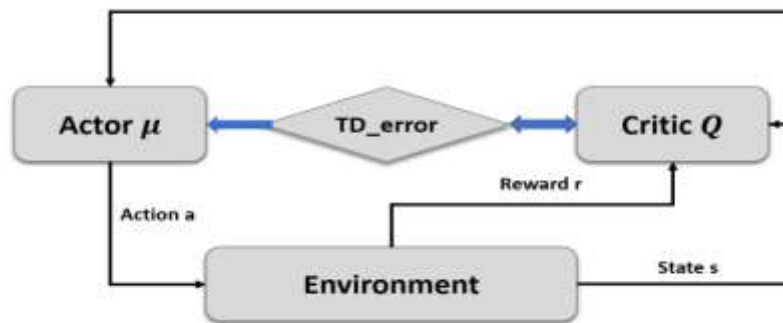


Figure 1. The actor critic model.

RL and its variants have been widely applied for robot path planning, especially in real-time tasks. Zhang et al. propose a path planning model called SG-RL. It uses the Simple Subgoal Graphs (SSG) to look for the best paths, and reinforces the decision-making process with the Least-Squares Policy Iteration (LSPI) method [7]. It's proved that this model can better adapt for the dynamically changing environment and achieve good performance on large-scale maps, since SSG resolves the limitations of sparse reward and local minima trap for RL agents and thus LSPI can be applied to deal with slight changes from the environment.

To raise the ability of the robots to perform more complex missions, GyeongTaek proposes a Goal-Conditioned RL method (Figure 2). It uses the bi-directional memory editing to raise its robustness and a sub-goal dedicated network to ensure the agent is fully controllable [8]. The reward function is redesigned to help shape the shortest path on propose for the optimal performance.

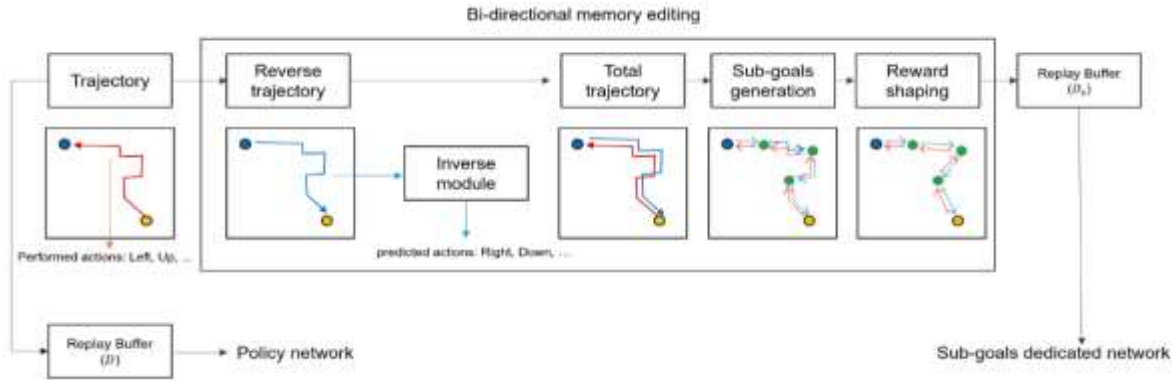


Figure 2. Illustration of the GC-RL method.

Deep Reinforcement Learning (DRL), a method combining RL and DL (Deep Learning) has also been a hot topic in path planning. Bouhamed et al. designed a Deep Deterministic Policy Gradient (DDPG) based DRL approach which exhibits stronger real-time learning ability from surrounding environment. In autonomous robot exploration tasks, Yuhong Cao et al. propose a neural network named ARiADNE, combining the attention mechanism and DRL [9]. It can learn spatial dependencies from the partial map and predict possible gains for unexplored areas, which assist the agent to make non-myopic movement decisions. It is proved to outperform most state-of-the-art methods. Miguel et al. compare two deep Q network (DQN) methods, the D3QN and rainbow algorithm in path planning tasks, and make the conclusion that the rainbow DQN performs better in most tasks, suggesting its feasibility in DRL approach design.

3.2. Deep learning

Compared to RL methods, Deep learning (DL) method learn to plan path based on extracted features from samples. It is suitable for tasks having a large number training samples.

Many scholars apply DL method in mobile robots, among which the Convolutional Neural Network (CNN) is the most widely adopted [10]. To improve the algorithm's efficiency in large and complex environment, Janderson et al. propose a CNN encoder to overcome the limit that the model can only extract features from linear information, thus reducing data dimensionality, namely useless paths in the environment. This model is proved to decrease the execution time by 54.43%. Shen et al. propose a Coverage Path Planning Network (CPPNet) [11]. A CNN network with graph-based input and output. The edge value of the output graph is the probability of belonging to the TSP tour, and a greedy search is used to find the best TSP tour. Liu et al. combine the CNN and RRT* algorithm, and design a learning-based algorithm, which regards the environment as a RGB image input to predict unexplored environment, assisting the RRT* planner to make faster path stipulation.

4. Sampling-based methods

Sampling-based method find the feasibility of the path through collision detections, avoiding the detailed expression of the environment. Sampling-based algorithm connects all of the feasible nodes and find a start-to-goal feasible path based on it [12-13]. The most widely used sampling-based approaches are probabilistic roadmaps (PRM) and random-exploring trees (RRT).

4.1. PRM algorithm

PRM algorithm includes two phases: a learning phase and a query phase. In the learning phase, it constructs the roadmaps with nodes and edges representing feasible paths and uses a local fast planner to calculate them. Then in the query phase, it will find the optimal one given the start and goal configurations.

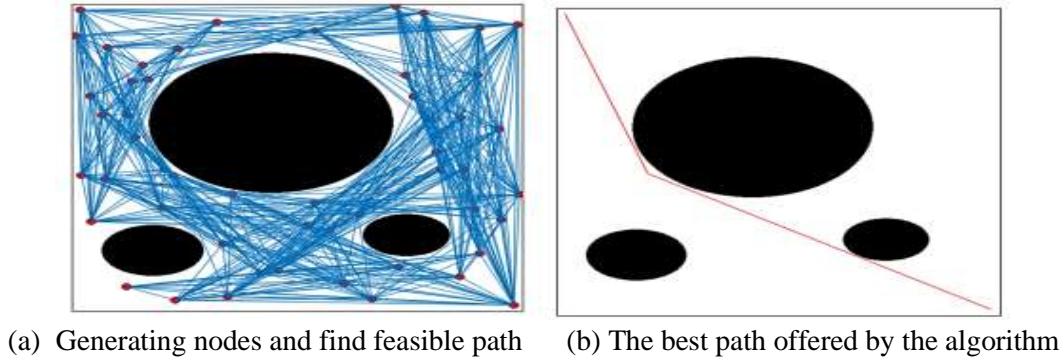


Figure 3. PRM algorithm working diagram.

PRM algorithm performs well in high-dimensional search space, but in some tasks, it is not that stable (Figure 3). To resolve this limitation, Yang et al. propose a sample adjustment method during the construction of the roadmaps. This post-processing method will adjust the randomly generated nodes to meet the soft constraints, which influence the behavior of the agent, required by the problems. To better adapt for the dynamic environment, Ahmed et al. improve the PRM algorithm by dividing the domain of motion and deal with the relevant path in each of it. Chen et al. propose a modified algorithm called P-PRM, which introduces in the concept of the potential field in its planning area. The new part is adopted to choose valuable nodes to avoid collision for the sampling points, and cost function is specifically designed to avoid local optimum. Compared to traditional PRM algorithm, this method has higher efficiency and faster execution time. Ankit et al. propose a method called HPPRM to solve narrow passage problems [14]. It distributes nodes through segmenting roadmaps into high and low potential areas and reduce the dispersion of sample set during roadmap construction. It is proved that it has greater success rate and lower calculation cost than traditional methods.

4.2. RRT algorithm

RRT algorithm was proposed by Lavelle et al. in 1998. It constructs random trees to achieve a start-to-end path without collision. The RRT regards the start point as the root and find the best path through costly regeneration (Figure 4).

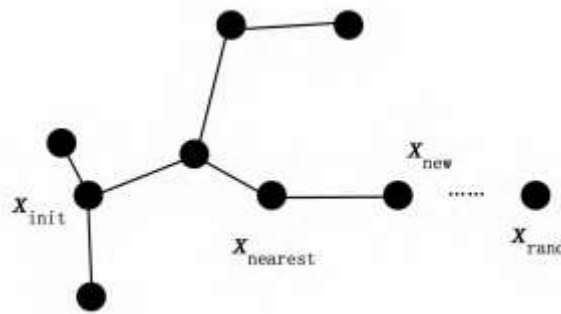


Figure 4. shows the process of the RRT.

The RRT algorithm has a great adaption to the environment, therefore it's applied to real-time task. To improve the performance of the RRT algorithm in different dynamic environment and multi-query tasks, Daniel et al. proposes a method, AM-RRT*, which can extend the RRT-based sampling approach and use an assisting metric to store beneficial results [15]. Experimentations demonstrate the improved method's effect in execution time reduction compared to RT-RRT*. Thomas et al. propose a Grounding-aware RRT* algorithm, which can be applied to marine tasks, to improve its ability to avoid collision. Previous data from both the environment and navigation experience is encoded and transferred to

RRT*'s cost function, so that path deviation can be penalized and a better path alternation is therefore made. Meng et al. combine the deep learning with the RRT algorithm to widen the applicable field of RRT planner. A new algorithm named NR-RRT is presented. It implements a neural network sampler to improve the safety of possible chosen state, and use the bi-directional search strategy to fasten the execution time. It is tested that this algorithm performs better trade-off between efficiency and safety than other state-of-order method [16]. Energy consumption of RRT algorithm remains a hot topic in the research of RRT algorithm. To have better efficiency, Pedram et al. propose the information-geometric RRT* (IG-RRT*) algorithm. The problem is better resolved by reducing the large number of nodes needed to deal with. Only a part of these nodes is chosen to be run on, and a smoothing algorithm, which can be seen as an optimization function, is applied to adjust the path planning result.

5. Conclusion

In conclusion, path planning is a critical component of robotics and autonomous systems, and there are various methods available for its implementation. This essay has introduced both traditional algorithms, such as Bug Algorithms, VFH, APF, and Grid-based methods, as well as neural-network based algorithms, such as DRL and CNN. Traditional algorithms have been in use for many years and have proven to be effective in many applications. They are often simpler and more interpretable than neural-network based algorithms, making them a popular choice for certain applications. However, traditional algorithms may struggle in complex, dynamic environments, and may require extensive tuning to achieve optimal performance. Neural-network based algorithms, on the other hand, are increasingly popular due to their ability to learn complex behaviors and adapt to changing environments. They have shown impressive results in many applications, particularly in robotics and autonomous vehicles. However, they can be computationally expensive and difficult to interpret, which may limit their use in some applications.

The future of path planning lies in the integration of both traditional and neural-network based algorithms, with each method being used for its strengths. The use of artificial intelligence-based approach in path planning is rapidly evolving, and it promises to revolutionize the way we interact with robots and autonomous systems in the future.

References

- [1] Tang, Z., & Ma, H. An overview of path planning algorithms. 2021 Earth and Environmental Science. 804 (2), p. 022024.
- [2] Li, X., Hu, X., Wang, Z., & Du, Z. Path planning based on combination of improved A-STAR algorithm and DWA algorithm. 2020 International Conference on Artificial Intelligence and Advanced Manufacture, 99-103.
- [3] Boots, B., Sugihara, K., Chiu, S. N., & Okabe, A. Spatial tessellations: concepts and applications of Voronoi diagrams 2009, John Wiley & Sons.
- [4] Chen, W., Wang, N., Liu, X., & Yang, C. VFH based local path planning for mobile robot. In 2019 China Symposium on Cognitive Computing and Hybrid Intelligence, 18-23.
- [5] Xu, Q. L., Yu, T., & Bai, J. The mobile robot path planning with motion constraints based on Bug algorithm. 2017 Chinese Automation Congress, 2348-2352.
- [6] Chen, Y. B., Luo, G. C., Mei, Y. S., Yu, J. Q., & Su, X. L. UAV path planning using artificial potential field method updated by optimal control theory. 2016 International Journal of Systems Science, 47(6), 1407-1420.
- [7] Zeng, J.; Qin, L.; Hu, Y.; Hu, C.; Yin, Q. Combining Subgoal Graphs with Reinforcement Learning to Build a Rational Pathfinder. 2019 Application. Science, 9, 323.
- [8] Lee, GyeongTaek. A Fully Controllable Agent in the Path Planning using Goal-Conditioned Reinforcement Learning. 2022 10.48550/arXiv.2205.09967.
- [9] Cao, Y., Hou, T., Wang, Y., Yi, X., & Sartoretti, G. ARiADNE: A Reinforcement learning approach using Attention-based Deep Networks for Exploration. 2023 ArXiv, abs/2301.11575.

- [10] Y. Zhang, J. Zhao and J. Sun, Robot Path Planning Method Based on Deep Reinforcement Learning, 2020 International Conference on Computer and Communication Engineering Technology, 49-53.
- [11] Z. Shen, P. Agrawal, J. P. Wilson, R. Harvey and S. Gupta, CPPNet: A Coverage Path Planning Network, 2021 OCEANS. 1-5.
- [12] J. Liu, B. Li, T. Li, W. Chi, J. Wang and M. Q.H. Meng, Learning-based Fast Path Planning in Complex Environments, 2021 IEEE International Conference on Robotics and Biomimetics. 1351-1358.
- [13] E. M. Ahmed, H. E. Abd El Munim and H. M. Shehata Bedour, An Accelerated Path Planning Approach, 2018 International Conference on Computer Engineering and Systems, 15-20.
- [14] A. A. Ravankar, T. Emaru and Y. Kobayashi, HPPRM: Hybrid Potential Based Probabilistic Roadmap Algorithm for Improved Dynamic Path Planning of Mobile Robots, 2020 IEEE Access, 8 221743-221766.
- [15] D. Armstrong and A. Jonasson, AM-RRT*: Informed Sampling-based Planning with Assisting Metric, 2021 IEEE International Conference on Robotics and Automation, Xi'an, 10093-10099.
- [16] Pedram, Ali Reza & Tanaka, Takashi. A Smoothing Algorithm for Minimum Sensing Path Plans in Gaussian Belief Space. 2023 IEEE Transactions on Robotics 32(5).